ELODI: ENSEMBLE LOGIT DIFFERENCE INHIBITION FOR POSITIVE-CONGRUENT TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Negative flips are errors introduced in a classification system when a legacy model is replaced with a new one. Existing methods to reduce negative flip rate (NFR) either do so at the expense of overall accuracy using model distillation, or use ensembles, which multiply inference cost prohibitively. We present a method to train a classification system that achieves paragon performance in both error rate and NFR, at the inference cost of a single model. Our method introduces a generalized distillation objective, Logit Difference Inhibition (LDI), that penalizes changes in the logits between the new and old model, without forcing them to coincide as in ordinary distillation. LDI affords the model flexibility to reduce error rate along with NFR. The method uses a homogeneous ensemble as the reference model for LDI, hence the name *Ensemble LDI*, or ELODI. The reference model can then be substituted with a single model at inference time. The method leverages the observation that negative flips are typically not close to the decision boundary, but often exhibit large deviations in the distance among their logits, which are reduced by ELODI.

1 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 (Yan et al., 2021) or cause friction with human users (Bansal et al., 2019). Positive-Congruent Training (PC-Training) refers to any training procedure that minimizes the *negative flip rate* (NFR) along with the error rate (ER). 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 (Yan et al., 2021; Xie et al., 2021). They are typically *not* close to the decision boundary, but instead new high-confidence mistakes that lead to *perceived "regression"* in performance compared to the old model.

Aside from compatibility with humans or across models with different architectures, inconsistencies are present even in identical architectures trained from different initial conditions, or with different data augmentations, or using different sampling of mini-batches. Yan et al. (2021) 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$\sim$5% even across models that have identical ER.

Two questions. A naive approach to cross-model compatibility is to bias one model to mimic the other, as done in model distillation (Hinton et al., 2015). In this case, however, compatibility comes at the expense of accuracy, even when distillation is tailored to PC-Training through Focal Distillation (FD) (Yan et al., 2021). But since NFR can be reduced at equal error rate by trading errors (Yan et al., 2021), a first question arises: *Is it possible to reduce NFR to the level of Focal Distillation without any impact on Error Rate*?

Another approach to PC-Training consists of averaging a number of models in a deep ensemble (Lakshminarayanan et al., 2017). It is puzzling that this method would achieve state-of-the-art performance, since it does not explicitly optimize NFR, nor any surrogates, as part of the training process. More importantly, ensembling is not viable in real applications since it multiples 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 cost of running a single model?*

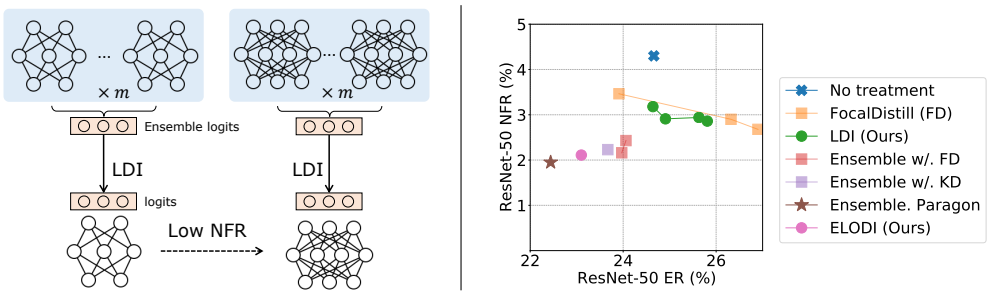


Figure 1: **Left:** In ELODI, one model is trained using the Logit Difference Inhibition (LDI) loss *w.r.t* an ensemble of m models with its same architecture. The result is a single model which achieves a significantly reduced negative flip rate (NFR) with the other. **Right:** Scatter plot of a ResNet-50's ER *vs* its NFR *w.r.t* a ResNet-18. The more left and lower, the better. ELODI improves both ER and NFR than baseline methods. Particularly, ELODI is close to the ensemble paragon, without the prohibitive computation cost of ensembles.

Key ideas and intuition. To address the *first* key question above, we propose *a generalized distillation objective,* termed *Logit Difference Inhibition (LDI),* which relaxes the requirement of exact logit matching to only limiting significant displacement in certain logit elements, thereby giving the model more flexibility to improve accuracy. To address the *second* question, we consider ensembles having the same architecture and trained on the same dataset. Instead of applying LDI as model distillation, we enforce LDI between a single model and the ensemble. The result is a single model that achieves similar NFR reduction as the ensemble without sacrificing accuracy.

These choices are motivated by the observation that negative flips, while not necessarily close to the decision boundary, frequently exhibit large variation in their relative distance across the ensemble. Thus, we use such deviations, which are reduced by ensembling, as a proxy of the likelihood of negative flip. However, we note that there are still negative flips that exhibit small deviations in their logits, so there is no shortage of exploration left to do. In practice, we train a single model by penalizing the difference of sample logits from the mean of a deep ensemble. While in theory the elements of the ensemble can be arbitrary, we independently train replicas of a single model with different random seeds. 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.[1]

These improvements are made possible by two main contributions: (1) a novel distillation loss, named *logit difference inhibition*, that reduces NFR without decreasing accuracy when the reference model has smaller capacity than the student (new), as customary in model updates; (2) ELODI, that integrates model ensemble and LDI by first training deep networks using LDI loss *w.r.t* 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.

2 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. (2019) empirically study prediction flip on training samples between epochs, termed "forgetting events", while Yan et al. (2021) address perceived regression using held-out sets between different models. Both are particular instance of cross-model compatibility (Shen et al., 2020; Bansal et al., 2019; Srivastava et al., 2020). Focal Distillation (Yan et al., 2021) minimizes the distance between the old and new predictions, with increased weights on samples correctly classified by the old model. Träuble et al. (2021) 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.

[1]Note that ELODI is able to deal with existing models trained without treatment, as shown in Appendix B.2.

Ensemble learning methods such as bagging (Breiman, 1996), boosting (Freund & Schapire, 1997; Chen & Guestrin, 2016; Hastie et al., 2009), and random forests (Breiman, 2001; Ke et al., 2017) are widely adopted in machine learning. The understanding for these methods is sometimes explained as enlarging the margins (Bartlett et al., 1998). Recently, the "multi-view" hypothesis (Allen-Zhu & Li, 2020) 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 apply ensemble learning as a teacher's model to guide the student model in reducing the 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 eventually replaced by its mean at inference time. The literature on variance reduction is too vast to survey here, but relevant references include (Hoeting et al., 1999; Fragoso et al., 2018).

Some other methods are summarized as follows: Deep ensemble (Lakshminarayanan et al., 2017) improves accuracy and allow estimating sample uncertainty; Snapshot Ensemble (Huang et al., 2017a) and Fast Geometric Ensemble (Garipov et al., 2018) train component models simultaneously, and Yan et al. (2021) show that ensembles help reduce regression. Ensembles are impractical in most real applications due to the multiplier they impose on inference cost. This has prompted research on "implicit ensembles" such as Dropout (Srivastava et al., 2014; Gal & Ghahramani, 2016), DropPath (Larsson et al., 2017) and Stochastic Depth (Huang et al., 2016). Wen et al. (2020) propose BatchEnsemble to generate ensemble weights while Havasi et al. (2021) use a MIMO design to train multiple sub-networks concurrently. These methods focus on accuracy instead of NFR.

Knowledge distillation (KD) (Hinton et al., 2015) 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 (Zhang et al., 2019), teacher and student are the same. Focal Distillation (Yan et al., 2021) 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* (Yuan et al., 2020), where it is used as regularization. Ensemble distillation uses multiple teachers to improve accuracy in vision and other applications (Reich et al., 2020; Fukuda et al., 2017; Asif et al., 2020; Malinin et al., 2020; Lin et al., 2020). 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) members of an ensemble in our methods have the same architecture and are trained on the same dataset with different initialization seeds, opposite to the diverse ensembles used in traditional ensemble distillation.

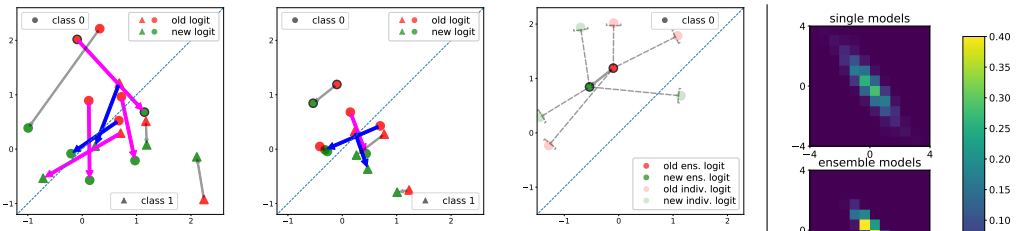
3 REPRESENTATION LANDSCAPE OF PC-TRAINING

To answer the two key questions in Section 1, we explore (1) how negative flips occur, and (2) why ensembles yield fewer negative flips (Yan et al., 2021). To do so, we analyze the so-called *logit space*, where the representations computed by a deep neural network before the SoftMax operation live. For a certain data sample, the logits produced by different models, trained on the same dataset, live in the same vector space which is defined by the class set of the training samples. This gives us insights on mechanisms underlying negative flips, which we first illustrate with a simple example.

3.1 A TWO-DIMENSIONAL EXAMPLE

To illustrate the behavior of models in logit space, we create a toy example by selecting *two* classes[2] from ImageNet (Deng et al., 2009) and training several ResNet-18 models for *binary* classification. The models differ by their initialization, determined by distinct random seeds; we then collect output logits for each test datum and model in the ensemble. In Figure 2a, we plot the two-dimensional logit vectors of multiple data points when updating from an 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; (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. Figure 2b

[2] "Labrador retriever" (n02099712) and "French bulldog" (n02108915).



(a) Two single models: Test samples flip even if not close to the boundary (long arrows).

(b) Two ensembles of 3 models each (3×): Fewer samples that are far from the boundary flip (shorter arrows).

(c) Logits of ensembles and their members: Individual models' logits (lighter circles) center around the mode.

(d) PMF of logit displacement.

Figure 2: **Visualization of a 2-class example**. **(a-c)**: Two-class logits of two single models and/or ensembles. ▲ and • refer to the ground-truth classes, while red and green data points refer to old and new model's logits. Magenta arrow, blue arrow, and gray arrow link negative flip, positive flip, and consistent (either both correct or both wrong) prediction pairs. All dots with black borders are depicting the same image. **(d)**: Estimated probability mass function (PMF) of logit displacement between two single models or ensembles. The x, y-axes denote the two classes' logit displacement. The heatmap value denotes the estimated probability density. The ensemble's co-variance is significantly smaller than the single model. The figure is best viewed in color.

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 Figure 2a, 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 Figure 2d, in updates between ensembles, the logit vectors are less likely to exhibit significant displacement[3]. This suggests that the ensemble may be reducing the negative flip rate through the removal of significant displacement of the logit vectors.

3.2 Ensembles in PC-Training

Logit spaces with more than 2 classes are not easily visualized. Instead, we formally analyze the representation landscape in higher dimensions as follows: Given an input image x, we have n models with their output logit vectors $\{\boldsymbol{\phi}^{(i)}(x)\}_{i=1}^{n}$. For any model pair $\boldsymbol{\phi}^{(i)}$ and $\boldsymbol{\phi}^{(j)}$, we define the *logit displacement* due to training randomness to be the embeddings' difference, *i.e* $\boldsymbol{\phi}^{(i)}(x) - \boldsymbol{\phi}^{(j)}(x)$. Once the difference reaches a threshold, the order of the top prediction changes, and a flip occurs.

Ensembles reduce negative flips by reducing the magnitude of the logit displacement. Without loss of generality, we assume $\{\boldsymbol{\phi}^{(1)}(x), \cdots, \boldsymbol{\phi}^{(n)}(x)\}$ to be n *i.i.d* random variables drawn from a distribution approximated to second-order by an expectation $\boldsymbol{\mu}$ and a co-variance matrix $\boldsymbol{\Sigma}$. Now we construct two *homogeneous* ensembles, each sampling m single models without overlapping and averaging the logit vectors of these models. Let $\mathbb{M}_1$ and $\mathbb{M}_2$ denote the set of models of each ensemble, and we have $\|\mathbb{M}_1\| = \|\mathbb{M}_2\| = m$ and $\mathbb{M}_1 \cap \mathbb{M}_2 = \varnothing$. For each ensemble we get the ensemble logit vector $\boldsymbol{\phi}^{(\text{ens},1)}(x)$ and $\boldsymbol{\phi}^{(\text{ens},2)}(x)$. The multi-dimensional central limit theorem (CLT) (Rvačeva, 1962; Ferguson, 2017) states that this average converges in distribution to a multivariate normal distribution with the increase of m, *i.e*

$$\phi^{(\text{ens})}(x) = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{\phi}^{(i)}(x) \overset{D}{\sim} \mathcal{N}(\boldsymbol{\mu}, \frac{1}{m}\boldsymbol{\Sigma}). \tag{1}$$

Therefore, the logit displacement between two ensembles with m non-overlapping members is

$$\boldsymbol{\phi}^{(\text{ens},1)}(x) - \boldsymbol{\phi}^{(\text{ens},2)}(x) = \frac{1}{m}\sum_{i\in\mathbb{M}_1} \boldsymbol{\phi}^{(i)}(x) - \frac{1}{m}\sum_{j\in\mathbb{M}_2} \boldsymbol{\phi}^{(j)}(x) \overset{D}{\sim} \mathcal{N}(\mathbf{0}, \frac{2}{m}\boldsymbol{\Sigma}), \tag{2}$$

This implies that the logit displacement between two ensembles reduces with more members. This can also be observed in the 2D example from Figure 2d.

[3]Data sample is randomly selected. We present the same visualization on more data points in Appendix D.

Logit displacement between ensembles with different functions. When two ensembles have different architectures or are trained on different datasets, the function that transforms the input to the logit vector is more prone to change. We refer to this case as updates between two *dissimilar* ensembles. Note that the members in each ensemble need to be the same type such that Equation 1 still holds. This results in a different distribution from which the logit vectors are sampled. In this case, it is likely that $\boldsymbol{\mu}^{(\text{ens},1)} \neq \boldsymbol{\mu}^{(\text{ens},2)}$. Then the logit displacement would follow a normal distribution with a *nonzero* mean:

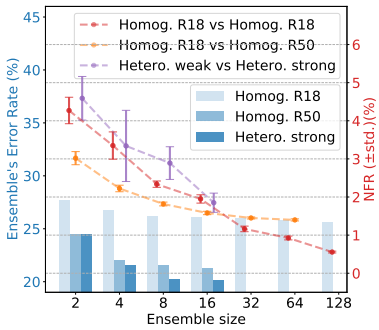


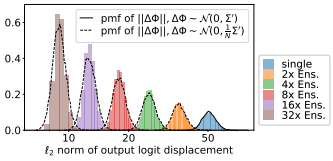
Figure 3: **ER/NFR *w.r.t* ensemble size**

$$\boldsymbol{\phi}^{(\text{ens},1)}(x) - \boldsymbol{\phi}^{(\text{ens},2)}(x) \overset{D}{\sim} \mathcal{N}\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2, \frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{m}\right). \tag{3}$$

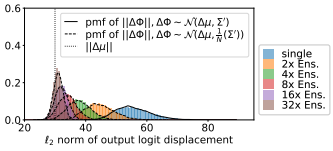
This results in an *extrinsic* logit displacement which does not go away when the ensembles get larger. This explains the observations in Figure 3: (1) Two very large ensembles with the same architecture (red curve) can have almost no flips; (2) NFR between two homogeneous ensembles with different architectures (orange curve) stagnates at a nonzero value; (3) NFR between two heterogeneous ensembles (purple curve) has significantly higher variance than the former two cases.

3.3 Validation in High Dimensions

We verify our hypothesis on the representation landscape in general cases as follows. We train a *standard* ResNet-18 on *full* ImageNet dataset with 256 random seeds. To visualize the logit displacement between two homogeneous ensembles with m models, we split the models into two halves. Then for a fixed input image, we randomly draw m models without replacement and compute the averaged logits of this drawn ensemble. The process is repeated on the same image with $m = 1, 2, \cdots, 32$. We present in Figure 4a the ℓ_2 norm histogram of logit displacement between two random ensembles with different m. As the ensemble size grows, the distribution of logit displacement condenses to lower values. We examine our hypothesis in Section 3.2 through probability mass function (PMF) of logit difference. First we use all available single models to estimate mean and co-variance $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for the logit vectors' distribution $\boldsymbol{\Phi}$. If we assume $\boldsymbol{\Phi}$ follows a Normal distribution, *i.e* $\boldsymbol{\Phi} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, so will the logit displacement between any two models, *i.e* $\Delta\boldsymbol{\Phi} \sim \mathcal{N}(\mathbf{0}, 2\boldsymbol{\Sigma})$. Then the norm of logit displacement will follow a generalized (central) χ^2 distribution (Mathai & Provost, 1992; Das & Geisler, 2021). Since the probability density function of a generalized χ^2 variable does not have a simple closed-form expression, we estimate it by Kernel Density Estimation (KDE) method (Parzen, 1962). From Figure 4a, 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 conduct the same experiments above on more images in Appendix D and the conclusion holds well, suggesting this property is not incidental. If we move to ensembles of m models each, the logit displacement follows another normal distribution whose co-variance matrix is scaled by m, *i.e* $\Delta\boldsymbol{\Phi}^{(\text{ens})} \sim \mathcal{N}\left(\mathbf{0}, \frac{2}{m}\boldsymbol{\Sigma}\right)$. We show that the other histograms are consistent with the estimated PMF of $\|\Delta\boldsymbol{\Phi}^{(\text{ens})}\|^2$ (dashed lines in Figure 4a), corroborating our hypothesis in Section 3.2.



(a) Homog. R-18 Ensemble *vs* Homog. R-18 Ensemble. $\Delta\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma}' = 2\boldsymbol{\Sigma}_{1,2}$.



(b) Homog. R-18 Ensemble *vs* Homog. R-50 Ensemble. $\Delta\boldsymbol{\mu} \neq \mathbf{0}, \boldsymbol{\Sigma}' = \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2$.

Figure 4: ℓ_2 **norm histogram of logit displacement between two random ensembles.** The bin size is 0.5. We also plot the simulated (PMF): the solid line for ℓ_2 norm of a simulated normal distribution $\mathcal{N}(\Delta\boldsymbol{\mu}, (\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2))$ whose parameters are estimated from all available single models; the dashed lines for extrapolated distribution $\mathcal{N}\left(\Delta\boldsymbol{\mu}, \frac{1}{m}\boldsymbol{\Sigma}'\right)$. Consistency between the histograms and PMFs supports our hypotheses in Section 3.2.

Finally, we examine the case of two dissimilar homogeneous ensembles by forming the other ensembles with ResNet-50 models and comparing them to the ResNet-18 ensembles obtained before. In this case we have $\boldsymbol{\Phi}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ for the ResNet-18 ensembles and $\boldsymbol{\Phi}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ for the ResNet-50 ensembles. Therefore the logit displacement follows $\Delta\boldsymbol{\Phi} \sim \mathcal{N}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)$ and its norm should follow a generalized *non-central* χ^2 distribution. From Figure 4b we can see that

the estimated distribution of the logit displacement norm fits the empirical distribution well. It still condenses as the enembles get large. But the mean converges to $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|$ (the dotted vertical line in Figure 4b) instead of decreasing to 0 as between the ResNet-18 ensembles in Figure 4a.

4 LOGIT DIFFERENCE INHIBITION

Based on the previous analysis, we derive a method to reduce NFR in classification model updates. To prevent a single model's output logits from having high variance from a reference model, we propose *logit difference inhibition* (LDI), rather than hard logit matching, which causes undesired impact to model accuracy. Next we extend LDI to ELODI to utilize ensemble's reduction of logit displacement and alleviate the need of precedent models in training.

4.1 LOGIT DIFFERENCE INHIBITION FOR SINGLE MODELS

Let $\boldsymbol{\phi}^{(\text{old})}(x)$ and $\boldsymbol{\phi}^{(\text{new})}(x)$ denote the old and new model's logit outputs before SoftMax for an input image x with label $\ell \in \{1, \cdots, C\}$. Each logit vector has C dimensions, *i.e* $\boldsymbol{\phi}^{(\cdot)}(x) \in \mathbb{R}^C$, and $\boldsymbol{\phi}_k^{(\cdot)}(x)$ denotes its k-th element. A straightforward way to reduce NFR is to apply a knowledge distillation loss that forces $\boldsymbol{\phi}^{(\text{new})}(x)$ to match $\boldsymbol{\phi}^{(\text{old})}(x)$ (Yan et al., 2021). However, as shown in Section 3.2, when $\boldsymbol{\phi}^{(\text{old})}(x)$ has a large displacement from its mode, forcing exact match can be detrimental to accuracy (Hinton et al., 2015; Yan et al., 2021). Therefore we only *inhibit* significant difference between old and new logit vectors' elements as follows:

$$\mathcal{L}_{\text{LDI}}(x) = \sum_{k \in \mathbb{K}} \max\left(0, \|\phi_k^{(\text{new})}(x) - \phi_k^{(\text{old})}(x)\| - \xi\right)^p, \tag{4}$$

where ξ is truncating threshold such that difference below ξ is tolerated. $\mathbb{K}$ is a data-dependent subset of classes where the difference will be inhibited. $\mathbb{K}$ can include either all classes ($\{1, \cdots, C\}$) or the classes with the top-K highest logit elements. Introducing a subset is motivated by our observation that flips are more likely to happen in high logit classes. As shown in experiments, the top-K form of LDI leads to no loss in NFR reduction and compared to the full form. It could instead help in reducing computation cost when number of classes are extremely large (An et al., 2020). p is set to be 2 in our experiments. When $\xi = 0$ and $\mathbb{K} = \{1, \cdots, C\}$, LDI degrades to the logit matching distance function that is used in Focal Distillation (Yan et al., 2021).

4.2 ELODI: ENSEMBLE LOGIT DIFFERENCE INHIBITION

Section 3.2 implies that model ensemble gives a good estimate of the logit for one architecture trained on the same data, *i.e* closer to its mode. Therefore the ensemble of many models appears to be a better reference than a single old model. For a model ensemble composed of m single models $\boldsymbol{\phi}^{(\text{ens},i)}, i \in \{1, \cdots, m\}$, the ensemble logit $\boldsymbol{\phi}_k^{(\text{ens})}(x)$ for sample x at class k is $\boldsymbol{\phi}_k^{(\text{ens})}(x) = \frac{1}{m}\sum_i \boldsymbol{\phi}_k^{(\text{ens},i)}(x)$. Replacing the old model's logits in Equation 4 by the ensemble's, we obtain the loss function for ensemble logit difference inhibition (ELODI) as

$$\mathcal{L}_{\text{ELODI}}(x) = \sum_{k \in \mathbb{K}} \max\left(0, \|\phi_k^{(\text{new})}(x) - \phi_k^{(\text{ens})}(x)\| - \xi\right)^p, \tag{5}$$

where $\boldsymbol{\phi}^{(\text{ens})}$ is obtained by averaging the logits of m models independently trained on the same dataset and having the same architecture as $\boldsymbol{\phi}^{(\text{new})}$. In Section 5 we show that this approach reduces NFR more and eases system updates.

The overall objective for training is a weighted sum of standard Cross Entropy loss and the LDI (or ELODI) loss, *i.e* $\mathcal{L} = (1-\alpha)\mathcal{L}_{\text{CE}} + \alpha\mathcal{L}_{\text{ELODI}}$, where the loss weight α is set such that the magnitude of $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{ELODI}}$ is similar. In single-model LDI, α is set to 0.5. From Figure 4a, we see that the expectation of logit difference norm decreases *w.r.t* the ensemble size. So in ELODI, we set α to 0.8.

The inference stage of ELODI does not involve the guiding ensembles. Only the single model trained with ELODI is deployed to replaced the old model, which is also trained with ELODI. This avoids the prohibitive inference cost of the ensembles while achieving similar level of NFR reduction to the ensembles (Yan et al., 2021) without sacrificing accuracy, as shown in Section 5.1. When the

Table 1: **The performance of different distillation methods on ImageNet. (a)**: Comparison between loss functions in PC training without ensembles. LDI is capable of reducing NFR without increasing ER much. **(b)**: Comparison between loss functions in PC training with ensembles. ELODI outperforms existing methods, *e.g* KD and FD. The ELODI-Top$K$ variant achieves similar or even slightly better performance.

(a) ResNet-18→ResNet-50 using LDI. Here ResNet-18 is *arbitrary* (without any positive-congruent treatment). The NFR reduction is largely insensitive to the choice of ξ, while a small non-zero ξ, *e.g* 0.5 or 1.0, can lead to slightly lower ER.

	$\text{ER}_\downarrow(\%)$	$\text{NFR}_\downarrow(\%)$
old model (R18)	30.24	N/A
No treatment	24.66	4.30
BCT (Shen et al., 2020)	25.00	4.34
KD (Hinton et al., 2015)	28.38	3.20
FD-KL (Yan et al., 2021)	26.32	2.90
FD-LM (Yan et al., 2021)	26.53	2.92
BU-CR (Träuble et al., 2021)	26.51	4.56
LDI ($\xi = 0.0$)	25.81	**2.86**
LDI ($\xi = 0.2$)	25.63	2.94
LDI ($\xi = 0.5$)	24.91	2.91
LDI ($\xi = 1.0$)	**24.64**	3.18

(b) ResNet-18$^\diamondsuit$→ResNet-50$^\diamondsuit$using ELODI. Both ResNet-18 and ResNet-50 are trained from an $8\times$ ensemble ($^\diamondsuit$denotes after distill). The baseline and ensemble paragon (in shaded gray) are included for comparison in the top rows.

Method	$\text{ER}_\downarrow(\%)$ R18$^\diamondsuit$	$\text{ER}_\downarrow(\%)$ R50$^\diamondsuit$	$\text{NFR}_\downarrow(\%)$
No treatment (single)	30.24	24.66	4.30
Ensemble ($8\times$)	26.34	22.44	1.95
Ensemble w/. $\text{KD}_{\tau=100}$	32.09	23.67	2.23
Ensemble w/. $\text{FD}_{\tau=100}$	32.19	23.97	2.16
Ensemble w/. $\text{FD}_{\tau=1}$	31.62	24.06	2.43
ELODI ($\xi = 0.0$)	31.34	23.15	2.18
ELODI (Top$K = 2$)	31.17	23.12	2.27
ELODI (Top$K = 5$)	31.35	23.21	2.21
ELODI (Top$K = 10$)	30.95	**23.10**	**2.11**

number of updates increase, this forms a chain of models with low NFR between them[4]. The fact that ELODI does not need the old models as references, contrary to previous methods (Yan et al., 2021; Bansal et al., 2019), would significantly simplify the training and model update process.

5 EXPERIMENTS

We validate the proposed approaches on two standard image classification datasets: ImageNet (Deng et al., 2009) and iNaturalist (Van Horn et al., 2018). For a new model in an update experiment, we measure top-1 error rate (**ER**) and its negative flip rate to the old model (**NFR**) (Yan et al., 2021), computed 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, ℓ_i is the label, and $\hat{y}_i^{(\text{new})}$ ($\hat{y}_i^{(\text{old})}$) is the new (old) model's prediction. The training recipe is described in Appendix A. Unless otherwise specified, ELODI experiments are done with ensemble size $m = 8$.

5.1 MAIN RESULTS OF ELODI

The effect of Logit Difference Inhibition. In Table 1a, we summarize the results of Logit Difference Inhibition for model update of ResNet-18→ResNet-50. The initial ResNet-18 model is trained with standard cross entropy loss (no treatment). We observe that LDI reduces NFR under all choices of ξ, the truncating threshold. With a relative large ξ, we can reduce NFR without harming the new model's accuracy, which was not possible in the baseline methods. However, we observe a trade-off between the new model's ER and NFR when varying ξ. Also, NFR stagnates at around 2.9%, indicating that it is hard to reduce NFR *w.r.t* an old model without treatment.

Next we switch to ELODI, where ResNet-18 and -50 are trained by ensemble with $m = 8$. From Table 1b, we first find that update with ELODI loss outperforms FD (Yan et al., 2021) (Ensemble w/. FD) or KD (Hinton et al., 2015) (Ensemble w/. KD) in both ER and NFR. Second, we observe that a smaller ξ achieves better final metrics compared to LDI. This supports our hypothesis that ensemble logits serve as a more reliable estimate than a single model. Third, we find that using the top-K highest-logit class subset in ELODI with $K \in \{2, 5, 10\}$ does not deteriorate the performance ($\pm 0.1\%$). From an ER-NFR scatter plot in Figure 1*(Right)*, ELODI achieves a similar level of ER-NFR results as the ensemble paragon (Yan et al., 2021) at the inference cost of a single model.

More update settings. Model updates may also come with the growth of training data, *e.g* (1) increasing number of classes, and (2) increasing number of per-class samples. We follow the same data/class split in (Yan et al., 2021), which uses 50% classes/samples for old model and full data for new. From Table 2, we find that conclusions from the full-data setting also hold for these settings.

[4]Results are presented in Appendix B.1 due to space limit.

Table 2: **ELODI in data-growth settings on ImageNet.** $^{\blacklozenge}$($^{\bigcirc}$) means that model is trained on half (full) data.

Method	Increasing #classes			Increasing #samples/class		
	Error Rate$_{\downarrow}$(%)		NFR$_{\downarrow}$	Error Rate$_{\downarrow}$(%)		NFR$_{\downarrow}$
	R18$^{\blacklozenge}$	R50$^{\bigcirc}$		R18$^{\blacklozenge}$	R50$^{\bigcirc}$	
None	22.02	24.66	14.07	34.26	24.66	3.52
FD (Yan et al., 2021)	-	39.96	5.45	-	33.06	2.65
LDI	-	26.67	4.82	-	26.17	2.60
Ens. (8×)	18.70	22.44	4.12	29.16	22.44	2.11
ELODI	21.80	23.15	4.19	34.08	23.15	2.25

Table 3: **ELODI on iNaturalist.** ELODI is effective when fine-tuning on iNaturalist under multiple settings.

Method	Increasing #classes			Increasing #samples/class			Full data		
	ER$_{\downarrow}$(%)		NFR$_{\downarrow}$	ER$_{\downarrow}$(%)		NFR$_{\downarrow}$	ER$_{\downarrow}$(%)		NFR$_{\downarrow}$
	R18$^{\blacklozenge}$	R50$^{\bigcirc}$		R18$^{\blacklozenge}$	R50$^{\bigcirc}$		R18$^{\bigcirc}$	R50$^{\bigcirc}$	
None	47.58	35.95	5.38	78.88	35.95	3.82	40.69	35.95	4.76
FD (Yan et al., 2021)	-	45.87	3.44	-	66.91	2.00	-	40.03	3.95
LDI	-	35.82	2.04	-	45.31	1.64	-	34.43	3.50
Ens. (8×)	36.03	29.47	1.68	47.04	29.47	1.23	36.25	29.47	2.10
ELODI	43.56	34.29	1.91	52.96	34.29	1.47	40.37	34.29	2.46

Table 4: **ELODI with different guiding ensembles.** We consider ResNet-18 →ResNet-50 via ELODI with an 8×-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 Appendix C. **Mixed-weak** (**-strong**) ensemble is a mixture of 4× ResNet-18 and 4× VGG-13 (4× ResNet-50 and 4× DenseNet-121).

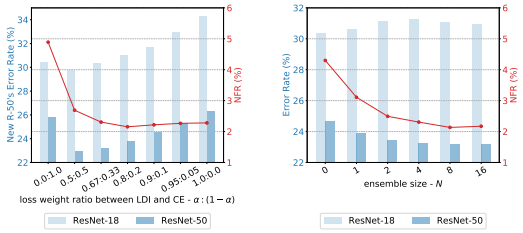
Old Reference	New Reference	Error Rate$_{\downarrow}$ (%)		NFR$_{\downarrow}$ (%)
		R18$^{\diamond}$	R50$^{\diamond}$	R18$^{\diamond}$→R50$^{\diamond}$
N/A	N/A	30.24	24.66	4.30
All-diff-weak	All-diff-weak	32.38	26.11	1.94
Mixed-weak	Mixed-weak	32.75	26.88	1.99
R-18 (×8)	R-18 (×8)	31.32	26.82	2.06
All-diff-weak	All-diff-strong	32.73	23.14	2.25
All-diff-weak (calibrated)	All-diff-strong (calibrated)	33.38	76.67	2.35
Mixed-weak	Mixed-strong	32.75	23.68	2.24
R-18 (×8)	R-50 (×8)	31.32	23.15	2.18

Fine-tuning on other datasets. We validate the effectiveness of ELODI when transferring to iNaturalist (Van Horn et al., 2018) following the protocol in (Yan et al., 2021). Results of both full-data and data-growth setting are summarized in Table 3, where ELODI consistently outperforms FD.

5.2 CHOICE OF THE GUIDING ENSEMBLE

Homogeneous *vs* heterogeneous ensembles. Analysis in Section 3.2 suggests that in ELODI we can use ensembles with all members having the same architectures, *i.e homogeneous ensembles*. However, in ensemble learning, members with strong diversity such as model architectures are usually favored for better generalization (Kuncheva & Whitaker, 2003). In Table 4, we observe that using homogeneous ensemble for guidance achieves comparable results in both NFR and ER than heterogeneous ("all-different") ensembles whether or not calibration (Guo et al., 2017) is considered in "all-different". This suggests that strong diversity in a guiding ensemble may not lead to better NFR reduction. Also, ELODI with homogeneous ensembles is easier to implement and extend.

Change of architecture for the guiding ensemble. In Table 4, we find that training a new model guided by an ensemble with the old model's architecture has to trade ER for reduction of NFR, which is not desired. This corroborates with the hypothesis in Section 3.2 that models with different architectures has different representation landscape and thus it is better to use ensemble with the



(a) Varying loss weights. (b) Ensemble size.

Figure 6: **Ablating loss weights and ensemble sizes for ELODI on ImageNet.** ER of ResNet-18 (-50) is shown in the light (dark) blue bar plot while NFR in the red curve.

Method	Error Rate$_\downarrow$(%)		NFR$_\downarrow$(%)
	R18$^\diamondsuit$	R50$^\diamondsuit$	
Offline	32.49	24.26	2.38
Online	30.97	23.81	2.15

Table 5: **Comparison between offline and online distill on ImageNet.** Inferring teacher logits during training (online) achieves both lower ER and NFR compared to pre-extracting it (offline).

Table 6: **ELODI with different architectures on ImageNet**. ELODI effectively reduces NFR on a wide range of architectures. † is obtained by our reproduction with different augmentation and training schedule. Note that all new models' NFR is measured *w.r.t* ResNet-18 listed in the leftmost column.

	ER$_\downarrow$ (%) R-18 (old)	ER$_\downarrow$ (%) R-101	NFR$_\downarrow$ (%) →R-101	ER$_\downarrow$ (%) D-161	NFR$_\downarrow$ (%) →D-161	ER$_\downarrow$ (%) SwinT	NFR$_\downarrow$ (%) →SwinT
None (single)	30.24	24.66	3.64	21.82	3.73	20.40†	3.77
LDI	-	23.71	2.57	22.16	2.85	19.74	2.56
Ensemble (8×)	26.34	20.05	1.72	18.90	2.06	18.37	2.60
ELODI ($N = 8$)	31.34	21.09	2.19	21.74	2.57	19.84	2.95

same architecture for guiding ELODI. When a system has gone through multiple updates, always guiding ELODI with the new model's architecture also provides a clear guideline for practice.

5.3 ABLATION STUDIES

The effect of loss weight. We experiment with different loss weight α and summarize the results in Figure 5a. $\alpha_{\text{ELODI}} = 0$ is equivalent to the no-treatment baseline. When α_{LDI} 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 $\sim$2.2%. We find $\alpha_{\text{ELODI}} = 0.8$ 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 efficiency for reducing NFR by varying the ensemble size m in Figure 5b. The case of $m = 1$ can be viewed as self distillation (Zhang et al., 2019) except that the new model's weight is re-initialized with a different random seed. NFR decreases from 4.30% to 2.15% by when the ensemble size increases from 1 to 8.

Online *vs* offline distillation. In ELODI, the ensemble's logits can be either inferred during training (online) or pre-extracted before training (offline). In Table 5, we find that offline distillation is less effective in reducing NFR and ER. Similar observation is also reported in (Beyer et al., 2022). Therefore we use the online approach in all experiments.

Updates to dissimilar architectures. In Table 6, we study if ELODI is applicable to updates across dissimilar architectures (ResNet-18→DenseNet-161/Tiny-Swin) besides similar ones (ResNet-101). We see that ELODI effectively reduces NFR in all cases, with retained or sometimes decreased ER.

6 DISCUSSION

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. But there could still be negative flip samples with small logit displacement. As discussed in Section 3.2 and observed in experiments, both ELODI and the ensemble paragon are not able to address the negative flips caused by the difference of representation landscape caused by architectural change. Mitigating this would require further analysis of the influence of neural network architecture design in PC training. 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

Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

Xiang An, Xuhan Zhu, Yang Xiao, Lan Wu, Ming Zhang, Yuan Gao, Bin Qin, Debing Zhang, and Fu Ying. Partial fc: Training 10 million identities on a single machine. In *Arxiv 2010.05222*, 2020.

Umar Asif, Jianbin Tang, and Stefan Harrer. Ensemble knowledge distillation for learning improved and efficient networks. In *ECAI*, 2020.

Gagan Bansal, Besmira Nushi, Ece Kamar, Daniel S Weld, Walter S Lasecki, and Eric Horvitz. Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff. In *AAAI*, 2019.

Peter Bartlett, Yoav Freund, Wee Sun Lee, and Robert E Schapire. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998.

Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *CVPR*, 2022.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *KDD*, 2016.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. In *NeurIPS*, 2017.

Abhranil Das and Wilson S Geisler. A method to integrate and classify normal distributions. *Journal of Vision*, 21(10):1–1, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. RepVGG: Making vgg-style convnets great again. In *CVPR*, 2021.

Thomas S Ferguson. *A course in large sample theory*. Routledge, 2017.

Tiago M Fragoso, Wesley Bertoli, and Francisco Louzada. Bayesian model averaging: A systematic review and conceptual classification. *International Statistical Review*, 86(1):1–28, 2018.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Takashi Fukuda, Masayuki Suzuki, Gakuto Kurata, Samuel Thomas, Jia Cui, and Bhuvana Ramabhadran. Efficient knowledge distillation from an ensemble of teachers. In *Interspeech*, 2017.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *NeurIPS*, 2018.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360, 2009.

Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew M Dai, and Dustin Tran. Training independent subnetworks for robust prediction. In *ICLR*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jennifer A. Hoeting, David Madigan, Adrian E. Raftery, and Chris T. Volinsky. Bayesian model averaging: a tutorial (with comments by M. Clyde, David Draper and E. I. George, and a rejoinder by the authors. *Statistical Science*, 14(4):382 – 417, 1999. doi: 10.1214/ss/1009212519. URL `https://doi.org/10.1214/ss/1009212519`.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. In *ICLR*, 2017a.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017b.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *NeurIPS*, 2017.

Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207, 2003.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. In *ICLR*, 2017.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *NeurIPS*, 2020.

Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation. In *ICLR*, 2020.

Arakaparampil M Mathai and Serge B Provost. *Quadratic forms in random variables: theory and applications*. Dekker, 1992.

Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *CVPR*, 2020.

Steven Reich, David Mueller, and Nicholas Andrews. Ensemble distillation for structured prediction: Calibrated, accurate, fast-choose three. In *EMNLP*, 2020.

EL Rvačeva. On domains of attraction of multi-dimensional distributions. *Selected Translations in Mathematical Statistics and Probability*, 2:183–205, 1962.

Yantao Shen, Yuanjun Xiong, Wei Xia, and Stefano Soatto. Towards backward-compatible representation learning. In *CVPR*, 2020.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Megha Srivastava, Besmira Nushi, Ece Kamar, Shital Shah, and Eric Horvitz. An empirical analysis of backward compatibility in machine learning systems. In *KDD*, 2020.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *ICLR*, 2019.

Frederik Träuble, Julius von Kügelgen, Matthäus Kleindessner, Francesco Locatello, Bernhard Schölkopf, and Peter Gehler. Backward-compatible prediction updates: A probabilistic approach. In *NeurIPS*, 2021.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist species classification and detection dataset. In *CVPR*, 2018.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *TPAMI*, 43(10):3349–3364, 2020.

Yeming Wen, Dustin Tran, and Jimmy Ba. BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning. In *ICLR*, 2020.

Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Yuqing Xie, Yi-an Lai, Yuanjun Xiong, Yi Zhang, and Stefano Soatto. Regression bugs are in your model! measuring, reducing and analyzing regressions in nlp model updates. In *ACL*, 2021.

Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi Deng, Meng Wang, Wei Xia, and Stefano Soatto. Positive-congruent training: Towards regression-free model updates. In *CVPR*, 2021.

Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, 2018.

Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *CVPR*, 2020.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *ICCV*, 2019.

A TRAINING RECIPE

We follow standard training recipe of ImageNet[5]. Specifically, all classification models are trained with SGD with momentum of 0.9 and base learning rate of 0.1, which is reduced by $\frac{1}{10}$ every 30 epochs until 90 epochs. The batchsize is 256 with 8 GPUs.

For ELODI with larger models and ensemble size (*e.g* $8\times$ DenseNet-161), GPU memory becomes a bottleneck. To handle the memory issue, we use gradient checkpointing (Chen et al., 2016) and reduce batch size while linearly scaling base learning rate (Goyal et al., 2017).

B MORE EXPERIMENTS ON ELODI

B.1 ELODI ON A CHAIN OF MODEL UPDATES

As discussed in Section 4.2, ELODI does not involve the guiding ensembles at the inference stage. Only the single model trained with ELODI is deployed to replaced the old model, which is also trained with ELODI. When the number of updates increase, this naturally forms a chain of models having low NFR between them.

We illustrate this *transitivity* of NFR reduction induced by ELODI in chain updates of three models, *i.e* ResNet-18→ResNet-50→ResNet-101. As shown in Table 7a, with ELODI, the NFR between the three models tested reduced to $2.04\% \sim 2.25\%$ from $3.92\% \sim 4.41\%$ (a relative reduction of $44.1\% \sim 52.3\%$), outperforming all previous methods, including variants of FD (Yan et al., 2021) and LDI. Note this is achieved without crafting the complex reference schemes which are necessary for baselines since they require old models' references.

B.2 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*

$$\mathcal{L}_{\text{total}} = \lambda \mathcal{L}_{\text{ELODI}}(\mathcal{M}^{\diamondsuit}_{\text{new}}; \mathcal{M}^{(\text{ens})}_{\text{new}}) + (1-\lambda)\mathcal{L}_{\text{LDI}}(\mathcal{M}^{\diamondsuit}_{\text{new}}; \mathcal{M}_{\text{old}}), \tag{6}$$

where $\mathcal{M}^{\diamondsuit}$ denote the model to be guided from the ensemble.

To show some empirical results, here we consider three models, ResNet-18→ResNet-50→ResNet-101, where ResNet-18 model is trained without ELODI. Therefore both ResNet-50$^{\diamondsuit}$ and -101$^{\diamondsuit}$ will have a higher NFR compared with ResNet-18. To handle this, we introduce an additional LDI loss targeted at ResNet-18 when training ResNet-50 and (or) ResNet-101 using ELODI. We can see that ELODI w/. LDI outperforms ELODI w/o. LDI on all pairwise NFRs, indicating that augmenting ELODI with LDI loss towards the existing model is effective in dealing with this legacy case.

C DETAILS OF HETEROGENEOUS ENSEMBLES IN FIGURE 3 AND TABLE 4

Model list of heterogeneous ensembles. The **all-diff-weak** ensemble is composed of 8 different weak models with Top-1 Accuracy $= 69 \sim 70\%$ on ImageNet, including: ResNet-18 (He et al., 2016), GoogleNet (Szegedy et al., 2015), VGG-11, VGG-13, VGG-11-BN, VGG-16 (Simonyan & Zisserman, 2014; Ioffe & Szegedy, 2015), HRNet-W18 (Wang et al., 2020), DLA-34 (Yu et al., 2018). The **all-diff-strong** ensemble is composed of 8 different strong models with Top-1 Accuracy $= 75 \sim 76\%$ on ImageNet, including ResNet-50 (He et al., 2016), DenseNet-121 (Huang et al., 2017b), Inception-V3 (Szegedy et al., 2016), VGG-19-BN (Simonyan & Zisserman, 2014), RegNetY (Radosavovic et al., 2020), RepVGG-A2 (Ding et al., 2021), DPN-68 (Chen et al., 2017), DLA-X-60-C (Yu et al., 2018). All of the model weights are adopted from PyTorch Image Models (TIMM) (Wightman, 2019).

Calibration of heterogeneous ensembles. We use temperature scaling Guo et al. (2017) for each individual model in the ensemble set to make the confidence score match the true correctness likelihood on validation set. The temperature for scaling is optimized based on the loss on validation

[5] https://github.com/pytorch/examples/tree/main/imagenet

Table 7: **Pairwise NFR on multiple models.** $\mathcal{M}_1 \rightarrow \mathcal{M}_2$ means that we measure $\mathcal{M}_2$'s NFR *w.r.t* $\mathcal{M}_1$. $\mathcal{M}_1 \Rightarrow \mathcal{M}_2$ means that $\mathcal{M}_2$ is trained with $\mathcal{M}_1$ being teacher using distillation loss, *e.g* FD or LDI. $\mathcal{M}^\diamond$ means that $\mathcal{M}$ is trained from ELODI.

Method	Pairwise NFR	Method	Pairwise NFR
None	R18 $\xrightarrow{4.28\%}$ R50 $\xrightarrow{4.41\%}$ R101, R18 $\xrightarrow{3.92\%}$ R101	ELODI	R18$^\diamond$ $\xrightarrow{\mathbf{2.04\%}}$ R50$^\diamond$ $\xrightarrow{2.25\%}$ R101$^\diamond$, R18$^\diamond$ $\xrightarrow{\mathbf{2.19\%}}$ R101$^\diamond$
FD (chain)	R18 $\xRightarrow{2.90\%}$ R50 $\xRightarrow{2.13\%}$ R101, R18 $\xrightarrow{3.46\%}$ R101	LDI (chain)	R18 $\xRightarrow{2.86\%}$ R50 $\xRightarrow{2.35\%}$ R101, R18 $\xrightarrow{2.69\%}$ R101
FD (radial)	R18 $\xRightarrow{2.90\%}$ R50 $\xrightarrow{2.33\%}$ R101, R18 $\xRightarrow{2.63\%}$ R101	LDI (radial)	R18 $\xRightarrow{2.86\%}$ R50 $\xrightarrow{3.09\%}$ R101, R18 $\xRightarrow{2.57\%}$ R101
FD (fc)	R18 $\xRightarrow{2.90\%}$ R50 $\xRightarrow{\mathbf{1.97\%}}$ R101, R18 $\xRightarrow{2.96\%}$ R101	LDI (fc)	R18 $\xRightarrow{2.86\%}$ R50 $\xRightarrow{2.96\%}$ R101, R18 $\xRightarrow{2.68\%}$ R101

(a) Sequential update. (1) **chain**: each model targets at its closest predecessor; (2) **radial**: each model targets at its farthest ancestor; (3) fully-connected (**fc**): each model targets at all its ancestors.

LDI usage	Pairwise NFR
None	R-18 $\xrightarrow{2.98\%}$ R-50$^\diamond$ $\xrightarrow{2.25\%}$ R-101$^\diamond$, R-18 $\xrightarrow{2.56\%}$ R-101$^\diamond$
Once	R-18 $\xRightarrow{\mathbf{2.85\%}}$ R-50$^\diamond$ $\xrightarrow{2.14\%}$ R-101$^\diamond$, R-18 $\xRightarrow{2.56\%}$ R-101$^\diamond$
Both	R-18 $\xRightarrow{\mathbf{2.85\%}}$ R-50$^\diamond$ $\xrightarrow{\mathbf{2.12\%}}$ R-101$^\diamond$, R-18 $\xRightarrow{\mathbf{2.48\%}}$ R-101$^\diamond$

(b) Integrating ELODI with existing models.

set with LBFGS. Since each model in the ensemble set has similar accuracy, after such calibration, their confidence scores are matched to the same level.

D VISUALIZATION ON MORE DATA POINTS

As mentioned in Section 3.1 and 3.3, we provide more examples to verify our hypothesis. We select four images of two classes from ImageNet (Deng et al., 2009), which are illustrated in Figure 7, as input data. With these input images, the estimated probability mass function (PMF) of logit displacement between two single model and two ensembles are shown in Figure 8. We can observe that the logit displacements are reduced with ensembles, which verifies our hypothesis of output logit vectors are actually independent and identically distributed (*i.i.d*) random variables and with multi-dimensional central limit theorem (CLT), their sum is a normal distribution (Equation 1).

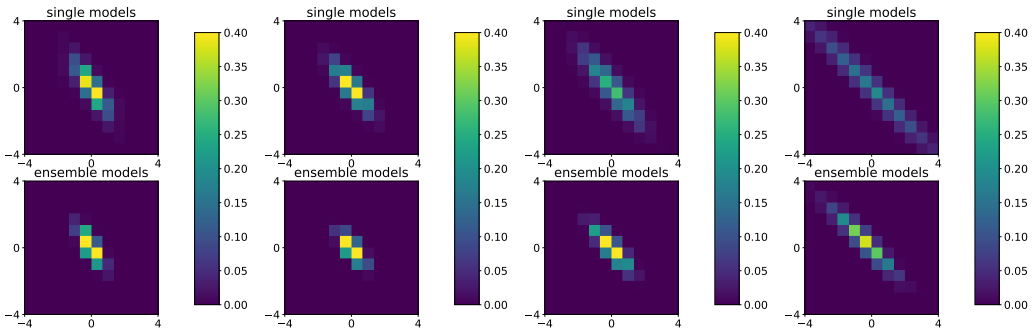
To verify our hypothesis in higher dimension space, we train a standard ResNet-18 on full ImageNet dataset with 256 random seeds. We take the images in Figure 7 as inputs and illustrate the ℓ_2 norm histogram of logit displacement between two random ensembles with different ensemble sizes in Figure 9. For heterogeneous case, we train a standard ResNet-50 on full ImageNet dataset and observe the ℓ_2 norm histogram of logit displacement between a random ResNet-18 ensembles and a random ResNet-50 ensembles with different ensemble sizes. The results are shown in Figure 10.

E REPRESENTATION LANDSCAPE OF PC-TRAINING AT THE FEATURE LEVEL

We have discussed the representation landscape of PC-Training in Section 3.2 at the *logit* space and provide some more data points above. The analysis can be done in the *feature* space as well. The main challenge is that features from two arbitrary models are not directly comparable and we address this *feature interoperability* issue by Backward-Compatible Training (Shen et al., 2020). We first introduce the BCT method and then derive that formulation to attain the feature at the *penultimate*

(a) val_00009585. (b) val_00015098. (c) val_00034619. (d) val_00014560.

Figure 7: **Four example images for visualization**. The two classes we select here are "French bulldog" (n02108915) and "Welsh Corgi" (n02113023).



(a) PMF of logit displacement (val_00009585). (b) PMF of logit displacement (val_00015098). (c) PMF of logit displacement (val_00034619). (d) PMF of logit displacement (val_00014560).

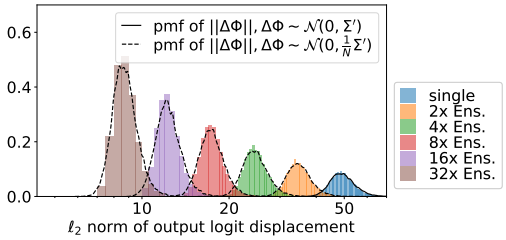
Figure 8: Estimated probability mass function (PMF) of logit displacement between two single models or ensembles. The x, y-axes denote the two classes' logit displacement. The heatmap value denotes the estimated probability density. The ensemble's co-variance is significantly smaller than the single model. The figure is best viewed in color.

layer of an ensemble. Based on these we can analyze two-dimensional examples and the higher dimension validation experiments.

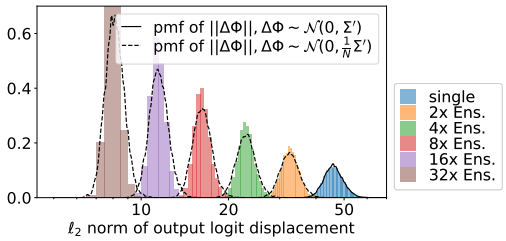
Preliminaries. Shen et al. (2020) propose an approach termed BCT to align two arbitrary deep models so that the embeddings are interoperable with each other. Formally speaking, a model $\mathcal{M}$ includes an embedding module ($\boldsymbol{z} = \mathcal{F}(x)$, *a.k.a backbone*) and a classification layer ($\boldsymbol{s} = \mathcal{H}(\boldsymbol{z})$, *a.k.a head*) on top, *i.e* $\mathcal{M}(x) = \boldsymbol{\phi}(x) = (\mathcal{H} \circ \mathcal{F})(x)$. Given a reference model $\mathcal{M}^{(\text{ref})}$, BCT imposes a loss term so that two model heads are close, *i.e* $\mathcal{H}_i^{(\text{bct})} \sim \mathcal{H}^{(\text{ref})}$[6]. As a result, $\mathcal{F}^{(\text{bct})}(x)$ and $\mathcal{F}^{(\text{ref})}(x)$ lie in a same vector space and are thus comparable, *i.e* $\mathcal{F}^{(\text{bct})}(x) \sim \mathcal{F}^{(\text{ref})}(x)$, regardless of the underlying architecture.

Ensemble of many feature-interoperable models. It is noteworthy that feature interoperability does **not** affect NFR as reported by Yan et al. (2021). We also re-validate that two models, $\mathcal{F}_1^{(\text{bct})}(x)$ and $\mathcal{F}_2^{(\text{bct})}(x)$, trained using BCT *w.r.t* $\mathcal{M}^{(\text{ref})}$ have similar NFR compared to two without BCT. However, their features are comparable, *i.e* $\mathcal{F}_1^{(\text{bct})}(x) \sim \mathcal{F}_2^{(\text{bct})}(x) \sim \mathcal{F}^{(\text{ref})}(x)$. So is any linear combination in between.

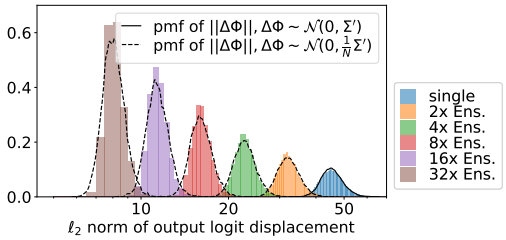
[6]In fact if we assume that $\mathcal{H}^{(\text{bct})}$ and $\mathcal{H}^{(\text{ref})}$ have the same shape, we can also do as follows: we train $\mathcal{M}^{(\text{ref})}$ and then $\mathcal{M}^{(\text{bct})}$ with parameters randomly initialized except the head copies weights from $\mathcal{H}^{(\text{ref})}$ and is *fixed*. Nevertheless, we follow BCT's formulation since it is more generic.



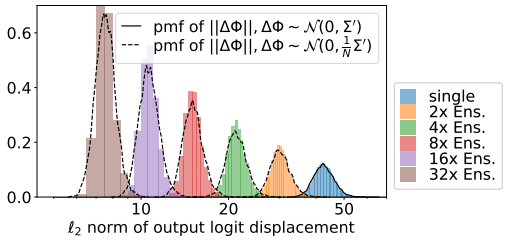
(a) Input image: `val_00009585`.



(b) Input image: `val_00015098`.

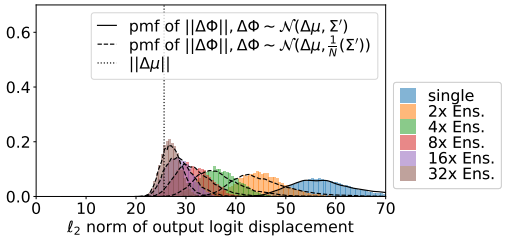


(c) Input image: `val_00034619`.

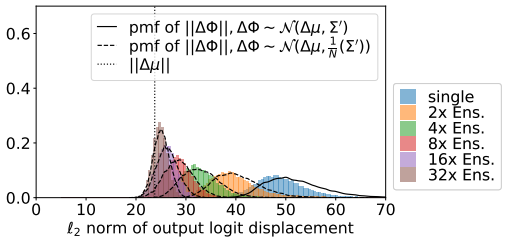


(d) Input image: `val_00014560`.

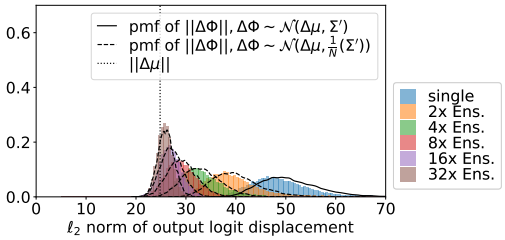
Figure 9: ℓ_2 **norm histogram of logit displacement between two random ensembles.** The bin size is 0.5. Two random ensembles are the same type (homogeneous ResNet-18 *vs* homogeneous ResNet-18). $\Delta\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 = \mathbf{0}, \boldsymbol{\Sigma}' = 2\boldsymbol{\Sigma}_1 = 2\boldsymbol{\Sigma}_2$.



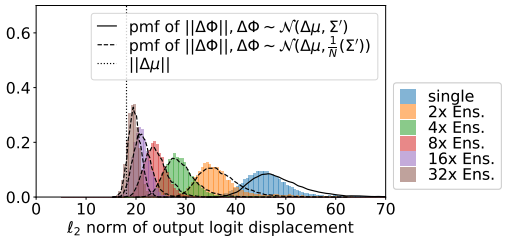
(a) Input image: `val_00009585`.



(b) Input image: `val_00015098`.



(c) Input image: `val_00034619`.



(d) Input image: `val_00014560`.

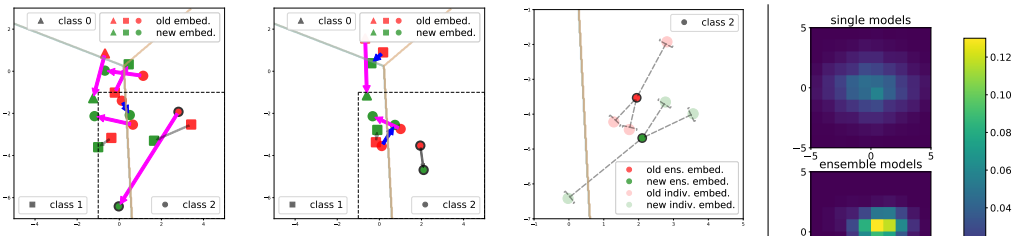
Figure 10: ℓ_2 **norm histogram of logit displacement between two random ensembles.** The bin size is 0.5. Two random ensembles are different types (homogeneous ResNet-18 vs homogeneous ResNet-50). $\Delta\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \neq \mathbf{0}, \boldsymbol{\Sigma}' = \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2$.

The arguments hold when the number of feature-interoperable models n increases. Therefore, if we write down their averaged logits, we can factor out the head, *i.e*

$$\phi^{(\text{bct,ens})}(x) = \frac{1}{N}\sum_n \phi_n^{(\text{bct})}(x) = \frac{1}{N}\sum_n \left(\mathcal{H}_n^{(\text{bct})} \circ \mathcal{F}_n^{(\text{bct})}\right)(x) \tag{7}$$

$$\approx \frac{1}{N}\sum_n \left(\mathcal{H}^{(\text{ref})} \circ \mathcal{F}_n^{(\text{bct})}\right)(x) = \mathcal{H}^{(\text{ref})} \circ \left(\frac{1}{N}\sum_n \mathcal{F}_n^{(\text{bct})}(x)\right). \tag{8}$$

It implies that the averaged feature can be viewed as this ensemble's feature, *i.e* $\mathcal{F}^{(\text{bct,ens})}(x) = \frac{1}{N}\sum \mathcal{F}_n^{(\text{bct})}(x)$.



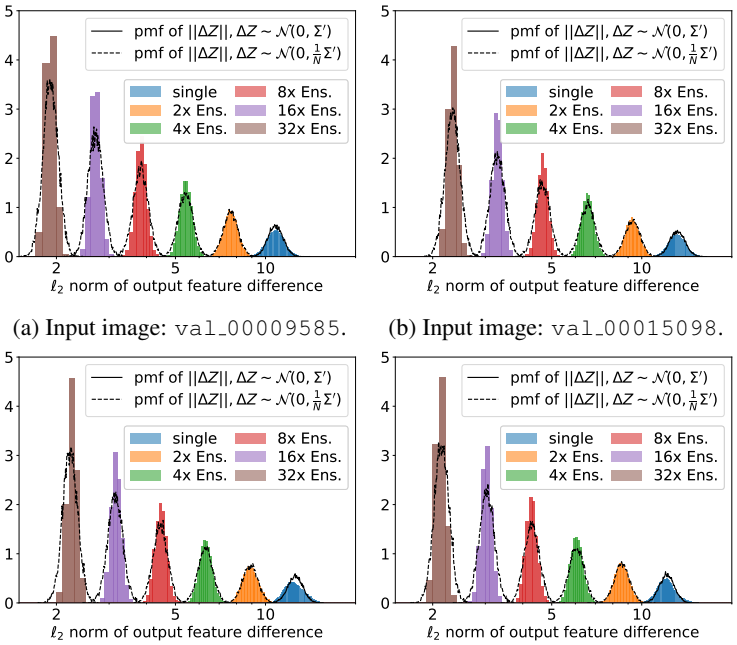
(a) Two single models: Test samples flip even if not close to the boundary (long arrows).

(b) Two ensembles of 3 models each (3×): Fewer samples that are far from the boundary flip (shorter arrows).

(c) Embeddings of ensembles and their members: Individual models' embeddings (lighter circles) center around the mode.

(d) PMF of feature difference $\Delta \boldsymbol{z}$.

Figure 11: **Visualization of a 3-class 2-dimensional example**. **(a-c)**: 2D feature embedding of two single models or ensembles. ▲, ■, and • refer to the ground-truth classes for each sample. Red and green data points refer to old and new model's embeddings. Magenta arrow, blue arrow, gray arrow link negative flip, positive flip, and consistent (either both correct or both wrong) prediction pairs. All dots with black borders are depicting the same image. **(d)**: Estimated probability mass function (PMF) of feature difference between two single models or ensembles. The x- and y-axes denote the 2D feature difference. The heatmap value denotes the estimated probability density. The ensemble's co-variance is significantly smaller than the single model. The figure is best viewed in color.



(a) Input image: `val_00009585`.

(b) Input image: `val_00015098`.

(c) Input image: `val_00034619`.

(d) Input image: `val_00014560`.

Figure 12: ℓ_2 **norm histogram of feature difference between two random ensembles.** Note that the bin size is 0.1. Two random ensembles are of the same type (ResNet-18 *vs* ResNet-18). $\Delta\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \neq \mathbf{0}, \boldsymbol{\Sigma}' = \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2$. We also plot the simulated probability mass function (PMF): the solid line for the norm of a simulated normal distribution $\mathcal{N}\left(\Delta\boldsymbol{\mu}, (\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)\right)$ whose parameters are estimated from all available single models; the dashed lines for extrapolated distribution $\mathcal{N}\left(\Delta\boldsymbol{\mu}, \frac{1}{m}(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)\right)$. Consistency between the ensembles' histograms and PMFs supports our hypotheses in Section E.

A two-dimensional example. To illustrate the behavior of models in feature space, we create a toy example by selecting three classes[7] from ImageNet (Deng et al., 2009) and training a ResNet-18-like

[7] "Labrador retriever" (n02099712), "Weimaraner" (n02092339), and "French bulldog" (n02108915).

models with a slight modification: the penultimate layer's dimension is changed to 2. The feature level visualization is presented in Figure 11a and 11d. We can observe the similar observations as in the logit space after the penultimate layer features are aligned with BCT (Shen et al., 2020).

Validations on higher dimensions. We repeat the high-dimensional validation in text on the penultimate layer features, the results are shown in Figure 12. We see that the PMF curve fits the histogram of single models well, implying that feature of these models could indeed follow a Normal distribution. We conduct the same experiments above on many more images and the conclusion holds well. If we move to ensembles of m models each, the feature difference follows another normal distribution whose co-variance matrix is scaled by a factor of m, *i.e* $\Delta \boldsymbol{z}^{(\mathrm{ens})} \sim \mathcal{N}\left(\mathbf{0}, \frac{2}{m}\boldsymbol{\Sigma}\right)$. We demonstrate that the rest of histograms are indeed consistent with the estimated PMF of $\|\Delta \boldsymbol{z}^{(\mathrm{ens})}\|^2$ (dashed lines in Figure 12).