ELODI: Ensemble Logit Difference Inhibition for Positive-Congruent Training

Yue Zhao, Yantao Shen, Yuanjun Xiong, Shuo Yang, Wei Xia, Zhuowen Tu, Bernt Schiele, Fellow, IEEE, and Stefano Soatto, Fellow, IEEE

Abstract—Negative flips are errors introduced in a classification system when a legacy model is updated. Existing methods to reduce the negative flip rate (NFR) either do so at the expense of overall accuracy by forcing a new model to imitate the old models, or use ensembles, which multiply inference cost prohibitively. We analyze the role of ensembles in reducing NFR and observe that they remove negative flips that are typically not close to the decision boundary, but often exhibit large deviations in the distance among their logits. Based on the observation, we present a method, called Ensemble Logit Difference Inhibition (ELODI), to train a classification system that achieves paragon performance in both error rate and NFR, at the inference cost of a single model. The method distills a homogeneous ensemble to a single student model which is used to update the classification system. ELODI also introduces a generalized distillation objective, Logit Difference Inhibition (LDI), which only penalizes the logit difference of a subset of classes with the highest logit values. On multiple image classification benchmarks, model updates with ELODI demonstrate superior accuracy retention and NFR reduction.

Index Terms—Positive-congruent training, cross-model compatibility, ensemble learning.

I. INTRODUCTION

The rapid development of visual recognition in recent years has led to the need for frequently updating existing models in production-scale systems. However, when replacing a legacy classification model, one has to weigh the benefit of decreased error rate against the risk of introducing new errors that may disrupt post-processing pipelines [1] or cause friction with human users [2]. To address this, we study Positive-Congruent Training (PC-Training) which refers to any training procedure that minimizes the negative flip rate (NFR) along with the error rate (ER) simultaneously.

Negative flips are instances that are misclassified by the new model but correctly classified by the old one. They are manifest in both visual and natural language tasks [1], [3]. They typically include not only samples close to the decision boundary, but also high-confidence mistakes that lead to perceived “regression” in performance compared to the old model. They are present even in identical architectures trained from different initial conditions, with different data augmentations, or using different sampling of mini-batches. Yan et al. [1] have shown that in state-of-the-art image classification models, where a 1% improvement is considered significant, NFR can be in the order of 4~5% even across models that have identical ER. These intriguing properties motivate us to investigate the causes of negative flips and the mechanism of reducing negative flips to establish a model update method that achieves cross-model compatibility, thus lowering NFR and ER, for better positive-congruent training.

Two questions: A naive approach to cross-model compatibility is to bias one model to mimic the other, as done in model distillation [4]. In this case, however, compatibility comes at the expense of accuracy [1], [2]. On the other hand, averaging a number of models in a deep ensemble [5] can reduce NFR without negative accuracy impact [1], even if it does not explicitly optimize NFR nor its surrogates. The role of ensembles in improving accuracy is widely known, but our first question arises: what is the role of ensembles in reducing NFR?

Even though using deep ensembles achieves state-of-the-art performance in terms of reducing NFR [1], it is nonviable in real applications at scale since it multiplying the cost of inference by an integer factor. Therefore, a second key question arises: Is it possible to achieve the PC-Training performance of ensembles at the inference cost of a single model?

Key ideas: To address the first key question above, we analyze the pattern of negative flip reduction in deep ensembles. We observe that deep ensembles reduce NFR by remedying potential flip samples that have relatively large variations in the logits space of different single models. When a deep ensemble is composed of member models with the same architecture but trained with independent initialization on the same dataset, which we denote as homogeneous ensembles, this behavior can be theoretically predicted and empirically validated.

As illustrated in Fig. 1(a), we independently train replicas of a single model with different random seeds to form the deep ensemble. We introduce a generalized distillation objective, Logit Difference Inhibition (LDI), which only penalizes the logit difference between the reference ensemble and the student single model on a subset of classes with the highest logit values. The result is what we call Ensemble Logit Difference Inhibition (ELODI).

Contributions: ELODI improves the state of the art in reducing perceived regression in model updates in three ways:
1) Generality, by not targeting distillation to a specific legacy model, yet reducing NFR; 2) Absence of collateral damage, by retaining the accuracy of a new model, or even improving it, while ensuring reduction of NFR; 3) Efficiency, as ELODI does not require evaluating ensembles of models at inference time. Moreover, ELODI is compatible with existing models trained without treatment. These improvements are made possible by two main contributions: 1) an analysis on deep ensembles which sheds light on their role in reducing NFR and the direction to obtain their performance for PC-training with single models; 2) ELODI, that integrates the NFR reduction of deep ensembles and running cost of single models by first training deep networks using the LDI loss with respect to an ensemble and then deploying the resulting single model at inference time. This results in a significant reduction of NFR (29% relative reduction on ImageNet for ResNet-18 → ResNet-50) over previous methods. As a side benefit, ELODI increases top-1 accuracy in several cases, and is comparable in others. Code is publicly available at https://github.com/amazon-science/regression-constraint-model-upgrade to facilitate future research in this emerging field.

II. RELATED WORK

Cross-model compatibility: is becoming increasingly important as real-world systems incorporate trained components that, if replaced, can wreak havoc with post-processing pipelines. Toneva et al. [6] empirically study prediction flip on training samples between epochs, termed “forgetting events”, while Yan et al. [1] address perceived regression using held-out sets between different models. Both are particular instances of cross-model compatibility [2], [7], [8]. Focal Distillation [1] minimizes the distance between the old and new predictions, with increased weights on samples correctly classified by the old model. Jiang et al. conduct experiments on more architectures and modalities including image, text, and tabular data in [9], [10] use a probabilistic approach to determine whether the prediction should update when a new model comes. While it improves cumulative NFR, it requires multiple models to be available at inference, which is prohibitive in practice.

Ensemble learning: methods [11], [12], [13] are widely adopted in machine learning. The understanding of these methods is sometimes explained as enlarging the margins [14]. Recently, the “multi-view” hypothesis [15] suggests that each independent model in an ensemble of deep networks learns a subset of feature views and memorizes data not separable using this subset. In practice, one can always boost the performance of a classifier by averaging multiple models that are trained separately under a certain level of variation in training including model type, training data, initialization, etc. In this paper, we take a different aspect of ensembling to reduce NFR, not to improve accuracy. We apply the ensemble as a teacher’s model to guide the student model in reducing negative flips during model updates. In particular, we present an alternative explanation from the perspective of representations’ dispersion in the logit space. ELODI can be thought of as variance reduction regularization in a Bayesian NN ensemble, which is replaced by its mean at inference time. The literature on variance reduction is too vast to survey here, but relevant references include [16], [17].

Some other ensemble learning techniques are summarized as follows: Deep ensemble [5] improves accuracy and allows estimating sample uncertainty; Snapshot Ensemble [18] and Fast Geometric Ensemble [19] train component models simultaneously, and Yan et al. [1] show that ensembles help reduce regression. Ensembles are impractical in most real applications due to the multiplier they impose on inference costs. This has prompted research on “implicit ensembles” such as Dropout [20] and its variants [21], DropPath [22] and Stochastic Depth [23].
Wen et al. propose BatchEnsemble [24] to generate ensemble weights, Havasi et al. use a MIMO [25] design to train multiple sub-networks concurrently. Different from all these methods, our method mainly focuses on reducing NFR instead of improving accuracy.

Knowledge distillation (KD) [4] was proposed to transfer “dark” knowledge from a larger “teacher” network to a smaller “student” by minimizing the distance between the distribution of predictions. In self-distillation [26], the teacher and student are the same. Focal Distillation [1] is a special case of KD with a sample-specific filtering function, developed for model updates where the legacy “teacher” model is actually weaker than the student (new) model, as in Reversed KD [27], where it is used as regularization. Ensemble distillation uses multiple teachers to improve accuracy in vision and other applications [28], [29], [30], [31], [32]. Our method is related to ensemble distillation while having two distinctive differences: 1) Our method uses a different term for the loss to achieve reduction of NFR; 2) ours uses a homogeneous ensemble whose members have the same architecture and are trained on the same dataset with different initialization seeds, unlike the traditional case that uses diverse models in the ensemble [33], which we call a heterogeneous ensemble.

III. REPRESENTATION LANDSCAPE OF ENSEMBLE-BASED PC-TRAINING

To answer the first key questions in Section I, we explore 1) how negative flips occur and 2) why ensembles yield fewer negative flips. To do so, we analyze the so-called logit space, where the representations are computed by a deep network before the softmax operation live. The reason we analyze logits rather than feature or softmax probabilities is as follows. Compared to the feature space, the logits of an arbitrary sample produced by different models trained on the same dataset live in the same vector space which is defined by the label set of the training samples. Compared to the space of post-softmax output, the logit distribution is easier to analyze because the softmax operation will skew the distribution. From a practical perspective, averaging in the logit space for ensembles is also common in recent works [34], [35].

A. Negative Flips and Logit Displacement Magnitude

Negative Flips: Given an input image $x$ with its label $\ell$ and a learned model $\phi$, let $\phi(x) \in \mathbb{R}^C$ denote its output logit vector before softmax, and $\phi_k(x)$ denote the $k$-th element, where $k \in \{1, \ldots, C\}$ and $C$ is the number of classes. The logit vector varies across models due to architecture, initialization, optimization method, and training dataset to name a few. For any model pair $\phi$ and $\psi$, where $\phi$ and $\psi$ take different forms due to architectural change in general, we define the logit displacement to be the difference between two output logits, i.e. $\phi(x) - \psi(x)$. Once the displacement is large enough to change the order of the top predictions, namely $\arg \max_k \phi_k(x) \neq \arg \max_k \psi_k(x)$, a flip occurs. We are particularly interested in negative flips, where we assume $\phi$ to be an old model and $\psi$ to be a new one and $\arg \max_k \phi_k(x) = \ell$ while $\arg \max_k \psi_k(x) \neq \ell$.

Homogeneous ensembles: Let $\mathbb{M}_1 = \{\ldots, \phi^{(i)} , \ldots\}$ and $\mathbb{M}_2 = \{\ldots, \psi^{(j)} , \ldots\}$ denote the set of member models from two homogeneous ensembles: Each $\phi^{(i)}$ has the same model architecture and is trained on the same dataset despite being independently initialized. So does each $\psi^{(j)}$, though $\phi$ and $\psi$ can have different architectures. For simplicity, we consider $\|\mathbb{M}_1\| = \|\mathbb{M}_2\| = m$, where $\|\cdot\|$ is the cardinality of each set.

Homogeneous ensembles reduce negative flips by reducing the magnitude of logit displacement: Given an input image $x$, $(\phi^{(i)}(x), \ldots, \phi^{(m)}(x))$ can be considered as $m$ i.i.d random variables drawn from a distribution approximated to second-order by an expectation $\mu_1$ and a co-variance matrix $\Sigma_1$, i.e. $\phi(x) \sim \mathcal{D}(\mu_1, \Sigma_1)$. Likewise we also have $\psi(x) \sim \mathcal{D}(\mu_2, \Sigma_2)$. The ensembles’ logit vectors are computed by averaging the individual models’ logits and denoted as $\phi^{(ens)} (x)$ and $\psi^{(ens)} (x)$.

The multi-dimensional central limit theorem [36], [37] states that this average converges in distribution to a multivariate normal distribution with the increase of $m$, i.e

$$\phi^{(ens)} (x) = \frac{1}{m} \sum_{i=1}^{m} \phi^{(i)}(x) \sim \mathcal{N}\left(\frac{1}{m} \mu, \frac{1}{m} \Sigma\right). \quad (1)$$

Therefore, the logit displacement between the two ensembles converges in distribution to another multivariate normal distribution, i.e

$$\phi^{(ens)} (x) - \psi^{(ens)} (x) = \frac{1}{m} \sum_{i \in \mathbb{M}_1} \phi^{(i)}(x) - \frac{1}{m} \sum_{j \in \mathbb{M}_2} \psi^{(j)}(x) \sim \mathcal{N}\left(\mu_1 - \mu_2, \frac{1}{m} \Sigma_1 + \frac{1}{m} \Sigma_2\right). \quad (2)$$

The norm of logit displacement will follow a generalized $\chi^2$ distribution [38], [39].

As a special case, if $\mathbb{M}_1$ and $\mathbb{M}_2$ have the same model architecture, then we have a normal distribution with zero mean and co-variance inversely scaled by the ensemble size:

$$\phi^{(ens)} (x) - \psi^{(ens)} (x) \sim \mathcal{N}\left(0, \frac{\Sigma_1 + \Sigma_2}{m}\right). \quad (3)$$

We connect the analysis to the observations shown in Fig. 2: 1) (4) implies that when ensembles become larger, the expectation of logit difference is zero and the covariance keeps decreasing, resulting in consistently decreasing NFR. This is consistent with the observation in [1], which is redrawn in the red curve, that two very large ensembles with the same architecture can have almost no flips. 2) In the case of two homogeneous ensembles with different architectures, $\mu_1$ and $\mu_2$ could have a non-zero difference which results in NFR not converging to zero. But the decrease of covariance part in (2) still contributes to a consistent non-trivial reduction in NFR. These explain the observation in the orange curve that NFR stagnates at a smaller non-zero value.

Heterogeneous ensembles show significantly higher variance on negative flips: Additionally, although we cannot conduct a similar analysis for a heterogeneous ensemble, we can empirically verify that the NFR between two heterogeneous ensembles
Approximating logit displacement norm with a few flipping-susceptible classes: In reality, the normal distribution in (2) and (4) is by no means isotropic especially when the underlying models \((\phi, \psi)\) learn meaningful representation. Conversely, as will be illustrated in Fig. 3(d) and Section III-B, this distribution is anisotropic enough such that the logit displacement norm can be approximated by the sum of logit difference at a few “outstanding” classes whose magnitudes are high:
\[
\|\phi^{(\text{ens})}(x) - \psi^{(\text{ens})}(x)\|_p = \sum_{k \in \mathbb{K}(x)} \left( \|\phi_k^{(\text{ens})}(x) - \psi_k^{(\text{ens})}(x)\|_p \right),
\]
where \(\mathbb{K}(x) \subset \{1, \ldots, C\}\) such that \(\forall j \in \mathbb{K}(x)\) and \(j' \neq 1, \ldots, C\setminus \mathbb{K}(x), \phi_j^{(\text{ens})}(x) \geq \phi_{j'}^{(\text{ens})}(x)\).

Furthermore, not coincidentally, this subset of classes with high logit values is more prone to prediction flipping. We take negative flipped samples between two ResNet-18 models and plot the rank of the wrongly predicted class’s logit among all logits in the other model, namely \(\text{Rank}(\psi_{\arg\max(\phi(x))}(x), \psi(x))\), in Fig. 4. We can see that these negative flips’ wrong prediction will also rank high, although not necessarily the topmost, in the other model that classifies it correctly. This phenomenon becomes more prominent if two models are ensemble. It motivates the formulation in (8).

B. A Two-Dimensional Example

To illustrate the behavior of models in logit space, we create a toy example by selecting two classes, i.e. “Labrador retriever” (n02099712) and “French bulldog” (n02108915), from ImageNet [40] and training several ResNet-18 models for binary classification. The models differ in their initialization, determined by distinct random seeds; we then collect output logits for each test datum and model in the ensemble. In Fig. 3(a), we plot the two-dimensional logit vectors of multiple data points when updating from one individual model to another. We can roughly categorize the negative flipped samples, highlighted with the purple arrows, into two types: 1) those close to the decision boundary in the old model; and 2) those far from the decision boundary in the old model but still flipped in the new one, due to significant displacement of the logit vector. Fig. 3(b) shows the logit vectors of the same set of data points but in the update case of two ensemble models each having 3 members (3×). Compared to Fig. 3(a), we can observe a clear reduction in the magnitude of displacement during the update. To validate that this observation is not incidental, we construct many cases of model updates and measure the distribution of the logit vector displacement on a certain data sample. As shown in Fig. 3(d), in updates between ensembles, the logit vectors are less likely to exhibit significant displacement. We repeat the same visualization on more data points in Section A to come. This suggests that the ensemble may be reducing the negative flip rate through the reduction of displacements of the logit vectors.

C. Validating the Hypothesis of Reducing Logit Displacement Magnitude

We validate our hypothesis on the representation landscape through large-scale experiments. Specifically, we train 256 ResNet-18 models on ImageNet with different seeds and split them into two halves. For an arbitrary image, we randomly draw \(m\) models without replacement in each half and compute the averaged logits of this drawn ensemble. We repeat the process and present in Fig. 5(a) the histogram of the logit displacement’s \(\ell_2\) norm between two random ensembles. Note that Seguin et al. [41] argue that the logit distribution is highly affected by the number of training epochs, therefore we follow the standard training recipe, which is detailed in Section V-A, for the assumption to hold.

We examine our hypothesis in Section III-A by comparing the histogram with the probability mass function (PMF) of the logit displacement norm. We start by using all available single models to estimate mean and co-variance \((\mu, \Sigma)\) for the logit vectors’ distribution \(\phi(x)\) and examine whether the norm of logit displacement will follow a generalized (central) \(\chi^2\) distribution. Since the probability density function (PDF) of a generalized \(\chi^2\) variable does not have a simple closed-form expression, we approximate it by Kernel Density Estimation (KDE) [42]. From Fig. 5(a), we see that the simulated PMFs in solid lines fit the histogram of single models well, implying that logits of these models could indeed follow a normal distribution. We conducted the same experiments above on more images in Section A and the conclusion holds well, suggesting this property is not incidental.

We also present a theoretical argument for the sample-wise output logit approximating a normal distribution. We assume the output logit \(y = \phi(x; w)\) to be locally linear to the parameters \(w\) given an arbitrary sample \(x\). Since \(w\) are initialized with a normal distribution, e.g Xavier [43] or Kaiming [44] initialization, \(\phi(x; w)\) will locally follow another normal distribution at initialization. The lazy-training theory [45] hypothesizes that a deep model behaves as its linearization around the initialization. If it holds, we can derive that \(\phi(x; \tilde{w})\), where \(\tilde{w}\) denotes the learned weights after training, again follows a normal distribution.
IV. Method: Ensemble Logit Difference Inhibition (ELODI)

The above analysis suggests the effectiveness of large homogeneous ensembles in reducing NFR, but an ensemble is less practical compared with a single model due to its multiplied inference cost to run every member model on a new input. In this work, we propose to re-purpose the knowledge distillation technique [4], which was previously used for improving model accuracy, for transferring the NFR reduction capability from ensembles to a single model.

A. Updates With Ensemble-Distilled Models

Given an ensemble $\phi^{(\text{ens})}$ composed of a set of models $\mathbb{M} = \{\phi^{(i)}\}_{i=1}^m$, we learn a single model $\hat{\phi}$ such that for each sample $x \in X$, the random variable of sample logits has ensemble-like reduced variance. We then use single models learned in this way in every model update of a system.

As illustrated in Fig. 1(a), when an old model $\phi^{(\text{old})}$ needs to be updated to a new model, we first train a homogeneous ensemble on the same dataset and having the same architecture as the desired new model. Next, we distill this reference ensemble to the actual new model $\phi^{(\text{new})}$. If the distillation process can convey the property of reducing NFR to the learned single model, updating from $\phi^{(\text{old})}$, which was preferably produced in the same manner, to $\phi^{(\text{new})}$ would result in significantly reduced NFR than that of a model pair without this treatment.

We empirically verify this model update method, called ELODI, in Section V-B and find it effective in reducing NFR while retaining the accuracy gain introduced by the new architecture used in the new model. It avoids the prohibitive inference cost of deep ensembles due to only using $\phi^{(\text{new})}$ in the model update, with reference ensemble discarded after the distillation. Another benefit of ELODI is that the new model does not need to target any specific existing model in an update, enabling a chain of models to yield relatively low NFR between any pair of them. This is rather helpful when multiple updates are consecutively executed, which is common for a long-running machine learning
Algorithm 1: Sequential Model Update With LODI.

Input: Dataset $\mathbb{X}$, number of versions to update $T$.
Output: A sequence of models $\phi^{(\text{Ver}-0)}$, $\ldots$, $\phi^{(\text{Ver}-T)}$.

while $t \leq T$ do
    for $i = 1$ to $m$ do
        $\phi^{(i,\text{Ver}-t)} \leftarrow \arg\min_{\phi^{(i,\text{Ver}-t)}} \sum_{x \in \mathbb{X}} LCE(\phi^{(\text{Ver}-t)})$
    end for
    for all $x \in \mathbb{X}$ do
        $\phi^{(\text{ens},\text{Ver}-t)}(x) \leftarrow \frac{1}{m} \sum_{i=1}^{m} \phi^{(i,\text{Ver}-t)}(x)$
    end for
    $\phi^{(\text{Ver}-t)} \leftarrow \arg\min_{\phi} \sum_{x \in \mathbb{X}} L_{\text{distill}}(\phi(x), \phi^{(\text{ens},\text{Ver}-t)}(x))$
    $t \leftarrow t + 1$
end while

The overall pipeline of a sequential model update via LODI is summarized in Algorithm 1.

B. Loss Choices of LODI

Generally we obtain the model to be deployed using the distillation technique [4],

$$\hat{\phi} = \arg\min_{\phi} \sum_{x \in \mathbb{X}} L_{\text{distill}}(\phi(x); \phi^{(\text{ens})}(x)).$$

Various types of distilling functions have been proposed to improve single-model accuracy: 1) Vanilla KD loss [4] minimizes the KL-divergence between two models’ output logits; 2) FitNet [46] and AttentionTransfer [47] mimics the intermediate hidden layer’s activation or attention; 3) FSP [48] mimics cross-layer Gram matrices.

We empirically find that applying exact logit matching, which minimizes the $\ell_p$-norm of logit difference from the single model to a reference ensemble model, achieves the goal of distilling reduced variance from an ensemble,

$$L_{\text{distill}}(x) = \sum_{k=1}^{C} \left( \| \phi_k(x) - \phi_k^{(\text{ens})}(x) \|_p \right)^p.$$  (7)

Furthermore, due to the fact stated in the last paragraph of Section III-A that the logit displacement magnitude is dominated by a few elements with a large difference in the high-dimensional case, we propose a top-$K$ variant of (7), called Logit Difference Inhibition (LDI) loss, which only inhibits logit difference on those classes with the top-$K$ highest logit elements,

$$L_{\text{LDI}}(x) = \sum_{k \in \mathbb{K}(x)} \left( \| \phi_k(x) - \phi_k^{(\text{ens})}(x) \|_p \right)^p,$$  (8)

where $\mathbb{K}(x)$ follows the definition in (5) or $\mathbb{K}(x) = \text{np}.\text{argsort}(-\phi_k(x)) [0:K]$ in pythonic pseudocode.

As will be shown in experiments, the Top-$K$ formulation leads to no loss in NFR reduction compared to the full form. It could instead help in reducing computation cost when the number of classes are extremely large [49]. Most importantly, it implies that LODI transfers the capability of NFR reduction in ensembles by reducing the variance of logit estimation indeed instead of exact logit matching through $\ell_p$ norm. $p$ is 2 in our experiments. $K$ is 10 by default according to Fig. 4.

The overall objective: of the distillation in LODI is a weighted sum of standard Cross-Entropy and the LDI loss, i.e $L = (1-\alpha)L_{\text{CE}} + \alpha L_{\text{LDI}}$, where the weight $\alpha$ is set such that the magnitude of $L_{\text{CE}}$ and $L_{\text{LDI}}$ is similar.

C. Integrating ELODI With Existing Models

Dealing with old models without ELODI is necessary when updating an existing system. We consider the simple case of one old model not trained with ELODI. In this case, we augment ELODI with an additional LDI loss w.r.t to the old model, i.e

$$L_{\text{total}} = \lambda L_{\text{LDI}}(M_{\text{new}}, M_{\text{ens}}) + (1-\lambda) L_{\text{LDI}}(M^\diamond_{\text{new}}, M_{\text{old}}),$$  (9)

where $M^\diamond$ denotes the model to be learned from the ensemble.
V. EXPERIMENTS

We conduct extensive experiments to showcase the effectiveness of the proposed ELODI from multiple aspects. In Section V-A, we describe the dataset statistics, evaluation metrics, and implementation details of training. In Section V-B, we show the main results of ELODI under the standard setting of updating a ResNet-18 model to a ResNet-50 model in comparison with other existing positive-congruent training (PC-Training) methods. In Section V-C, we demonstrate that ELODI is effective on more practical settings of model updates, including 1) the data-growth setting where more data is available for training the new model, 2) more than two rounds of model updates, 3) integrating ELODI with existing models, and 4) updating the old model to new models with various architectures. In Section V-D, we study the design choices in ELODI. Particularly, we show the advantages of using homogeneous ensembles as guidance instead of heterogeneous ones in ELODI to support the analysis in Section III-A. In Section V-E, we provide some exploratory studies on the parameters in ELODI.

A. Experimental Setup

Datasets: We validate the proposed approaches on two standard image classification datasets: ImageNet [40] and iNaturalist [50]. For ImageNet, we use ILSVRC12 [51] which contains 1,000 categories. It has around 1.2 million training images and 50,000 validation images. For iNaturalist, we use the version released in 2017. It covers 5,089 categories of fine-grained species, with 579,184 training images and 95,986 validation images. Besides, to showcase the efficacy of ELODI on more data modalities, we also conduct experiments on the AG News Corpus [52], a text classification dataset which contains 4 classes (“World”, “Sports”, “Business”, “Sci/Tech”) of news articles. It has 30,000 training and 1,900 test samples for each class.

Metrics: In a model update experiment, we measure the top-1 error rate (ER) of both the old and new models.

\[
ER_{\text{old}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\hat{y}_i^{\text{(old)}} \neq \ell_i),
\]

\[
ER_{\text{new}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\hat{y}_i^{\text{(new)}} \neq \ell_i).
\]

In addition, we report the negative flip rate with respect to the old model (NFR) [1], which is defined as

\[
\text{NFR} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\hat{y}_i^{\text{(new)}} \neq \ell_i, \hat{y}_i^{\text{(old)}} = \ell_i),
\]

where \(\mathbb{I}(\cdot)\) is the indicator function, \(\ell_i\) is the label, and \(\hat{y}_i^{\text{(new)}}\) and \(\hat{y}_i^{\text{(old)}}\) are the new (old) model’s prediction.

Since the absolute value of NFR is upper bounded by the error rate, comparing methods with different accuracies is hard. To this end, we also report Relative NFR (Rel-NFR) [1], which is defined as

\[
\text{Rel-NFR} = \frac{\text{NFR}}{1 - ER_{\text{old}}} \times ER_{\text{new}},
\]

where the denominator is the expected error rate on the subset of samples predicted correctly by the old model. Relative NFR factors out overall model accuracies.

Implementation Details: We follow the standard training recipe of ImageNet. Specifically, all classification models are trained with SGD with a momentum of 0.9 and a base learning rate of 0.1, which is reduced by a tenth every 30 epochs until 90 epochs. The batch size is 256 with 8 GPUs.

For ELODI with larger models and ensemble size, GPU memory becomes a bottleneck. To handle the memory issue, we use gradient checkpointing [53] and reduce batch size while linearly scaling the base learning rate [54].

Unless otherwise specified, ELODI experiments are done with ensemble size \(m = 8\).

B. Main Results of ELODI

Comparison with other PC-Training Methods: We first compare ELODI with other PC-Training methods under the setting of updating ResNet-18 to ResNet-50 in Table I. “No treatment” means that both models are trained with standard cross entropy loss. Previous methods include (a) Backward-Compatible Training (BCT) [7], (b) Regression-Alleviating Compatible Training (RACT) [55], (c) Knowledge Distillation (KD) [4], (d) Focal Distillation with either KL divergence or Logit Matching as the objective (FD-KL/LM) [1], and (e) Bayesian Update (BU) [10].

BCT and RACT belong to compatible-training-based methods while KD and FD-KL/LM are distillation-based methods. We can see that ELODI outperforms previous methods in terms of both absolute and relative NFR. Note that these baselines are all targeted model update, meaning that the old ResNet-18 is used as the target when training ResNet-50. In contrast, ELODI does not target any legacy model.

Table I

| Method               | ER\(_{\text{RN-18}}\) (%) | ER\(_{\text{RN-50}}\) (%) | NFR\(_{\text{RN-18}}\) (%) | Rel-NFR\(_{\text{RN-18}}\) (%) |
|---------------------|---------------------------|---------------------------|---------------------------|-------------------------------|
| No treatment (single) | 30.24                     | 24.66                     | 4.30                      | 25.00                         |
| Ensemble Paragon (8×) | 26.34*                    | 22.44*                    | 1.95                      | 11.80                         |
| Dropout [20]         | 30.97                     | 24.50                     | 4.08                      | 21.92                         |

(LODI: ENSEMBLE LOGIT DIFFERENCE INHIBITION FOR POSITIVE-CONGRUENT TRAINING 7535)

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Since Dropout [20] implicitly makes a network behave similar to an ensemble [21], we also report its NFR performance using a dropout probability of \( p = 0.2 \). The NFR is still relatively high (4.08 v.s 4.30%) partially due to the increased ER of both ResNet-18 and ResNet-50. Though the relative NFR is smaller than the non-treatment baseline (21.92 v.s 25.00%), it is still significantly greater than ELODI (13.23%). One reason is that the ensemble property of dropout is “weak”. The other might be a considerable correlation among the hypothetical “ensemble members” in the dropout setting due to the shared internal representations hindering the reduction of NFR as ensembles.

Finally, we study a few variants of ELODI. Using the Top-\( K \) highest-logit class subset in ELODI with \( K = 10 \) even slightly outperforms distilling all classes of logits by 0.07%. From an ER-NFR scatter plot in Fig. 1(b), ELODI achieves a similar level of ER-NFR results as the ensemble paragon [1] at the inference cost of a single model.

C. More Update Settings of ELODI

In this subsection, we move beyond the standard comparison with other PC-Training methods and look at a few practical scenarios where model updates occur. First, we study the case of model updates when the training data grows in either the number of classes or the number of per-classes samples. Second, we show the applicability of ELODI on other datasets. Third, we demonstrate that ELODI enables multiple rounds of model updates. Fourth, we showcase that ELODI is seamlessly integrated with existing models that have been deployed without positive-congruent training. Finally, we reveal that the NFR reduction ability of ELODI is agnostic to architectures.

Data-growth settings: Model updates may also come with growing training data, e.g., 1) increasing number of classes, and 2) increasing number of per-classes samples. We follow the same data/class split in [1], which uses 50% classes/samples for the old model and full data for the new. From Table III, we find that conclusions from the full-data setting hold for these settings. In particular, ELODI achieves a significantly lower error rate compared to Focal Distillation (FD) because ELODI is not targeted at the old model whose performance is typically much worse due to less training data.

Fine-tuning on other datasets: We validate the effectiveness of ELODI when transferring to a different dataset. We examine iNaturalist [50] and follow the protocol in [1]. Results of both full-data and data-growth settings are given in Table II, where ELODI consistently outperforms FD.

| Method | | | | Full data | | |
|--------|--------|--------|--------|--------|--------|--------|--------|
|        | ER\(_\%\) | NFR\(_\%\) | ER\(_\%\) | NFR\(_\%\) | ER\(_\%\) | NFR\(_\%\) | ER\(_\%\) | NFR\(_\%\) |
| No treatment | 47.58 | 33.95 | 5.38 | 78.88 | 35.95 | 3.82 | 40.69 | 33.95 | 4.76 |
| Focal Distillation [1] | 45.87 | 3.44 | 66.91 | 2.00 | 40.03 | 3.95 |
| Ensemble Paragon (8×) | 36.94* | 29.47* | 52.96 | 34.29 | 1.47 | 40.37 | 34.29 | 2.46 |
| ELODI | 36.94* | 29.47* | 52.96 | 34.29 | 1.47 | 40.37 | 34.29 | 2.46 |

*In ensemble paragon means that the number is from a collection of models. *\(^{(*)}\)* means that the model is trained on half (full) data. ELODI is effective when fine-tuning on Naturalist under multiple settings.

**ELODI on more data modalities:** To show the efficacy of ELODI on other modalities, we conduct a set of text classification experiments on the AG News corpus. Particularly, we consider the model update between two BERT-base models [56]. Both old and new models take the pre-trained weights and are then fine-tuned on the AG News corpus, following [57]. The results are shown in Table IV. We can see that ELODI reduces both absolute and relative NFR by almost 70% compared to the no-treatment baseline.

**ELODI on a chain of model updates:** As discussed in Section IV-A, ELODI does not involve the guiding ensembles at the inference stage. Nor does it target any legacy model. Only the single model trained with ELODI is deployed to replace the old model, which is also trained with ELODI. When the number of updates increases, this naturally forms a chain of models having low NFR between them, inducing a transitive reduced NFR.

We illustrate this transitivity of NFR reduction in a chain of updates of three models, i.e., ResNet-18→ResNet-50→ResNet-101. In Table V, with ELODI, NFRs between the three models reduce to 2.04%~2.25% from 3.92%~4.41% (a relative reduction of 44.1%~52.3%), outperforming all previous methods, including three variants of FD [1]: 1) chain, where each model targets at its closest predecessor; 2) radial, where each model

**TABLE II**

Elodi on Inaturalist

| Method | Increasing #classes | Increasing #samples/class | Full data |
|--------|---------------------|---------------------------|-----------|
|        | ER\(_\%\) | NFR\(_\%\) | ER\(_\%\) | NFR\(_\%\) | ER\(_\%\) | NFR\(_\%\) |
| No treatment | 47.58 | 33.95 | 5.38 | 78.88 | 35.95 | 3.82 | 40.69 | 33.95 | 4.76 |
| Focal Distillation [1] | 45.87 | 3.44 | 66.91 | 2.00 | 40.03 | 3.95 |
| Ensemble Paragon (8×) | 36.94* | 29.47* | 52.96 | 34.29 | 1.47 | 40.37 | 34.29 | 2.46 |
| ELODI | 36.94* | 29.47* | 52.96 | 34.29 | 1.47 | 40.37 | 34.29 | 2.46 |

*In ensemble paragon means that the number is from a collection of models. *\(^{(*)}\)* means that the model is trained on half (full) data. ELODI is effective when fine-tuning on Naturalist under multiple settings.

**TABLE III**

Elodi in Data-Growth Settings on Imagenet

| Method | Increasing #classes | Increasing #samples/class | Full data |
|--------|---------------------|---------------------------|-----------|
|        | ER\(_\%\) | NFR\(_\%\) | ER\(_\%\) | NFR\(_\%\) | ER\(_\%\) | NFR\(_\%\) |
| No treatment | 22.02 | 24.66 | 14.07 | 34.26 | 24.66 | 3.52 |
| Focal Distillation [1] | 18.70* | 22.44* | 10.19 | 28.16* | 22.44* | 2.11 |
| Ensemble Paragon (8×) | 21.80 | 23.15 | 4.19 | 34.08 | 23.15 | 2.25 |
| ELODI (K = 10) | 21.24 | 23.10 | 4.10 | 33.93 | 23.10 | 2.21 |

*In ensemble paragon means that the number is from a collection of models. *\(^{(*)}\)* means that the model is trained on half (full) data.

**TABLE IV**

Elodi on textual data

| Method | ER\(_\%\) | NFR\(_\%\) | Rel-NFR\(_\%\) |
|--------|--------|--------|---------------|
| No treatment (single) | 5.95 | 5.95 | 0.934 | 16.69 |
| Ensemble Paragon (8×) | 5.82* | 5.62* | 0.263* | 4.97* |
| ELODI | 5.90 | 5.89 | 0.289 | 5.21 |

On the AG News Corpus, ELODI reduces the relative NFR between two text classification models by 68.8% compared to the no-treatment baseline. The performance is close to the ensemble paragon, where all numbers are obtained from a collection of models. Here, both the old and new models follow the BERT-base architecture [56].
Homogeneous vs heterogeneous ensembles: We use a homogeneous ensemble in both Sections III-A and IV-A. However, in ensemble learning, members with strong diversity such as model architectures are usually favored for better generalization. To study the potential advantage of homogeneous ensembles in ELODI over heterogeneous ones, we construct an all-diff-weak ensemble, which is composed of 8 different weak models with Top-1 Accuracy $= 69:\sim 70\%$ on ImageNet. Similarly, we build an all-diff-strong ensemble, which is composed of 8 different strong models with Top-1 Accuracy $= 75:\sim 76\%$ on ImageNet. The model list can be found in Table IX and the model weights are adopted from timm [68].

In Table VIII, we observe that using a homogeneous ensemble for guidance achieves comparable or slightly better results in both NFR and ER than heterogeneous (“all-different”) ensembles. Some may argue that the confidence score might vary across different architectures. To rule out potential issues on miscalibration, we also additionally calibrate “all-different” ensembles using temporal scaling [58] but observe no gain. This suggests that strong diversity in a guiding ensemble may not lead to lower NFR. Also, ELODI with homogeneous ensembles is easier to implement and extend in practice - a homogeneous ensemble requires only one architecture and needs neither recalibrating prediction score nor balancing weights.

Change of architecture for the guiding ensemble: In Table VIII, we find that training a new model guided by an ensemble with the old model’s architecture has to trade ER for NFR reduction, which is not desired. This corroborates with the hypothesis in Section III-A that models with different architectures have different representation landscapes and thus it is better to use an ensemble with the same architecture for guiding ELODI. When a system has undergone multiple updates, always guiding ELODI with the new model’s architecture also provides a clear guideline for practice.

Choices of distillation functions: We compare different choices of distillation loss functions in Table X. We can see that ELODI outperforms FD [1] (Ensemble w/ KD) in both ER and NFR.

E. Exploratory Studies of Parameters in ELODI

In this subsection, we provide more detailed ablation on the hyper-parameters in ELODI, namely the loss weight and size of the reference ensemble. We end this subsection with a comparison between offline and online distillation.

The effect of loss weight: We experiment with different loss weights $\alpha$ and summarize the results in Fig. 6(a). $\alpha_{ELODI} = 0$ is equivalent to the no-treatment baseline where only the standard Cross-Entropy loss is applied. $\alpha_{ELODI} = 1$ means that we drop the Cross-Entropy loss and distill logits only. When $\alpha_{ELODI}$ increases from 0.5 to 1, the distilled model’s ER first decreases and then increases for both models. On the other hand, NFR consistently decreases and stays at around 2.2%. We find $\alpha_{ELODI} = 0.8$ achieves a good balance between the distilled model’s ER and NFR. Therefore we use it by default for all ELODI experiments.

The size of reference ensemble: We study ELODI’s effectiveness for reducing NFR by varying the ensemble size $m$ in
TABLE VI

| LDI with Different Architectures on ImageNet |
|--------------------------------------------|
| ER\(_1\) (%) | ER\(_1\) (%) | NFR\(_1\) (%) |
|--------------|--------------|---------------|
| **None** (single) | 30.24 | 24.66 | 3.64 |
| **Ensemble Paragon (8 x)** | 26.34* | 20.05* | 1.72 |
| **ELODI** | 31.34 | 21.09 | 2.19 |

ELODI effectively reduces NFR on a wide range of architectures. *In ensemble paragon means that the number is from a collection of models. † Is obtained by our reproduction with different augmentation and training schedules from the official one. Note that all new models’ NFR is measured w.r.t ResNet-18 listed in the leftmost column.

TABLE VII

| Integrating LDI with Existing Models |
|--------------------------------------|
| LDI usage | Pairwise NFR |
|-----------|--------------|
| **None** | **RN-18** | 2.56% |
| **Once** | **RN-18** | 2.56% |
| **Both** | **RN-18** | 2.14% |

ELODI can reduce NFR w.r.t old models that have been deployed without ELODI by being jointly optimized with targeted distillation (denoted by \(M_1 \rightarrow M_2\)).

TABLE VIII

| ELDI with Different Guiding Ensembles |
|----------------------------------------|
| Old Reference | New Reference | Error Rate\(_1\) (%) | NFR\(_1\) (%) |
|---------------|---------------|----------------------|--------------|
| **N/A** | **N/A** | 30.24 | 24.66 |
| All-diff-weak | All-diff-weak | 32.38 | 26.11 |
| Mixed-weak | Mixed-weak | 32.75 | 26.88 |
| RN-18 (x8) | RN-18 (x8) | 31.32 | 26.82 |
| All-diff-weak | All-diff-strong | 32.73 | 23.14 |
| All-diff-weak | All-diff-strong | 33.38 | 23.33 |
| Mixed-weak | Mixed-weak | 32.75 | 23.68 |
| RN-18 (x8) | RN-50 (x8) | 31.32 | 23.15 |

We consider ResNet-18 \(\rightarrow\) ResNet-50 via ELODI with an 8 x -model ensemble. All-diff-weak -strong ensemble is composed of 8 different weak (strong) models with Top-1 Acc. \(\approx 69\% (75\%)\) on ImageNet. The model list can be found in Table 9. Mixed-weak -strong ensemble is a mixture of 4 x ResNet-18 and 4 x VGG-13 (4 x ResNet-50 and 4 x DenseNet-121). † means that the models in each ensemble are calibrated [58].

TABLE IX

| Details of Heterogeneous Ensembles in Fig. 2 and Table VIII |
|-----------------------------------------------------------|
| Weak models (69–70%) | ResNet-18 [59], GoogleNet [60], VGG-11, VGG-13, VGG-11-BN, VGG-16 [61], HRNet-W18 [62], DLA [63] |
| Strong models (75–76%) | ResNet-50 [59], DenseNet-121 [64], Inception-V3 [65], VGG-19-BN [61], RegNetY [66], RepVGG-A2 [67], DPN-68 [53], DLA-X-60-C [63] |

Fig. 6(b). The case of \(m = 0\) is the no-treatment baseline. The case of \(m = 1\) can be viewed as self-distillation [26] except that the new model’s weight is re-initialized. We can see that NFR consistently decreases from 4.30% to 2.15% when the ensemble size increases from 1 to 8.

TABLE X

| Distilling Ensembles with Different Loss Functions |
|---------------------------------------------------|
| Method | ER\(_1\) (%) | NFR\(_1\) (%) |
|---------|--------------|---------------|
| **ELODI** | 30.95 | 23.10 |
| Ensemble w/ \(K_D = 100\) | 32.09 | 23.67 |
| Ensemble w/ \(F_D = 100\) | 32.19 | 23.97 |
| Ensemble w/ \(F_D = 1\) | 31.62 | 24.06 |

Considering the model update of ResNet-18 \(\rightarrow\) ResNet-50, ELODI achieves lower NFR and ER than KD/FD.

TABLE XI

| Comparison between Offline and Online Distill on ImageNet |
|------------------------------------------------------------|
| Method | Error Rate\(_1\) (%) | NFR\(_1\) (%) |
|---------|----------------------|---------------|
| Offline | 32.49 | 24.26 |
| Online | 30.97 | 23.81 |

Inferring teacher logits during training (online) achieves both lower ER and NFR compared to pre-extracting it (offline).

Online v.s offline distillation: In ELODI, the ensemble’s logits can be either inferred during training (online) or pre-extracted before training (offline). In Table XI, we find that offline distillation is less effective in reducing NFR and ER. A similar observation is also reported in [69]. Therefore we use the online approach in all experiments.

VI. Conclusion

Our experiments show that ELODI performs positive congruent training by reducing negative flips with large logit
displacement and reducing the variance of logits from the ensemble estimates. This behavior can be transferred to single models through the ELODI method can benefit updates with single models.

As discussed in Section III-A and later observed in experiments, the difference in representation landscape could still lead to non-zero NFR in the updates even with ELODI, which requires future works on in-depth characterizing of representation landscape change between model architectures. Another limitation of ELODI is that the training cost is still higher than the normal training process of a classification model update, due to the additional training of the ensemble and online inference of the ensemble logits, calling for further efficiency improvement.

REFERENCES

[1] S. Yan et al., “Positive-congruent training: Towards regression-free model updates,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 14294–14303.

[2] G. Bansal, B. Nushi, E. Kamar, D. S. Weld, W. S. Lasecki, and E. Horvitz, “Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff,” in Proc. AAAI Conf. Artif. Intell., 2019, pp. 2429–2437.

[3] Y. Xie, Y.-A. Lai, Y. Xiong, Y. Zhang, and S. Soatto, “Regression bugs are in your model! Measuring, reducing and analyzing regressions in NLP model updates,” in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2021, pp. 6589–6602.

[4] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” 2015, arXiv:1503.02531.

[5] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2017, pp. 6405–6416.

[6] S. Toseva, A. Sotiriou, R. T. D. Combes, A. Trischler, Y. Bengio, and G. J. Gordon, “An empirical study of example forgetting during deep neural network learning,” in Proc. Int. Conf. Learn. Representations, 2019.

[7] Y. Shen, Y. Xiong, W. Xia, and S. Soatto, “Towards backward-compatible representation learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6367–6376.

[8] M. Srivastava, B. Nushi, E. Kamar, S. Shah, and E. Horvitz, “An empirical analysis of backward compatibility in machine learning systems,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2020, pp. 3272–3280.

[9] H. Jiang, H. Narasimhan, D. Bahri, A. Cotter, and A. Rostamizadeh, “Churn reduction via distillation,” in Proc. Int. Conf. Learn. Representations, 2022.

[10] F. Träuble, J. von Kügelgen, M. Kleindessner, F. Locatello, B. Schölkopf, and P. Gehler, “Backward-compatible prediction updates: A probabilistic approach,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2021, vol. 34, pp. 116–128.

[11] L. Breiman, “Bagging predictors,” Mach. Learn., vol. 24, no. 2, pp. 123–140, 1996.

[12] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” J. Comput. System Sci., vol. 55, no. 1, pp. 119–139, 1997.

[13] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[14] P. Bartlett, Y. Freund, W. S. Lee, and R. E. Schapire, “Boosting the margin: A new explanation for the effectiveness of voting methods,” Ann. Statist., vol. 26, no. 5, pp. 1651–1686, 1998.

[15] Z. Allen-Zhu and Y. Li, “On domains of attraction of multi-dimensional distributions,” in Proc. Int. Conf. Learn. Representations, 2020.

[16] E. Rvaˇcek, “On domains of attraction of multi-dimensional distributions,” Sel. Translations Math. Statist. Probability, vol. 2, pp. 183–205, 1962.

[17] T. S. Ferguson, “A Course in Large Sample Theory. London, UK: Routledge, 2017.

[18] A. M. Mathai and S. B. Provost, Quadratic Forms in Random Variables: Theory and Applications. New York, NY: Dekker, 1992.

[19] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, “Loss surfaces, mode connectivity, and fast ensembling of DNNs,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2018, vol. 31, pp. 8803–8812.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” J. Mach. Learn. Res., vol. 15, pp. 1929–195, 2014.

[21] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in Proc. Int. Conf. Mach. Learn., 2016, pp. 1050–1059.

[22] G. Larsson, M. Maire, and G. Shakhnarovich, “FractalNet: Ultra-deep neural networks without residuals,” in Proc. Int. Conf. Learn. Representations, 2017.

[23] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, “Deep networks with stochastic depth,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 646–661.

[24] Y. Wei, D. Tran, and J. Ba, “Batchensemble: An alternative approach to efficient ensemble and lifelong learning,” in Proc. Int. Conf. Learn. Representations, 2020.

[25] M. Havasi et al., “Training independent subnetworks for robust prediction,” in Proc. Int. Conf. Learn. Representations, 2021.

[26] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, “Be your own teacher: Improve the performance of convolutional neural networks via self distillation,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 3712–3721.

[27] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng, “Revisiting knowledge distillation via label smoothing regularization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 3902–3910.

[28] S. Reich, D. Mueller, and N. Andrews, “Ensemble distillation for structured prediction: Calibrated, accurate, fast-choose three,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2020, pp. 5583–5595.

[29] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran, “Efficient knowledge distillation from an ensemble of teachers,” in Proc. Interspeech, 2017, pp. 3697–3701.

[30] U. Asif, J. Tang, and S. Harer, “Ensemble knowledge distillation for learning improved and efficient networks,” in Proc. Eur. Conf. Artif. Intell., 2020, pp. 953–960.

[31] A. Malinin, B. Mlodozeniec, and M. Gales, “Ensemble distillation distribution,” in Proc. Int. Conf. Learn. Representations, 2020.

[32] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, “Ensemble distillation for robust model fusion in federated learning,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, vol. 33, pp. 2351–2363.

[33] L. I. Kuncheva and C. J. Whitaker, “Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy,” Mach. Learn., vol. 51, no. 2, pp. 181–207, 2003.

[34] R. Gontijo-Lopes, Y. Dauphin, and E. D. Cubuk, “No one representation to rule them all: Overlapping features of training methods,” in Proc. Int. Conf. Learn. Representations, 2022.

[35] M. Wortsman et al., “Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time,” in Proc. Int. Conf. Mach. Learn., 2022, pp. 23965–23998.

[36] E. Rvaˇcek, “On domains of attraction of multi-dimensional distributions,” Sel. Translations Math. Statist. Probability, vol. 2, pp. 183–205, 1962.

[37] T. S. Ferguson, A Course in Large Sample Theory. London, UK: Routledge, 2017.

[38] A. M. Mathai and S. B. Provost, Quadratic Forms in Random Variables: Theory and Applications. New Britain, CT, USA: Dekker, 1992.

[39] A. Das and W. S. Geisler, “A method to integrate and classify normal tuned models improves accuracy without increasing inference time,” in Proc. Int. Conf. Learn. Representations, 2020, pp. 6466–6474.

[40] E. Parzen, “On estimation of a probability density function and mode,” Ann. Math. Statist., vol. 33, pp. 1065–1076, 1962.

[41] L. Seguin, A. Ndirango, N. Mishra, S. Chung, and T. Lee, “Understanding robust model fusion in federated learning,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, vol. 33, pp. 2351–2363.

[42] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proc. Eur. Conf. Comput. Vis., 2010, pp. 249–256.
S. Zagoruyko and N. Komodakis, “Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer,” in Proc. Int. Conf. Learn. Representations, 2017.

J. Yim, D. Joo, J. Baek, and J. Kim, “A gift from knowledge distillation: Fast optimization, network minimization and transfer learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 7130–7138.

X. An et al., “Partial FC: Training 10 million identities on a single machine,” 2020, arXiv: 2010.05222.

G. Van Horn et al., “The iNaturalist species classification and detection dataset,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8769–8778.

O. Russakovsky et al., “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vis., vol. 115, pp. 211–252, 2015.

X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2015, pp. 649–657.

T. Chen, B. Xu, C. Zhang, and C. Guestrin, “Training deep nets with sublinear memory cost,” 2016, arXiv:1604.06174.

P. Goyal et al., “Accurate, large minibatch SGD: Training imageNet in 1 hour,” 2017, arXiv: 1706.02677.

B. Zhang et al., “Hot-refresh model upgrades with regression-free compatible training in image retrieval,” in Proc. Int. Conf. Learn. Representations, 2022.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, 2018, pp. 4171–4186.

C. Sun, X. Qiu, Y. Xu, and X. Huang, “How to fine-tune BERT for text classification?,” in Proc. China Nat. Conf. Chinese Comput. Linguistics, 2019, pp. 194–206.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 1321–1330.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

C. Szegedy et al., “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 1–9.

K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

J. Wang et al., “Deep high-resolution representation learning for visual recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

F. Yu, D. Wang, E. Shelhamer, and T. Darrell, “Deep layer aggregation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 2403–2412.

G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2261–2269.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2818–2826.

I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, “Designing network design spaces,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 10425–10433.

X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, “RepVGG: Making VGG-style convnets great again,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 13728–13737.

R. Wightman, “PyTorch image models,” 2019. [Online]. Available: https://github.com/rwightman/pytorch-image-models

L. Beyer, X. Zhai, A. Royer, L. Markeeva, R. Anil, and A. Kolesnikov, “Knowledge distillation: A good teacher is patient and consistent,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 10915–10924.

Yue Zhao received the BE degree from Tsinghua University, and the MPhil degree from the Chinese University of Hong Kong, supervised by Prof. Dahua Lin. He is currently working toward the PhD degree with the University of Texas at Austin, supervised by Prof. Philipp Krähenbühl. His research interests include computer vision, particularly video understanding. He is a recipient of the NVIDIA Graduate Fellowship.

Yuantjun Xiong received the BS degree from Tsinghua University, Beijing, China, in 2012, and the PhD degree in information engineering from the Chinese University of Hong Kong, in 2016. He is a principal applied scientist with Amazon Rekognition. Before joining Amazon, he was a postdoctoral fellow with the Department of Information Engineering, CUHK. His research interests include computer vision, image understanding, and video content analysis.

Shuo Yang received BEng degree from the School of Software, Wuhan University, in 2013, and the PhD degree from the Department of Information Engineering, CUHK, in 2017. He is currently a senior applied scientist with AWS-AI Computer Vision. He is supervised by Prof. Xiaou Tang and Prof. Chen Change Loy. His research interests include face/object detection and object recognition.

Wei Xia received the PhD degree in computer vision and machine learning from the National University of Singapore and performed as a visiting researcher with Panasonic and Lund University. He is a principal scientist with AWS AI, leading the research for AWS ReKognition. Previously, he was the chief scientist with Orbeus Inc, an AI startup acquired by Amazon in 2015, respectively. He has published around 40 papers and filed more than 30 patents. He also served as a reviewer and workshop organizer in top-tiered conferences and journals, like ICCV/CVPR/ECCV, etc. During his years in AWS, he and his team started a new research area in model compatibility. He has won the winner awards of Pascal VOC classification and segmentation competitions in 2012, runner-up award in ILSVRC challenge 2013 and winner award of ILSVRC detection challenge in 2014.

Yantao Shen received the BE degree from North-eastern University, Shenyang, China, in 2015, and the PhD degree in electronic engineering from the Chinese University of Hong Kong, in 2020. He is an applied scientist with AWS AI.

Zhuowen Tu (Fellow, IEEE) received the ME degree from Tsinghua University, and the PhD degree from Ohio State University. He is a full professor with Cognitive Science and also affiliated with the Department of Computer Science and Engineering, University of California San Diego. Before joining UCSD in 2013 as an assistant professor, he was a faculty member with UCLA. Between 2011 and 2013, he took a leave to work with Microsoft Research Asia. He is a recipient of the David Marr Prize Award 2003 and a recipient of the David Marr Prize Honorable Mention Award 2015.
Bernt Schiele (Fellow, IEEE) received the PhD degree from INP Grenoble, France, in 1997, under the supervision of Prof. James L. Crowley in the field of computer vision. He has been Max Planck director with MPI for Informatics and Professor at Saarland University since 2010. He studied computer science with the University of Karlsruhe, Germany. He worked on his master thesis in the field of robotics in Grenoble, France, where he also obtained the “diplome d’études approfondies dinformatique”. In 1994 he worked in the field of multi-modal human-computer interfaces at Carnegie Mellon University, Pittsburgh, PA, USA in the group of Alex Waibel. The title of his thesis was “Object Recognition using Multidimensional Receptive Field Histograms”. Between 1997 and 2000, he was postdoctoral associate and visiting assistant professor with the group of Prof. Alex Pentland with the Media Laboratory of the Massachusetts Institute of Technology, Cambridge, MA, USA. From 1999 until 2004 he was assistant professor with the Swiss Federal Institute of Technology in Zurich (ETH Zurich). Between 2004 and 2010 he was full professor with the computer science Department of TU Darmstadt.

Stefano Soatto (Fellow, IEEE) is professor of computer science with the University of California, Los Angeles (UCLA), where he is also professor of electrical engineering and founding director with the UCLA Vision Lab. He is also a VP in AWS AI. He was named fellow with the Institute of Electrical and Electronics Engineers (IEEE), in 2013 for contributions to dynamic visual processes. He received the David Marr Prize in Computer Vision, in 1999.