Diversity Matters When Learning From Ensembles

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep ensembles excel in large-scale image classification tasks both in terms of prediction accuracy and calibration. Despite being simple to train, the computation and memory cost of deep ensembles limits their practicability. While some recent works propose to distill an ensemble model into a single model to reduce such costs, there is still a performance gap between the ensemble and distilled models. We propose a simple approach for reducing this gap, i.e., making the distilled performance close to the full ensemble. Our key assumption is that a distilled model should absorb as much function diversity inside the ensemble as possible. We first empirically show that the typical distillation procedure does not effectively transfer such diversity, especially for complex models that achieve near-zero training error. To fix this, we propose a perturbation strategy for distillation that reveals diversity by seeking inputs for which ensemble member outputs disagree. We empirically show that a model distilled with such perturbed samples indeed exhibits enhanced diversity, leading to improved performance.

1 Introduction

Deep Ensemble (DE) [Lakshminarayanan et al., 2017], a simple method to ensemble the same model trained multiple times with different random initializations, is considered to be a competitive method for various tasks involving deep neural networks both in terms of prediction accuracy and uncertainty estimation. Several works have tried to reveal the secret behind DE's effectiveness. As stated by Duvenaud et al. [2016] and Wilson and Izmailov [2020], DE can be considered as an approximate Bayesian Model Average (BMA) procedure. Fort et al. [2019] studied the loss landscape of DEs and showed that the effectiveness comes from the diverse modes reached by ensemble members, making it well suited for approximating BMA. It is quite frustrating that most sophisticated approximate Bayesian inference algorithms, especially the ones based on variational approximations, are not as effective as DEs in terms of exploring various modes in parameter spaces.

Despite being simple to train, DE incurs significant overhead in inference time and memory usage. It is therefore natural to develop a way to reduce such costs. An example of such a method is Knowledge Distillation (KD) [Hinton et al., 2015], which transfers knowledge from a large *teacher network* to a relatively smaller *student network*. The student network is trained with a loss that encourages it to copy the outputs of the teacher networks evaluated at the training data. With KD, there have been several works that learn a single student network by distilling from DE teacher networks. A naïve approach would be to directly distill ensembled outputs of DE teachers to the single student network. A better way proven to be more effective is to set up a student network having multiple subnetworks (multiple heads [Tran et al., 2020] or rank-one factors [Mariet et al., 2020]) and distill outputs of each ensemble member to each subnetwork in a one-to-one fashion. However, the empirical performance of such distilled networks is still far inferior compared to that of DE teachers.

There may be many reasons why the students are not doing as well as DE teachers. We argue that a critical limitation of current distillation schemes is that they do not effectively transfer the diversity

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

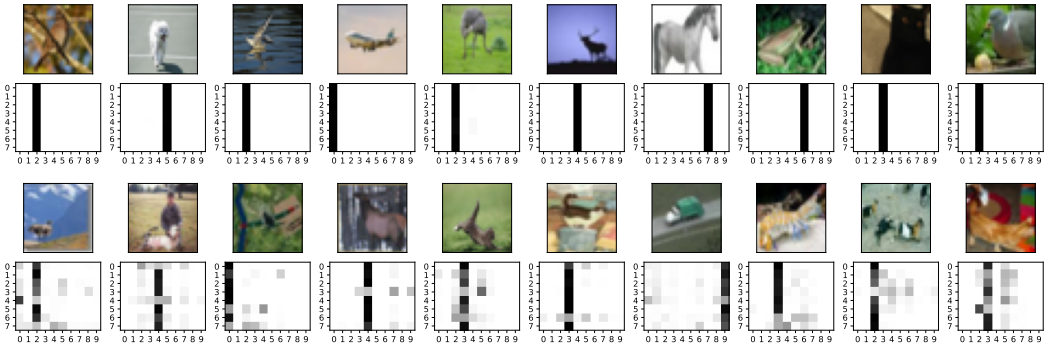


Figure 1: The top two rows show the randomly selected train examples in CIFAR-10 and corresponding predictions given by 8 ensembles of ResNet-32 models. In the second row, the vertical axis denotes the ensemble members, and the horizontal axis denotes the class index. The bottom two rows show the train examples having the lowest confidences. The ensemble predictions are still quite similar.

of DE teacher predictions to students. Consider an ensemble of deep neural networks trained for an image classification task. Given sufficient network capacity, each ensemble member is likely to achieve near-zero training error, meaning that the resulting outputs of ensemble members evaluated at the training set will be nearly identical (Fig. 1). In such a situation, reusing the training data during distillation will not encourage the student to produce diverse predictions. This is quite critical, considering various results establishing that the effectiveness of DE comes from averaging out its diverse predictions.

To this end, we propose a simple method that can amplify the diversities of students learning from DE teachers. Our idea is simple: instead of using the same training set, use the perturbed training set for which the predictions of ensemble members are likely to be diversified. To implement this idea, we employ Output Diversified Sampling (ODS) [Tashiro et al., 2020], a sampling scheme that finds small input perturbations that result in significant changes in outputs. Empirically, we confirm that the inputs perturbed with ODS result in large deviations in the DE teacher outputs. We further justify our method by analyzing the role of ODS perturbation. Specifically, we show that distilling with the training data perturbed with ODS can be interpreted as an approximate gradient matching procedure where ODS improves the sampling efficiency of the approximate gradient matching. Intuitively, by approximately matching gradients of teachers and students, we are transferring the outputs of teacher networks evaluated not only on the training data points but also on nearby points.

Using standard image classification benchmarks, we empirically validate that our distillation method promotes diversities in student network predictions, leading to improved performance, especially in terms of uncertainty estimation.

2 Backgrounds

2.1 Settings and notations

The focus of this paper is on the K-way classification problem taking D-dimensional inputs. We denote a student neural network as $\mathcal{S}(\boldsymbol{x}) : \mathbb{R}^D \to [0,1]^K$ and a teacher neural network as $\mathcal{T}(\boldsymbol{x}) : \mathbb{R}^D \to [0,1]^K$. $\mathcal{S}(\boldsymbol{x})$ and $\mathcal{T}(\boldsymbol{x})$ outputs class probabilities, and we denote the logits before softmax as $\hat{\mathcal{S}}(\boldsymbol{x})$ and $\hat{\mathcal{T}}(\boldsymbol{x})$. The kth element of an output is denoted as $\mathcal{S}^{(k)}(\boldsymbol{x})$. For DE teachers with $j = 1, \ldots, M$ members, the jth ensemble member is denoted as $\mathcal{T}_j(\boldsymbol{x})$. For a student network having M subnetworks, the jth subnetwork is denoted as $\mathcal{S}_j(\boldsymbol{x})$.

2.2 Knowledge distillation

KD [Hinton et al., 2015] aims to train a student network $\mathcal{S}(\boldsymbol{x})$ by matching its outputs to the outputs of a teacher network $\mathcal{T}(\boldsymbol{x})$. In addition to the standard cross-entropy loss between student outputs

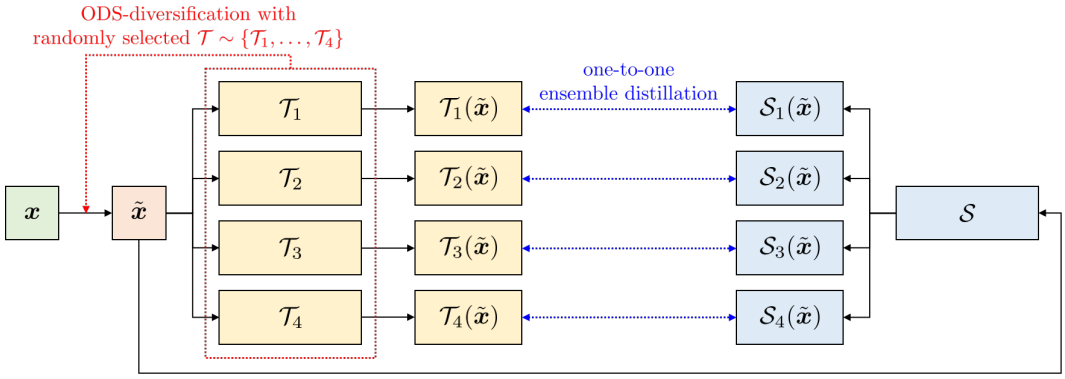


Figure 2: Overall structure of our method: (1) We first perturb train examples with respect to deep ensemble (DE) teachers, *i.e.*, $\boldsymbol{x} \to \tilde{\boldsymbol{x}}$. (2) DE teachers produce *diverse* predictions given perturbed train examples. (3) Since the diversity of teachers has been revealed, a student can effectively learn its diversity in the knowledge distillation framework.

and the ground-truth labels, i.e.,

$$\mathcal{L}_{\text{CE}}(\mathcal{S}(\boldsymbol{x}), \boldsymbol{y}) = -\sum_{k=1}^{K} y^{(k)} \log \mathcal{S}^{(k)}(\boldsymbol{x}), \tag{1}$$

KD additionally encourages the student's output to be as close as possible to that of the teacher by also minimizing the KL-divergence between output class distributions of the student and the teacher.

$$\mathcal{L}_{\text{KD}}(\mathcal{S}(\boldsymbol{x}), \mathcal{T}(\boldsymbol{x}); \tau) = -\tau^2 \sum_{k=1}^{K} \text{softmax}^{(k)}\left(\frac{\hat{\mathcal{T}}(\boldsymbol{x})}{\tau}\right) \log \text{softmax}^{(k)}\left(\frac{\hat{\mathcal{S}}(\boldsymbol{x})}{\tau}\right). \tag{2}$$

Here, $\tau > 0$ is a *temperature* parameter which controls the smoothness of the distributions. We scale the KL divergence by τ^2 since the soft targets scale as $1/\tau^2$. As a result, in the KD framework, the student imitates the teacher's outputs for training examples by minimizing the following loss,

$$\mathcal{L} = (1-\alpha)\mathcal{L}_{\text{CE}} + \alpha \mathcal{L}_{\text{KD}}, \tag{3}$$

where $\alpha \in (0, 1)$ is a hyperparameter to be specified.

2.3 BatchEnsemble and one-to-one distillation

BatchEnsemble (BE) [Wen et al., 2020] is a lightweight ensemble method that reduces the number of parameters by weight sharing. Specifically, each layer of a BE consists of shared weights $\boldsymbol{W}$ and rank-one factors $\boldsymbol{r}_j \boldsymbol{s}_j^\top$ for $j = 1, \ldots, M$ with M being number of subnetworks. Unlike the full ensemble that constructs full set of M weights, BE shares $\boldsymbol{W}$ across all subnetworks while the rank-one factors are kept specific to each subnetwork. The weight matrix of the jth subnetwork is computed as $\boldsymbol{W} \circ \boldsymbol{r}_j \boldsymbol{s}_j^\top$ and consequently BE requires far fewer parameters than a full ensemble.

During training, BE simultaneously trains all subnetworks by feeding a duplicate of the current minibatch for each rank-one factor $\boldsymbol{r}_j \boldsymbol{s}_j^\top$ into the network. This procedure is easily vectorized because the computation graph for each subnetwork has the same structure, differing only in weight values. Mariet et al. [2020] proposed to further improve the performance of BE by distilling from fully trained DE teachers instead of training from scratch. During distillation, the parameters of jth subnetwork $\mathcal{S}_j(\boldsymbol{x})$ is trained by distilling from the jth DE teacher $\mathcal{T}_j(\boldsymbol{x})$. The corresponding objective function is then written as (c.f., Fig. 2 without perturbation)

$$\mathcal{L} = \sum_{j=1}^{M} \Big\{ (1-\alpha)\mathcal{L}_{\text{CE}}(\mathcal{S}_j(\boldsymbol{x}), \boldsymbol{y}) + \alpha \mathcal{L}_{\text{KD}}(\mathcal{S}_j(\boldsymbol{x}), \mathcal{T}_j(\boldsymbol{x}); \tau) \Big\}. \tag{4}$$

Throughout the rest of the paper, we will use the BE as student networks and follow this one-to-one distillation framework.

2.4 Output diversified sampling

ODS [Tashiro et al., 2020] is a sampling method that can maximize diversities in outputs of a given function. Let $\mathcal{F} : \mathbb{R}^D \to \mathbb{R}^K$ be a target function. ODS draws a perturbation vector guided from the *gradient* of the target function as follows.

$$\boldsymbol{w} \sim \mathrm{Unif}([-1, 1])^K, \quad \boldsymbol{v}_{\mathrm{ODS}}(\boldsymbol{x}, \mathcal{T}, \boldsymbol{w}) = \frac{\nabla_{\boldsymbol{x}}(\boldsymbol{w}^\top \mathcal{F}(\boldsymbol{x}))}{\|\nabla_{\boldsymbol{x}}(\boldsymbol{w}^\top \mathcal{F}(\boldsymbol{x}))\|_2} \in \mathbb{R}^D. \tag{5}$$

The intuition behind this sampling strategy is to seek the direction that maximizes the similarity between the function output and a randomly sampled vector $\boldsymbol{w}$. Following such a direction can increase the diversity in outputs due to the randomness in $\boldsymbol{w}$.

3 Learning from Ensembles with Output Diversification

3.1 One-to-one distillation with ODS

As discussed before, vanilla one-to-one distillation evaluates DE teachers on points where their outputs are nearly identical, so the students trained using them tend to exhibit low diversity in outputs. To avoid this undesirable behavior, we propose to instead train students on points perturbed with ODS. Let $(\boldsymbol{x}, \boldsymbol{y})$ be an training input pair. We first pick a *random teacher* $\mathcal{T}_r$ uniformly from $\{\mathcal{T}_1, \ldots, \mathcal{T}_M\}$ and perturb $\boldsymbol{x}$ as

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \eta \boldsymbol{v}_{\mathrm{ODS}}(\boldsymbol{x}, \mathrm{softmax}(\hat{\mathcal{T}}_r(\boldsymbol{x})/\tau), \boldsymbol{w}), \tag{6}$$

where $\eta > 0$ is a step-size. Ideally, we would like to evaluate ODS vectors for each teacher to build M perturbed versions of $\boldsymbol{x}$, but this would take too much time for gradient computation, especially when we have a large number of ensemble models. Instead, we just pick one of the teachers, use it to perturb the input, and share the perturbed input to train all students with the following loss function,

$$\mathcal{L} = \sum_{j=1}^{M} \Big\{ (1-\alpha)\mathcal{L}_{\mathrm{CE}}(\mathcal{S}_j(\boldsymbol{x}), \boldsymbol{y}) + \alpha \mathcal{L}_{\mathrm{KD}}(\mathcal{S}_j(\tilde{\boldsymbol{x}}), \mathcal{T}_j(\tilde{\boldsymbol{x}}); \tau) \Big\}, \tag{7}$$

where we are evaluating the KL-divergence between teacher outputs and student outputs on the *perturbed input*. An implicit assumption here is that *a diversifying direction computed from a specific teacher can be transferred to other teachers.* This assumption was empirically justified in Tashiro et al. [2020], where ODS was computed on a surrogate model when we do not have access to the true model. Our experiments in Section 5.2 re-confirm this assumption in the context of DE.

We empirically found that scaling the ODS updates according to confidence slightly improves the performance. That is, we perturb an input $\boldsymbol{x}$ as

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \eta C_{\mathrm{min}}(\boldsymbol{x}) \boldsymbol{v}_{\mathrm{ODS}}(\boldsymbol{x}, \mathrm{softmax}(\hat{\mathcal{T}}_r(\boldsymbol{x})/\tau), \boldsymbol{w}), \tag{8}$$

where $C_{\mathrm{min}}(\boldsymbol{x}) = \min_k \mathrm{softmax}^{(k)}(\hat{\mathcal{T}}_r(\boldsymbol{x})/\tau)$ denotes the minimum class probability. This scheme takes a larger step towards ODS directions when computed from confident examples. We call this modification ConfODS.

The training procedure with ODS perturbation is straightforward (Algorithm 1); we can just add the ODS computation procedure and replace the loss function from the original KD training procedure. Fig. 2 depicts the overall training procedure as a diagram.

3.2 Interpretation as approximate gradient matching

We can also interpret our distillation strategy as an approximate gradient matching procedure. Srinivas and Fleuret [2018] showed that a KD procedure on inputs perturbed by small noise implicitly encourages matching gradients (Jacobians to be more precise, because we are working with K-dimensional outputs) of a teacher and a student. More specifically, let $\mathcal{S}(\boldsymbol{x})$ and $\mathcal{T}(\boldsymbol{x})$ be student and teacher models. Using Taylor expansion, the expected KD loss on perturbed inputs can be written as[1]

$$\mathbb{E}_{\boldsymbol{\varepsilon}}[\mathcal{L}_{\mathrm{KD}}(\mathcal{S}(\boldsymbol{x}+\boldsymbol{\varepsilon}), \mathcal{T}(\boldsymbol{x}+\boldsymbol{\varepsilon})] = \mathcal{L}_{\mathrm{KD}}(\mathcal{S}(\boldsymbol{x}), \mathcal{T}(\boldsymbol{x})) + \mathcal{L}_{\mathrm{JM}}(\mathcal{S}(\boldsymbol{x}), \mathcal{T}(\boldsymbol{x})) + o(\sigma^2), \tag{9}$$

[1] We assume $\tau = 1$ for notational simplicity, but the argument applies equally well to any τ.

Algorithm 1 Knowledge distillation from deep ensembles with ODS perturbations

Require: Training data $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$
Require: Knowledge distillation weight α, temperature τ, learning rate β, ODS step-size η.
Require: Deep ensemble teacher networks $\{\mathcal{T}_j\}_{j=1}^M$ where M denotes the size of ensembles

Let $\mathcal{S}$ be a student network which consists of M subnetworks $\{\mathcal{S}_j\}_{j=1}^M$.
Initialize parameters $\boldsymbol{\theta}$ of the student network $\mathcal{S}$.
while not converged **do**
  Sample $(\boldsymbol{x}, \boldsymbol{y}) \sim \mathcal{D}$, $\boldsymbol{w} \sim \mathrm{Unif}([-1, 1])^K$, and a teacher index r uniformly from $\{1, \ldots, M\}$.
  Sample diversified inputs $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \eta \boldsymbol{v}_{\mathrm{ODS}}(\boldsymbol{x}, \mathrm{softmax}(\hat{\mathcal{T}}_r(\boldsymbol{x})/\tau), \boldsymbol{w})$
  $\mathcal{L}_{\mathrm{CE}} \leftarrow -\sum_{j=1}^M \sum_{k=1}^K y^{(k)} \log \mathcal{S}_j^{(k)}(\boldsymbol{x})$
  $\mathcal{L}_{\mathrm{KD}} \leftarrow -\sum_{j=1}^M \sum_{k=1}^K \mathcal{T}_j^{(k)}(\tilde{\boldsymbol{x}}) \log \mathcal{S}_j^{(k)}(\tilde{\boldsymbol{x}})$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}}((1-\alpha)\mathcal{L}_{\mathrm{CE}} + \alpha \mathcal{L}_{\mathrm{KD}})$
end while

with $\sigma^2 = \mathbb{E}[\boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon}]$ and

$$\mathcal{L}_{\mathrm{JM}}(\mathcal{S}(\boldsymbol{x}), \mathcal{T}(\boldsymbol{x})) = -\mathbb{E}_{\boldsymbol{\varepsilon}}\left[\boldsymbol{\varepsilon}^\top \mathcal{J}_{\mathcal{T}}(\boldsymbol{x})^\top \frac{\mathcal{J}_{\mathcal{S}}(\boldsymbol{x})}{\mathcal{S}(\boldsymbol{x})} \boldsymbol{\varepsilon}\right], \tag{10}$$

where $\mathcal{J}_{\mathcal{T}}(\boldsymbol{x})$ and $\mathcal{J}_{\mathcal{S}}(\boldsymbol{x})$ are the Jacobian matrices of $\mathcal{S}$ and $\mathcal{T}$ evaluated at $\boldsymbol{x}$, respectively. While Srinivas and Fleuret [2018] suggested using isotropic Gaussian noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_D)$, we argue that simple Gaussian noise is ineffective, especially for high-dimensional data that are likely to be embedded in low-dimensional manifolds. In such a case, most of the directions of perturbations are orthogonal to the data manifold, and thus Gaussian noise becomes uninformative. On the other hand, ODS directly uses the Jacobian of the teacher to construct the perturbation direction. We conjecture that due to this fact, KD with ODS perturbation is much more efficient approximate Jacobian matching compared to using Gaussian perturbations. We empirically validate this in Section 5.3.

The above analysis is for the usual KD framework with a single teacher and a single student. In our case, we are matching M teachers to M student subnetworks, but with ODS perturbations computed from a single teacher randomly sampled from M teachers. Again, we are operating under the assumption that the Jacobian computed from a specific teacher will transfer well to other teachers, so the approximate Jacobian matching using it is still more effective than the one with Gaussian perturbations.

4 Related Works

Knowledge distillation KD is a method for transferring the information inside a large model into a smaller one [Hinton et al., 2015] by minimizing the KL divergence between teacher and student networks. [Wang et al., 2020] finds that additionally augmenting the data brings additional benefits because such augmentations allow the KD loss to tap into teacher information outside of the training set. Furthermore, modifications to the standard KD objective allow one to distill the knowledge in an ensemble of teachers into one student network while preserving the benefits in uncertainty estimation of ensembles [Malinin et al., 2020, Tran et al., 2020]. KD can be enhanced via gradient matching, and several existing methods [Srinivas and Fleuret, 2018, Czarnecki et al., 2017] propose sampling methods for knowledge distillation up to higher-order gradients. However, such existing methods suffer from a high computational burden or inefficient random perturbation.

Ensembles Ensemble methods [Hansen and Salamon, 1990, Dietterich, 2000], which construct a set of learners and make predictions on new data points by a weighted average, have been studied extensively. Ensembles of models perform at least as well as each individual member [Krogh and Vedelsby, 1994], and achieve the best performance when each member makes errors independently. Aside from benefits in model accuracy, Deep Ensembles [Lakshminarayanan et al., 2017, Ovadia et al., 2019] have recently shown to be a simple and scalable alternative to Bayesian neural networks because of their superior uncertainty estimation performance. Wen et al. [2020] proposes a network architecture that uses a rank-one parameterization to train an ensemble with low computation and memory cost.

5 Experiments

In this section, we try to answer the following questions with empirical validation.

- Do images perturbed by ODS increase diversity in DE teacher predictions? - Section 5.2.
- Do ODS perturbations encourage gradient matching? - Section 5.3.
- Does enhanced diversity of students lead to improved performance in terms of prediction accuracy and uncertainty estimation? - Section 5.4.

5.1 Experimental Setup

Datasets and networks We compared our methods on CIFAR-10 and CIFAR-100. We used ResNet-32 [He et al., 2016] for CIFAR-10 and WideResNet-28x10 [Zagoruyko and Komodakis, 2016]. Please refer to Appendix A.1 for detailed training settings.

Hyperparameter settings The important hyperparameters for KD are the pair (α, τ); for CIFAR-10, after through hyperparameter search, we decided to stay consistent with the convention [Hinton et al., 2015, Cho and Hariharan, 2019, Wang et al., 2020] where $(\alpha, \tau) = (0.9, 4)$ for all methods. For CIFAR-100, we used the value $(\alpha, \tau) = (0.9, 1)$ for all methods because it uniformly worked the best. The step-size η for ODS is fixed to $1/255$. See Appendix A.1 for more details.

Uncertainty metrics We measure the uncertainty calibration performances with common metrics including Negative Log-Likelihood (NLL), Expected Calibration Error (ECE) and Brier Score (BS). See Appendix A.2 for the definition of the metrics. For all methods, the metrics are computed at optimal temperatures obtained from Temperature Scaling (TS) [Guo et al., 2017], as suggested by Hansen and Salamon [1990]. We also report the Deep Ensemble Equivalent (DEE) score [Ashukha et al., 2020], which measures the effective number of models by comparing it to a full DE. For a model $\mathcal{S}$, DEE score is defined as

$$\text{DEE}(\mathcal{S}) = \min\{\ell \geq 0 \,|\, \text{NLL}(\text{Ensemble of } \ell \text{ models}) \geq \text{NLL}(\mathcal{S})\}. \tag{11}$$

That is, DEE is computing the minimum number of independent ensemble members needed to achieve the same performance as a given model $\mathcal{S}$. The NLL values of ensembles with non-integer ℓ values are obtained by linear interpolation.

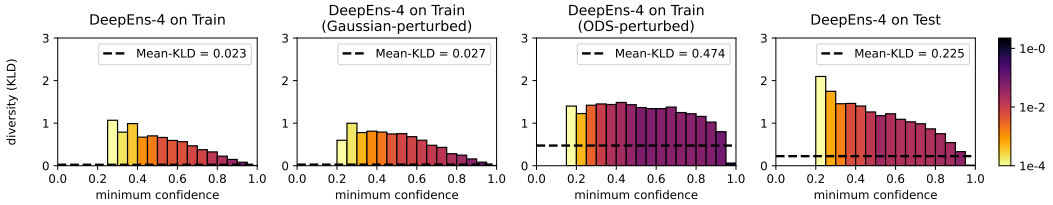
5.2 Impact of ODS on diversities

In order to assess the effect of ODS perturbation on the diversities of DE teachers or student sub-networks, we draw *diversity plots* in Fig. 3. To draw a diversity plot, we first collect the outputs of target models evaluated on a specific dataset and binned the samples w.r.t. their confidence values (minimum confidence among M models). For each confidence bin, we computed average pairwise KL-divergence between class output probabilities of ensemble members. We also report the representative value (mean-KLD), which is defined as average KL-divergence values weighted by bin counts.

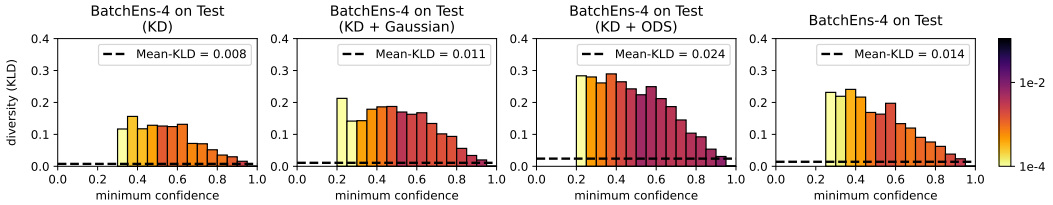
Fig. 3a shows the diversity plots of DE teachers on CIFAR-10. Intuitively, the samples predicted with high confidences usually come with low diversities, and this is indeed depicted in the plots. For the DE evaluated on the training set, the mean-KLD value is very small (the first box) compared to the one computed on the test set (the fourth box). While Gaussian perturbation does not affect the diversities (the second box), ODS dramatically improves the diversities (the third box), even outperforming the one evaluated at the test set.

Fig. 3b shows the diversity plots of the BE students trained on CIFAR-10. Although evaluated at the test set, the BE distilled without any perturbation (the first box), the BE distilled with Gaussian perturbations (the second box), and the BE trained from scratch (the fourth box) exhibit very low diversity values. On the other hand, the BE distilled with ODS perturbations show relatively high diversity value, and as we will show later in Section 5.4, this indeed leads to improved performances.

In Section 3, we assumed that an ODS computed from a specific teacher can be transferred to other teachers to diversify them as well. To see this, we draw the diversity plot using ODS computed from an external teacher network (a network trained in the same way as the teachers but not actually being



(a) Diversity plots of BE-4 teacher for ResNet-32 on train examples of CIFAR-10.



(b) Diversity plots of BE-4 students for ResNet-32 on test examples of CIFAR-10.

Figure 3: Colors of the bars denote the density of the bins, i.e., ratio of samples belonging to the bins.

used during distillation) in Fig. 6a. The mean-KLD values are indeed increased compared to the original DE or DE evaluated on examples perturbed with Gaussian noises, confirming the validity of our assumption.

5.3 ODS for gradient matching

Effectiveness We empirically validate our conjecture that ODS perturbations are more effective in gradient matching. To this end, we computed the cosine similarities between the gradients of ResNet-32 teacher and student networks trained with vanilla KD. We computed the same quantity using student networks distilled with Gaussian perturbations and ODS perturbations. We draw Receiver Operating Characteristic (ROC) curves to compare the similarities of distilled students and the baseline trained from scratch. Results in Fig. 5 clearly show that the gradients of the student distilled with ODS better match the gradients, while the student distilled with Gaussian perturbations did not significantly improve upon the student distilled with vanilla KD.

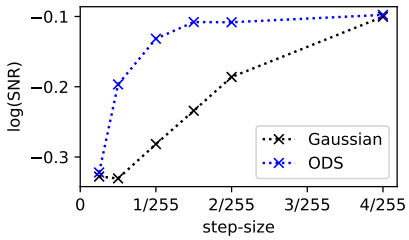


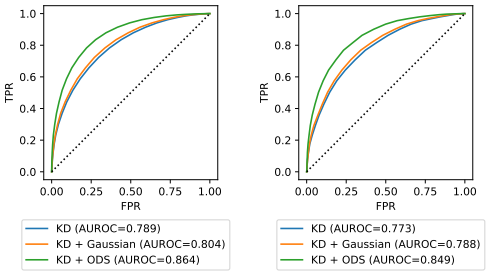
Figure 4: SNR values of the gradients computed with Gaussian perturbations and ODS perturbations with varying step-sizes.

Next, we compute the gradient of the term related to the gradient matching, i.e.,

$$\nabla_{\boldsymbol{\theta}}(\mathcal{L}_{\text{KD}}(\mathcal{S}(\boldsymbol{x}+\boldsymbol{\varepsilon}),\mathcal{T}(\boldsymbol{x}+\boldsymbol{\varepsilon}))-\mathcal{L}_{\text{KD}}(\mathcal{S}(\boldsymbol{x}),\mathcal{T}(\boldsymbol{x}))). \tag{12}$$

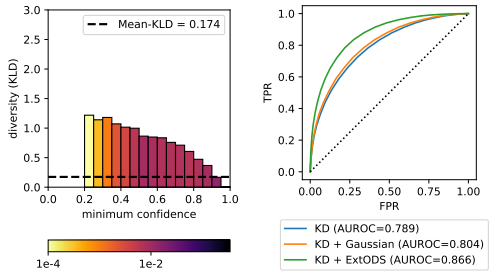
Using both Gaussian perturbations and ODS perturbations, we collected multiple samples of the gradients and measured the Signal-to-Noise Ratio (SNR) of the gradient norms. As shown in Fig. 4, the gradients computed with ODS perturbations exhibit higher SNR, indicating that learning with such gradients is more effective.

Transferability As in Section 5.2, we performed an experiment to verify our assumption that ODS perturbations for a specific model are transferable to other models, for the purpose approximate gradient matching. We trained a BE student with ODS perturbations computed from an external teacher and measured the cosine similarities between the student and teachers (*not* between the students and the external teacher). As one can see from Fig. 6b, the resulting student network copied the gradients of the teachers significantly better than the baselines.



(a) Training set **(b)** Validation set

Figure 5: ROC-curves with cosine similarities of gradients between a teacher and students.



(a) Diversity plot **(b)** ROC-curve

Figure 6: Transferability analysis using diversity plot and ROC-curve.

Table 1: Knowledge distillation from DE-M into BE-M where M denotes the size of ensembles: ACC (accuracy) and calibrated metrics including NLL, BS, ECE, and DEE. All values for ResNet-32 and WideResNet-28x10 are averaged over four and three experiments, respectively.

		BatchEns « DeepEns (ResNet-32 on CIFAR-10)				
Method	# Params	ACC	NLL	BS	ECE	DEE
$\mathcal{T}$: DeepEns-4	1.86 M	94.42	0.167	0.081	0.008	-
$\mathcal{S}$: BatchEns-4	0.47 M	93.37±0.11	0.204±0.002	0.107±0.001	0.008±0.001	1.419±0.075
+ KD		93.98±0.20	0.188±0.003	0.099±0.002	0.009±0.002	2.019±0.174
+ KD + Gaussian		93.93±0.12	0.187±0.001	0.099±0.002	0.009±0.002	2.042±0.089
+ KD + ODS		93.89±0.10	0.181±0.002	0.094±0.001	**0.006**±0.001	2.486±0.164
+ KD + ConfODS		**94.01**±0.19	**0.180**±0.001	**0.093**±0.001	0.007±0.001	**2.524**±0.080
$\mathcal{T}$: DeepEns-8	3.71 M	94.78	0.157	0.077	0.005	-
$\mathcal{S}$: BatchEns-8	0.48 M	93.47±0.14	0.202±0.005	0.104±0.003	**0.006**±0.002	1.494±0.156
+ KD		94.15±0.13	0.182±0.001	0.095±0.001	0.010±0.001	2.391±0.113
+ KD + Gaussian		94.09±0.08	0.184±0.002	0.097±0.001	0.010±0.002	2.206±0.175
+ KD + ODS		94.13±0.08	0.175±0.003	**0.090**±0.001	**0.006**±0.002	2.991±0.322
+ KD + ConfODS		**94.18**±0.12	**0.174**±0.002	0.091±0.001	0.007±0.001	**3.064**±0.246

		BatchEns « DeepEns (WideResNet-28x10 on CIFAR-100)				
Method	# Params	ACC	NLL	BS	ECE	DEE
$\mathcal{T}$: DeepEns-4	146.15 M	82.52	0.661	0.250	0.022	-
$\mathcal{S}$: BatchEns-4	36.62 M	80.34±0.08	0.755±0.007	0.298±0.002	0.027±0.001	1.449±0.085
+ KD		80.51±0.22	0.744±0.003	0.283±0.001	**0.021**±0.003	1.582±0.035
+ KD + Gaussian		80.39±0.12	0.761±0.006	0.286±0.000	0.022±0.003	1.379±0.072
+ KD + ODS		**81.88**±0.32	0.674±0.016	**0.257**±0.006	0.026±0.003	3.303±0.769
+ KD + ConfODS		81.85±0.32	**0.672**±0.010	**0.257**±0.003	0.024±0.000	**3.333**±0.491

5.4 Main results: image classification tasks

We validate whether the enhanced diversity of student networks is helpful for actual prediction. We compared our methods against three baselines: BE trained from scratch [Wen et al., 2020], BE distilled without perturbation [Mariet et al., 2020], and BE distilled with Gaussian noises. The results are presented in Table 1. For both datasets, BE distilled with ODS or ConfODS significantly outperformed baselines in terms of every metric we measured. The improvement is much more significant for the larger ensembles ($M = 8$ for CIFAR-10) and the larger dataset (CIFAR-100). We note that using our diversification strategy improves performance both in terms of raw accuracy and uncertainty metrics. An especially noteworthy result on CIFAR-100 is that BE distilled with ODS achieved a DEE score over three. This is quite remarkable considering the BE model has far fewer parameters than the full DE which, by definition, has an expected DEE of four.

One great benefit of the KD framework is that it can transfer knowledge to networks having different architectures, greatly reducing the number of parameters required to achieve a certain level of performance. Using the CIFAR-100 dataset, we compared our method to the baselines with student

Table 2: Cross-architecture knowledge distillation for a model compression: BE-4 students with smaller networks (*i.e.*, WideResNet-28x2 and WideResNet-28x5), learn from the DE-4 teacher which consists of larger networks (*i.e.*, WideResNet-28x10).

		BatchEns-4 (WRN-28x2) « DeepEns-4 (WRN-28x10)			
Method	# Params	ACC	NLL	BS	ECE
$\mathcal{T}$: DeepEns-4	146.15 M	82.52	0.661	0.250	0.022
$\mathcal{S}$: BatchEns-4	1.50 M	75.17$\pm$0.27	**0.896**$\pm$0.009	0.383$\pm$0.004	**0.013**$\pm$0.002
+ KD		75.19$\pm$0.36	0.904$\pm$0.007	0.377$\pm$0.007	0.016$\pm$0.002
+ KD + Gaussian		74.50$\pm$0.17	0.937$\pm$0.008	0.389$\pm$0.004	0.015$\pm$0.000
+ KD + ODS		**76.03**$\pm$0.22	0.898$\pm$0.014	**0.333**$\pm$0.003	0.021$\pm$0.003
+ KD + ConfODS		76.01$\pm$0.16	0.900$\pm$0.004	0.334$\pm$0.002	0.028$\pm$0.004
		BatchEns-4 (WRN-28x5) « DeepEns-4 (WRN-28x10)			
Method	# Params	ACC	NLL	BS	ECE
$\mathcal{T}$: DeepEns-4	146.15 M	82.52	0.661	0.250	0.022
$\mathcal{S}$: BatchEns-4	9.20 M	78.75$\pm$0.11	0.801$\pm$0.012	0.324$\pm$0.004	0.021$\pm$0.002
+ KD		78.89$\pm$0.10	0.804$\pm$0.012	0.314$\pm$0.004	0.022$\pm$0.001
+ KD + Gaussian		78.80$\pm$0.41	0.815$\pm$0.009	0.313$\pm$0.007	**0.020**$\pm$0.002
+ KD + ODS		80.24$\pm$0.05	0.742$\pm$0.008	0.279$\pm$0.002	0.028$\pm$0.004
+ KD + ConfODS		**80.62**$\pm$0.25	**0.733**$\pm$0.007	**0.275**$\pm$0.003	0.027$\pm$0.001

networks having different architectures. The results displayed in Table 2 show that KD with ODS or ConfODS outperforms the baselines by a wide margin.

6 Conclusion

In this paper, we proposed a simple yet effective training scheme for distilling knowledge from an ensemble of teachers. The key idea of our approach is to perturb training data in a proper way to expose student networks to the diverse outputs produced by ensemble teachers. We employed ODS to find such diversifying perturbation. Our distillation scheme with ODS perturbation can be interpreted as a sample-efficient approximate gradient matching procedure. We empirically verified that our method enhances the diversities of students, and that such enhanced diversities help to reduce the performance gap between student and teacher models in KD. An interesting future work would be to rigorously analyze the gradient matching aspect of our method and to provide theoretical justification and a better distillation method to harness it. Also, in this paper, we constrained ourselves to BE students, applying our framework to other student network architectures such as Bayesian neural networks would be interesting.

Limitations Although we have shown promising results, our method still has several limitations. Due to its nature as a KD framework, we still need to train M teachers has a considerable computational cost. Also, our method cannot improve the student beyond its capacity. For instance, in Table 2, WRN-28x2 distilled from WRN-28x10 does not greatly benefit from our method. Arguably the most critical limitation is that our method is a heuristic without any theoretical guarantee. In that sense, as mentioned above, rigorously analyzing the effect of ODS perturbation will be important future research.

References

A. Ashukha, A. Lyzhov, D. Molchanov, and D. P. Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International Conference on Learning Representations (ICLR)*, 2020.

J. H. Cho and B. Hariharan. On the efficacy of knowledge distillation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Swirszcz, and R. Pascanu. Sobolev training for neural networks. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

T. G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, 2000.

D. Duvenaud, D. Maclaurin, and R. P. Adams. Early stopping as nonparametric variational inference. In *Proceedings of The 19th International Conference on Artificial Intelligence and Statistics (AISTATS 2016)*, 2016.

S. Fort, H. Hu, and B. Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv:1912.02757*, 2019.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *Proceedings of The 34th International Conference on Machine Learning (ICML 2017)*, 2017.

L. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

G. E. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *Deep Learning and Representation Learning Workshop, NIPS 2014*, 2015.

A. Krogh and J. Vedelsby. Neural network ensembles, cross validation and active learning. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, 1994.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

A. Malinin, B. Mlodozeniec, and M. J. F. Gales. Ensemble distribution distillation. In *International Conference on Learning Representations (ICLR)*, 2020.

Z. Mariet, R. Jenatton, F. Wenzel, and D. Tran. Distilling ensembles improves uncertainty estimates. *3rd Symposium on Advances in Approximate Bayesian Inference*, 2020.

Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv:1906.02530*, 2019.

S. Srinivas and F. Fleuret. Knowledge transfer with jacobian matching. In *Proceedings of The 35th International Conference on Machine Learning (ICML 2018)*, 2018.

Y. Tashiro, Y. Song, and S. Ermon. Diversity can be transferred: Output diversification for white- and black-box attacks. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

L. Tran, B. S. Veeling, K. Roth, J. Swiatkowski, J. V. Dillon, J. Snoek, S. Mandt, T. Salimans, S. Nowozin, and R. Jenatton. Hydra: Preserving ensemble diversity for model distillation. *arXiv:2001.04694*, 2020.

H. Wang, S. Lohit, M. Jones, and Y. Fu. Knowledge distillation thrives on data augmentation. *arXiv:2012.02909*, 2020.

Y. Wen, D. Tran, and J. Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations (ICLR)*, 2020.

A. G. Wilson and P. Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

S. Zagoruyko and N. Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 1.
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] As our work deals with training a generic lightweight neural network, we do not find any negative impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix A
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix A.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 5.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Appendix A.
   (b) Did you mention the license of the assets? [Yes] See Appendix A.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See appendices.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We are using open-source dataset only.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Despite not discussed in the paper, our work do not contain any personally identifiable information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]