Noise as a Resource for Learning in Knowledge Distillation

Elahe Arani*, Fahad Sarfraz*, and Bahram Zonooz
Advanced Research Lab, NavInfo Europe, Eindhoven, The Netherlands
{elahe.arani, fahad.sarfraz}@navinfo.eu, bahram.zonooz@gmail.com

Abstract

While noise is commonly considered a nuisance in computing systems, a number of studies in neuroscience have shown several benefits of noise in the nervous system from enabling the brain to carry out computations such as probabilistic inference as well as carrying additional information about the stimuli. Similarly, noise has been shown to improve the performance of deep neural networks. In this study, we further investigate the effect of adding noise in the knowledge distillation framework because of its resemblance to collaborative subnetworks in the brain regions. We empirically show that injecting constructive noise at different levels in the collaborative learning framework enables us to train the model effectively and distill desirable characteristics in the student model. In doing so, we propose three different methods that target the common challenges in deep neural networks: minimizing the performance gap between a compact model and large model (Fickle Teacher), training high performance compact adversarially robust models (Soft Randomization), and training models efficiently under label noise (Messy Collaboration). Our findings motivate further study in the role of noise as a resource for learning in a collaborative learning framework.

1. Introduction

Noise permeates every level of the nervous system, from the perception of sensory signals to the generation of motor responses [13]. Despite its pervasiveness, noise has been predominantly viewed as a nuisance in computing systems [35]. Recently, though, there has been a shift in neuroscience and a number of studies argue for the beneficial role of noise [13, 35, 37]. Multiple noise sources contribute to trial-to-trial variability, variations in neural responses to the same stimuli, which can be considerably different across stimuli, suggesting that it could also provide an important contribution to the information conveyed by the neural responses about the stimuli [48]. Instead of a mere consequence of inherently stochastic processes on the molecular level, noise and trial-to-trial variability can be considered as salient components of the computational strategy of the brain [35]. The brain exploits noise to carry out computations, such as probabilistic inference through sampling [35]. Furthermore, neural networks that have been formed in the presence of noise will be more robust and explore more states, which will facilitate learning and adaptation to the changing demands of a dynamic environment [13].

Analogously, numerous empirical studies have shown that noise plays a crucial role in effective and efficient training of neural networks [62]. Noise has been used as a common regularization technique to improve the generalization performance of overparameterized deep neural networks (DNNs) by adding it to the input data, the weights, or the hidden units [1, 5, 17, 50, 51, 56]. Many noise techniques have been shown to improve generalization such as Dropout [50] and injection of noise to the gradients [6, 41]. Previous studies also showed that noise is crucial for non-convex optimization [26, 32, 58, 63]. Zhou *et al*. [62] showed that noise enables the gradient descent algorithm to efficiently escape from the spurious local optimum and converge to a global optimum.

All of these studies suggest that noise can indeed be a critical resource for learning. To further investigate the role of noise, we focus on the knowledge distillation framework because of its resemblance to the collaborative learning between different regions in the brain. It also enables training high-performance compact models for efficient real-world deployment on resource-constrained devices. Knowledge distillation involves training a smaller model (student) under the supervision of a larger pre-trained model (teacher) and consistently provides generalization gains. Despite the promising performance gain, there is still a significant generalization gap between the student and teacher. Consequently, an optimal method of capturing knowledge from the larger model and transferring it to a smaller model remains an open question. Inspired by trial-to-trial variability in the brain, we introduce variability through noise at the input level, supervision signal from the teacher, or target

*Equal contribution.

level. We propose novel ways of injecting noise into the knowledge distillation framework as general and scalable techniques and exhaustively evaluate their performance.

Our contributions are as follows:

- We empirically show that noise can be used as a critical resource for learning in the knowledge distillation framework and evaluate the effect of injecting constructive noise at different levels of the framework.
- “Fickle Teacher" (FT), a novel approach to simulate trial-to-trial response variability of biological neural networks. The method exposes the student to the uncertainty of the teacher which results in consistent generalization improvement over the vanilla knowledge distillation method (from 94.28% to 94.67%).
- “Soft Randomization" (SR), a novel approach for increasing robustness to input variability. The method considerably increases the capacity of the model to learn robust features with even small additive noise with a minimal loss in generalization compared to Gaussian data augmentation (GA). On SVHN, SR achieves 93.39% generalization with 51.39% robustness compared to GA's 93.22% generalization and 25.94% robustness for $\sigma = 0.2$.
- “Messy Collaboration" (MC), an approach for using target variability as a strong deterrent to cognitive bias. We show the effectiveness of MC in learning with noisy labels.

2. Methodology

In this section, we provide details for the methods relevant to our study.

2.1. Knowledge Distillation.

Hinton *et al.* [23] proposed to use the final softmax function with a raised temperature and use the smooth logits of the teacher as soft targets for the student. The method involves minimizing the Kullback–Leibler divergence between the smoother output probabilities:

$$\mathcal{L} = (1-\alpha)\mathcal{L}_{CE}(S(x), y) + \alpha\tau^2 D_{KL}(S^\tau(x)||T^\tau(x)) \quad (1)$$

where $\mathcal{L}_{CE}$ denotes cross-entropy loss, τ and α are the hyperparameters which denote softmax temperature and balancing ratio. $S(.)$ denotes the softmax output of student, $S^\tau(.)$ and $T^\tau(.)$ denote the student's and teacher's softmax output with raised temperature, respectively.

2.2. Out-of-Distribution Generalization

Neural networks tend to generalize well when the test data comes from the same distribution as the training data [11, 20]. However, models in the real world often have to deal with some form of domain shift which adversely affects the generalization performance of the models [25, 33, 39, 49]. Therefore, test set performance alone is not the optimal metric for evaluating the generalization of the models in the test environment. To measure the out-of-distribution performance, we use the ImageNet images from the CINIC dataset [10]. CINIC contains 2100 images randomly selected for each of the CIFAR-10 categories from the ImageNet dataset. Hence, the performance of models trained on CIFAR-10 on these 21000 images can be considered as an approximation for a model's out-of-distribution generalization performance.

2.3. Adversarial Robustness

Deep Neural Networks are highly vulnerable to carefully crafted imperceptible perturbations designed to fool a neural network by an adversary [4, 53]. This vulnerability poses a real threat to the deep learning model's deployment in the real world [29]. Robustness to these adversarial attacks has therefore gained a lot of traction in the research community and progress has been made to better evaluate robustness to adversarial attacks [8, 16, 38] and defend the models against these attacks [36, 61].

To evaluate the adversarial robustness of models in this study, we use the Projected Gradient Descent (PGD) attack from Madry *et al.* [36]. The PGD attack initializes the adversarial image with the original image with the addition of a random noise within some epsilon bound, ϵ. For each step, it takes the loss with respect to the input image and moves in the direction of loss with the step size and then clips it within the epsilon bound and the range of the valid image. In all of our experiments, we use l_∞ attack with 0.031 epsilon bound, 0.03 step size. We use the notation PGD-N for an N steps PGD attack and report the worst performance of 5 random initialization runs.

2.4. Natural Robustness

While robustness to adversarial attack is important from a security perspective, it is an instance of a worst-case distribution shift. The model also needs to be robust to naturally occurring perturbations which it will encounter frequently in the test environment. Recent works have shown that Deep Neural Networks are also vulnerable to commonly occurring perturbations in the real world which are far from the adversarial examples manifold. Hendrycks *et al.* [22] curated a set of real-world, unmodified, and naturally occurring examples that cause classifier accuracy to degrade sharply. Gu *et al.* [18] measured model's robustness to the minute transformations found across video frames which they refer to as natural robustness and found state-of-the-art classifiers to be brittle to these transformations. In our study we use robustness to the common corruptions and perturbations proposed by Hendrycks *et al*. [21] in CIFAR-C as a proxy for natural robustness. We evaluate average robustness to the 19 distortions across 5 severity levels. Furthermore, we calculate mean Corruption Accuracy (mCA) over the 19 distortions. Following [18] we use a fine-grained measure of natural robustness, by computing accuracy on the corrupted image conditional on the clean image being classified correctly.

3. Experimental Setup

To study the effect of injecting noise in the knowledge distillation framework, we use Hinton method [23] which trains the student by minimizing the Kullback–Leibler divergence between the smoother output probabilities of the student and teacher. In all of our experiments we use the balancing parameter $\alpha = 0.9$ and softmax temperature $\tau = 4$ which are commonly used in knowledge distillation literature [55, 59]. We conduct our experiments on Wide Residual Networks (WRN) [60]. Unless otherwise stated, we normalize the images between 0 and 1 and use the standard training scheme as used in [55, 59]: SGD with 0.9 momentum; 200 epochs; batch size 128; and an initial learning rate of 0.1, decayed by a factor of 0.2 at epochs 60, 120 and 150. We conduct our experiments on CIFAR-10 [28] and SVHN [42], with WRN-40-2 with 2.2M parameters as the teacher, and WRN-16-2 with 0.7M parameters as the student because of their pervasiveness in literature. We also compare all the methods with the baseline which refers to WRN-16-2 trained alone with standard cross-entropy loss. In all of our experiments, we train each model for five different seed values. For the teacher, we select the model with the highest test accuracy and then use it to train the student again for five different seed values and report the mean and 1 std for our evaluation metrics. For a fair comparison, we train the knowledge distillation methods under the same experimental setup using the publicly available code.

4. Empirical study of Noises

In this section, we propose injecting different types of noise in the student-teacher collaborative learning framework and analyze their effect on the performance of the student.

4.1. Fickle Teacher

Trial-to-trial response variability in the brain can be considerably different across stimuli, suggesting that it could also provide an important contribution to the information conveyed by the neural responses about the stimuli [48]. Similarly, in deep neural networks, dropout [50] which randomly switches off a subgroup of hidden units results in response variability and can be used to obtain principled uncertainty estimates for an input image [15]. We, therefore, propose to use the response variability resulting from keeping the dropout in the teacher model active to simulate the trial-to-trial response variability in the brain. Fickle Teacher (FT) involves first training the teacher with dropout and in the subsequent step keeping the dropout active in the teacher while distilling knowledge to the student. This results in variability in the supervision signal from the teacher to the student for the same input across different epochs, thereby exposing the student to its uncertainty. It is important to note that the student itself does not use dropout. The teacher, on the other hand, not only uses dropout during its training but also keeps it active when providing supervision to the student in order to provide additional information about the uncertainty of its prediction on a particular data point.

We systematically change the dropout rate used for training the teacher and study its effect on the generalization performance of the student. Because of the variability in the teacher's supervision signal, the student needs to be trained for more epochs in order for it to converge and be effectively exposed to the uncertainty of the teacher. We use the same initial learning rate of 0.1 and a decay factor of 0.2 as per the standard training scheme. For a dropout rate of 0.1 and 0.2, we train for 250 epochs and reduce the learning rate at 75, 150, and 200 epochs. For dropout rate 0.3, we train for 300 epochs and reduce the learning rate at 90, 180, and 240 epochs. Finally for a dropout rate of 0.4 and 0.5, due to the increased variability, we train for 350 epochs and reduce the learning rate at 105, 210, and 280 epochs. We show the efficacy of our approach by comparing it with the state-of-the-art knowledge distillation methods under the same experimental settings. Following the parameters used in the paper, we use $\beta = 1000$ for Attention (AT) [59], $\gamma = 3000$ for Similarity Preserving (SP) [55] whereas for Relational Knowledge Distillation (RKD), we use $\lambda_{RKD-D} = 25$ and $\lambda_{RKD-A} = 50$.

Table 1 shows that FT improves the in-distribution and out-of-distribution generalization on CIFAR-10 as well as the robustness to common corruptions. Notably, even when the accuracy of the teacher decreases after a dropout rate of 0.2, the student accuracy still improves up to a dropout rate of 0.4. FT provides similar generalization gains for SVHN. For FT-0.5, the student even outperforms the teacher.

Furthermore, since it is a common practice to add Hinton loss on top of other distillation methods, Table 2 compares the effect of adding Hinton vs FT on top of the other distillation methods. The higher generalization gains with FT across all the distillation methods show that it better complements these methods compared to Hinton. For relational knowledge distillation methods (SP, all variants of RKD) where adding Hinton loss reduces performance, FT provides further improvement over the original method.

FT maintains the simplicity and versatility of the Hinton

Table 1. Fickle Teacher (FT) consistently improves both in-distribution and out-of-distribution (CINIC) generalization on CIFAR-10 as well as robustness to common corruptions (mCA). Similar generalization gain is observed on SVHN. The dropout rate, x, used for training the teacher is indicated by FT-x. The best performing models are in bold.

	CIFAR-10				**SVHN**	
Method	Teacher	Test Acc.	CNIC Acc.	mCA	Teacher	Test Acc.
Baseline	-	93.95±0.18	68.89±0.08	74.38±0.67	-	96.14±0.15
Hinton	95.11	94.28±0.09	69.13±0.29	74.57±0.29	96.78	96.80±0.08
AT[59]	95.11	94.50±0.18	69.23±0.18	74.70±0.58	96.78	96.28±0.13
SP[55]	95.11	94.64±0.17	69.39±0.32	74.93±0.43	96.78	96.61±0.06
RKD-D[43]	95.11	94.42±0.15	69.34±0.17	74.75±0.60	96.78	96.49±0.05
RKD-A[43]	95.11	94.62±0.14	69.57±0.17	**75.33±0.22**	96.78	96.57±0.06
RKD-DA[43]	95.11	94.52±0.11	69.43±0.23	74.93±0.43	96.78	96.58±0.03
FT-0.1	95.19	94.43±0.15	69.49±0.23	74.99±0.48	96.90	96.79±0.03
FT-0.2	**95.38**	94.46±0.16	69.59±0.13	74.61±0.41	96.85	96.74±0.06
FT-0.3	95.12	94.56±0.14	69.84±0.22	75.06±0.14	96.94	96.90±0.07
FT-0.4	95.18	**94.67±0.09**	69.50±0.21	75.09±0.51	96.95	96.93±0.07
FT-0.5	94.88	94.50±0.23	**69.95±0.25**	74.67±0.30	**97.00**	**97.09±0.02**

Table 2. Fickle Teacher (+FT) better complements the other distillation methods compared to Hinton (+H). We train the models on CIFAR-10 using the same experimental setup as for the original methods. We use FT-0.4 for all the +FT experiments. The best performing models are in bold.

	Test Acc.			CNIC Acc.		
Method	Original	+H	+FT	Original	+H	+FT
AT[59]	94.50±0.18	94.63±0.24	**94.83±0.05**	69.23±0.18	**69.57±0.13**	69.46±0.23
SP[55]	94.64±0.17	94.39±0.18	**94.66±0.09**	**69.39±0.32**	69.33±0.09	69.36±0.31
RKD-D[43]	94.42±0.15	94.39±0.15	**94.76±0.04**	69.34±0.17	69.21±0.11	**69.60±0.07**
RKD-A[43]	94.62±0.14	94.38±0.21	**94.68±0.06**	69.57±0.17	69.15±0.17	**69.68±0.33**
RKD-DA[43]	94.52±0.11	94.43±0.04	**94.59±0.14**	69.43±0.23	69.36±0.05	**69.62±0.17**

method and provides even higher generalization gains over the recently proposed knowledge distillation variants which not only adds more constraints to student training, hence limiting their versatility but also require a lot of parameter tuning. Also, FT can be easily added on top of other distillation methods to further improve the performance of the model. The effectiveness of FT in improving the generalization of the model motivates further exploration into techniques that add noise so that it encodes the uncertainty in the supervision signal.

4.2. Soft Randomization

Pinot *et al*. [45] show that the injection of noise drawn from an exponential family such as Gaussian or Laplace noise leads to guaranteed robustness to adversarial attack. However, this improved adversarial robustness comes at the cost of significant loss in generalization. Some studies even argue that there is an inherent trade-off between robustness and generalization and consider them as contradictory goals [24, 54]. Therefore, an important consideration for methods proposed to increase the adversarial robustness is to reduce this loss in generalization.

Since knowledge distillation provides an opportunity to combine multiple sources of information, we hypothesize that combining information from a teacher with high generalization while training the student to be robust to noisy input can reduce the trade-off. The extra supervision signal from the teacher acts as a regularizer which encourages the student to align its feature distribution with the teacher. This effectively adds a prior, encouraging the student to learn semantically relevant features which are robust to the spurious directions introduced by Gaussian noise. It can also be considered as distilling knowledge from the clean domain to the noisy domain.

To test the hypothesis, we propose a novel technique for improving robustness to input variability in the student which utilizes the teacher trained on clean data, to train the student on noisy data. Here, we minimize the dissimilarity between the student's distribution on noisy data with the teacher's distribution on clean data (Figure 1). Therefore, loss function for SR adapts the vanilla knowledge distillation loss (Eq. 1):

$$\mathcal{L} = (1-\alpha)\mathcal{L}_{CE}(S(x+\delta), y) + \alpha\tau^2 D_{KL}(S^{\tau}(x+\delta)||T^{\tau}(x)) \tag{2}$$

where $\delta \sim \mathcal{N}(0, \sigma^2)$ is white Gaussian noise.

We train Soft Randomization (SR) for both low noise and high noise intensities. We compare Soft Randomization to the compact model (WRN-16-2) trained alone with

Table 3. Soft Randomization (SR) consistently achieves higher in-distribution and out-of-distribution generalization on CIFAR-10 and for the majority of the noise intensities on SVHN compared to Gaussian Augmentation (GA). The best performing models are in bold.

Sigma	CIFAR-10 Test Acc.		CINIC Acc.		SVHN Test Acc.	
	GA	SR	GA	SR	GA	SR
0.01	93.80±0.25	**94.29±0.13**	69.08±0.21	**69.19±0.21**	96.14±0.11	**96.78±0.10**
0.02	93.65±0.10	**94.07±0.16**	68.98±0.14	**69.11±0.40**	96.29±0.10	**96.79±0.12**
0.03	93.14±0.20	**93.53±0.29**	68.59±0.22	**68.77±0.26**	96.24±0.08	**96.77±0.05**
0.04	92.67±0.10	**93.04±0.18**	68.05±0.19	**68.33±0.37**	96.08±0.11	**96.72±0.09**
0.05	92.14±0.26	**92.57±0.22**	67.52±0.16	**68.07±0.27**	95.93±0.16	**96.56±0.08**
0.1	89.09±0.26	**89.88±0.14**	64.89±0.20	**65.44±0.37**	95.55±0.22	**96.01±0.14**
0.2	83.41±0.21	**84.07±0.30**	59.31±0.20	**60.08±0.17**	93.22±0.20	**93.39±0.22**
0.3	78.00±0.49	**78.51±0.48**	54.55±0.34	**55.19±0.25**	**89.75±0.29**	89.70±0.36
0.4	72.88±0.46	**73.35±0.37**	50.30±0.18	**51.03±0.36**	**85.51±0.35**	85.08±0.31
0.5	68.39±0.45	**68.95±0.21**	46.75±0.31	**47.44±0.13**	**80.48±0.53**	79.69±0.83

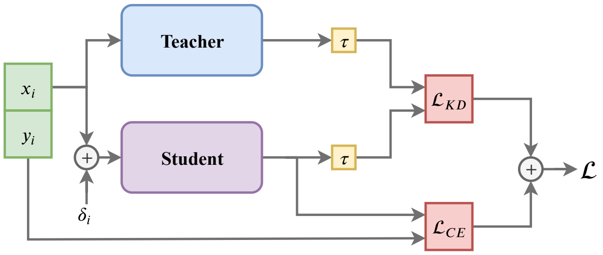


Figure 1. Soft Randomization uses supervision from a static teacher (trained on clean data) to train the student model effectively using Gaussian data augmentation. For training the student, SR minimizes the KL divergence between student's response on noisy input and teacher's response on the clean input in addition to minimizing the cross-entropy loss on the noisy input.

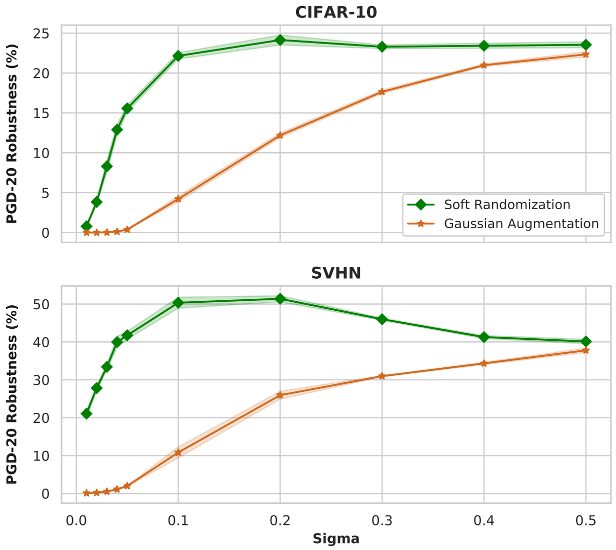


Figure 2. Soft Randomization consistently achieves higher adversarial robustness compared to Gaussian Augmentation. For lower noise intensities, Soft Randomization provides significantly higher robustness to PGD-20 attacks. We observe peak robustness at 0.2 where the model achieves both high generalization and robustness. Shaded regions show 1 std. For values, see Supplementary Material Table S1.

Gaussian data augmentation, referred to as Gaussian augmentation (GA). Figure 2 shows that SR consistently improves the adversarial robustness of the student over GA for both datasets. Especially for lower noise intensities, SR outperforms GA by a considerable margin, indicating that SR enables training a robust model even with low noise intensities which have the advantage of higher generalization performance as well. Table 3 shows the corresponding generalization performance of the models. To complement the robustness gains, SR consistently achieves better in-distribution and out-of-distribution generalization for all σ values compared to GA on CIFAR-10 and the majority of noise intensities on SVHN. For $\sigma = 0.05$, SR achieves 15.56% robustness to PGD-20 attack and 92.57% test accuracy compared to only 0.38% robustness and 92.14% test accuracy for GA on CIFAR-10. For SVHN, the difference is even more pronounced, with SR achieving 93.39% generalization with 51.39% robustness compared to GA's 93.22% generalization and 25.94% robustness for $\sigma = 0.2$.

To further analyze the robustness of SR, we evaluate the models on PGD attacks of varying strengths. Table 4 shows the effect of increasing the number of iterations for a fixed epsilon bound ($\epsilon = 8/255$). SR consistently provides higher robustness compared to GA across the PGD attacks of varying strengths and maintains its robustness level after 100 steps. To further show that gradient-based methods are indeed effective on the proposed method, Table 5 shows that increasing the allowed perturbation budget (epsilon) for a fixed number of iterations (20) effectively reduces the robustness of the models to 0. This follows the analysis suggested by Lamb *et al*. [30] and shows that SR does not suffer from gradient obfuscation [3].

Finally, Figure 3 shows the natural robustness of models trained on CIFAR-10 with SR. The mCA improves over the

Table 4. Comparison of Soft Randomization (SR) and Gaussian Augmentation (GA) on PGD attacks with fixed epsilon bound ($\epsilon = 8/255$) and an increasing number of iterations. SR consistently provides higher robustness against the PGD attacks of varying strengths. As expected, models trained with higher noise intensity, defend better against stronger attacks. The best performing models are in bold.

	σ		1	5	10	15	20	100	200	1000
CIFAR-10	0.02	GA	72.85±0.19	5.08±0.22	0.10±0.02	0.00±0.01	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
		SR	**78.15±0.17**	**39.06±0.63**	**18.02±0.85**	**8.07±0.37**	**3.83±0.28**	**0.11±0.05**	**0.06±0.04**	**0.03±0.03**
	0.05	GA	81.24±0.23	26.53±0.71	4.91±0.38	1.10±0.10	0.38±0.04	0.08±0.02	0.07±0.02	0.07±0.01
		SR	**83.20±0.19**	**49.74±0.34**	**31.30±0.51**	**21.18±0.39**	**15.56±0.44**	**3.80±0.69**	**2.80±0.62**	**1.91±0.53**
	0.2	GA	76.62±0.21	49.12±0.32	26.62±0.24	16.06±0.33	12.19±0.29	9.83±0.22	9.73±0.23	9.68±0.23
		SR	**78.05±0.22**	**53.64±0.42**	**35.92±0.48**	**28.05±0.65**	**24.14±0.63**	**16.87±0.57**	**16.36±0.58**	**15.98±0.52**
	0.5	GA	63.81±0.65	47.51±0.35	33.35±0.34	25.51±0.38	22.34±0.33	20.87±0.35	20.80±0.37	20.79±0.37
		SR	**64.53±0.32**	**48.11±0.27**	**34.16±0.20**	**26.52±0.38**	**23.54±0.37**	**21.85±0.29**	**21.81±0.27**	**21.79±0.27**
SVHN	0.02	GA	86.77±0.43	24.24±1.51	2.97±0.47	0.59±0.14	0.21±0.06	0.03±0.01	0.02±0.01	0.02±0.01
		SR	**90.92±0.32**	**67.50±0.66**	**49.11±0.79**	**35.87±0.83**	**27.82±0.83**	**6.20±0.44**	**4.32±0.36**	**2.71±0.30**
	0.05	GA	90.01±0.29	45.14±0.92	12.48±0.73	4.05±0.29	1.99±0.13	0.51±0.07	0.43±0.06	0.37±0.05
		SR	**92.38±0.21**	**72.74±0.60**	**58.21±0.69**	**47.97±0.91**	**41.75±1.06**	**17.72±0.97**	**14.76±0.90**	**11.68±0.88**
	0.2	GA	89.56±0.33	67.44±0.59	43.24±0.75	30.08±0.88	25.94±1.05	17.94±1.02	17.57±1.04	17.40±1.02
		SR	**89.80±0.24**	**71.85±0.43**	**59.11±0.67**	**52.23±0.88**	**51.39±0.80**	**36.51±0.74**	**35.33±0.70**	**34.48±0.69**
	0.5	GA	**76.41±0.54**	**60.48±0.50**	**44.43±0.48**	34.70±0.59	37.78±0.60	27.02±0.50	26.85±0.49	26.82±0.48
		SR	75.50±0.81	59.24±0.53	44.10±0.23	**35.74±0.22**	**40.13±0.61**	**28.07±0.32**	**27.96±0.33**	**27.94±0.33**

Table 5. Comparison of SR and GA on PGD attacks with a fixed number of steps (20) and increasing epsilon bounds. The robustness effectively goes to zero as the epsilon budget increases which shows that gradient-based attacks perform as expected on SR. SR consistently provides higher robustness against the PGD attacks of increasing strengths. The best performing models are in bold.

	σ		1	2	10	20	25	50	100	200
CIFAR-10	0.02	GA	60.27±0.59	19.60±0.54	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
		SR	**66.35±0.37**	**35.85±0.63**	**2.42±0.18**	**0.40±0.05**	**0.19±0.02**	**0.02±0.01**	0.00±0.00	0.00±0.00
	0.05	GA	74.51±0.35	48.26±0.71	0.14±0.02	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
		SR	**76.94±0.21**	**55.01±0.39**	**13.03±0.47**	**7.37±0.60**	**5.32±0.54**	**0.38±0.09**	**0.02±0.04**	0.00±0.00
	0.2	GA	73.74±0.18	62.56±0.13	7.57±0.18	3.09±0.11	2.53±0.11	1.23±0.12	0.11±0.03	0.00±0.00
		SR	**75.47±0.16**	**64.91±0.23**	**20.98±0.58**	**16.91±0.75**	**16.12±0.85**	**13.57±0.81**	**4.90±0.58**	**0.03±0.03**
	0.5	GA	62.14±0.65	55.73±0.50	16.99±0.37	9.32±0.34	8.17±0.27	6.03±0.27	3.84±0.20	0.55±0.05
		SR	**62.97±0.42**	**56.43±0.33**	**18.56±0.35**	**12.00±0.31**	**11.00±0.31**	**9.16±0.31**	**6.60±0.25**	**1.02±0.08**
SVHN	0.02	GA	77.80±0.73	40.52±1.66	0.08±0.03	0.01±0.01	0.01±0.00	0.00±0.00	0.00±0.00	0.00±0.00
		SR	**85.82±0.35**	**67.78±0.48**	**23.14±0.83**	**14.53±0.60**	**11.59±0.49**	**2.98±0.19**	**0.18±0.04**	0.00±0.00
	0.05	GA	85.59±0.40	63.31±0.64	0.88±0.11	0.19±0.04	0.12±0.03	0.02±0.01	0.00±0.00	0.00±0.00
		SR	**89.30±0.25**	**76.25±0.48**	**36.56±1.07**	**28.33±1.04**	**25.30±1.02**	**10.30±0.99**	**0.83±0.19**	0.00±0.00
	0.2	GA	87.71±0.32	79.01±0.53	17.47±1.01	9.11±0.91	7.87±0.86	5.11±0.74	1.96±0.35	0.03±0.01
		SR	**88.09±0.23**	**80.18±0.32**	**45.11±0.88**	**40.15±0.86**	**38.97±0.81**	**34.69±0.75**	**21.61±0.61**	**1.60±0.23**
	0.5	GA	**74.86±0.56**	**68.63±0.50**	23.41±0.50	13.80±0.38	12.32±0.28	9.38±0.20	6.57±0.16	1.49±0.05
		SR	73.85±0.84	67.37±0.66	**27.25±0.35**	**21.50±0.48**	**20.56±0.47**	**18.50±0.48**	**14.66±0.46**	**4.18±0.24**

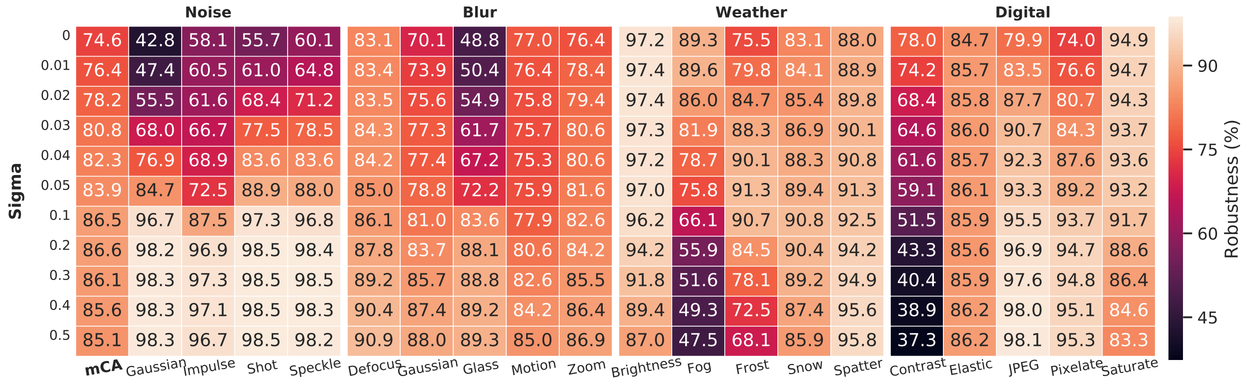


Figure 3. Soft Randomization consistently improves the average robustness to common corruptions (mCA). SR most notably improves the robustness to *noise* and *blurring distortions*.

Hinton method for all noise intensities. While robustness drops most notably for color distortions (*contrast*, *brightness* and *saturation*), robustness to *noise* and *blurring corruptions* improves as the Gaussian noise intensity increases. We also observe changes in the effect of different noise intensities. For *frost*, the robustness increases at a lower noise level and then decreases for higher intensities.

The empirical results show that SR increases the capacity of the student to learn robust features even with lower noise intensities. This allows the use of lower noise intensity for increasing adversarial robustness while keeping the loss in generalization much lower compared to the GA model with a comparable robustness level. This essentially provides a better trade-off between robustness and generalization, enabling the training of compact students with both high robustness and generalization. SR, hence, provides greater flexibility in finding a suitable trade-off based on the application. The empirical results suggest that adding noise in the input data is more effective in a collaborative learning framework compared to a model in isolation.

4.3. Messy Collaboration

Human decision-making shows systematic simplifications and deviations from the tenets of rationality ('heuristics') which may lead to sub-optimal decisional outcomes ('cognitive biases') [27]. We believe this cognitive bias is manifested in deep neural networks in the form of memorization [2] and over-generalization. This makes it even more challenging to learn efficiently on real-world datasets with some fraction of corrupted labels (mislabeled) which have been shown to adversely affect the model performance [14]. Furthermore, in order to utilize the vast amount of open-source data available, researchers have proposed methods to generate labels automatically using user tags and keywords. However, these methods lead to noisy labels. Considering the abundance of noisy labels, it is imperative to develop methods that can effectively learn from noisy labels.

We propose a counter-intuitive regularization technique based on target variability to mitigate cognitive bias and subsequently reduce memorization in DNNs. We term this technique as Messy Collaboration (MC). While distilling knowledge from the teacher, for each sample in the training batch, with rate r, we randomly change the true labels to random targets sampled uniformly from the number of the classes. The target variability is added independently for each batch. We hypothesize that the target variability[1] discourages memorization in DNNs and prevents overconfident predictions. Also, the soft targets from the teacher provide additional information about the similarities between the different classes which can mitigate the adverse effect of incorrect annotations and together with target variability improve the efficiency of DNNs to learn with noisy labels.

Here, we first show the effectiveness of knowledge distillation in learning with noisy labels at varying rates of label corruption on CIFAR-10 and SVHN and then further show that MC improves the generalization over vanilla knowledge distillation. Figure 4 shows that the generalization drops as the corruption rate increases (cf. Teacher and Baseline). For all corruption levels, knowledge distillation (r=0) improves the generalization performance and even outperforms the teacher on both datasets. The gain in generalization gets higher as the label corruption rate increases. It also shows the effect of varying the noise rate in MC for the different label corruption rates. On CIFAR-10, for label corruption rate 0.1 and higher, MC improves the generalization over the Baseline and teacher for all noise rates. For the majority of the corruption levels across the two datasets, MC further improves generalization over vanilla knowledge distillation. This shows that the target variability in MC makes the model more tolerant to label noise which allows efficient learning with noisy labels. The performance gain with MC over Hinton is less pronounced for SVHN, possibly because of the ease of the task. Figure 5 shows similar performance gains in out-of-distribution generalization as in-distribution for models trained on CIFAR-10.

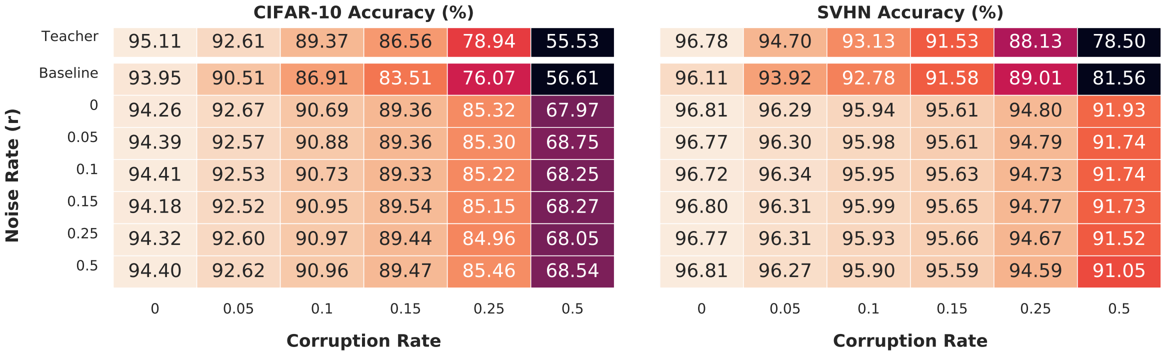
The empirical results confirm that knowledge distillation is an effective framework for training models under noisy labels and provides consistent performance gain over models trained alone with cross-entropy. The further improvement with MC shows that target variability can discourage memorization in DNNs. This suggests that using noise in the target labels as a regularizer against memorization and over-confident predictions in DNNs is a promising direction.

5. Related Work

A number of experimental and computational methods have reported the presence of noise in the nervous system and how it affects the function of the system [13, 48]. Analogously, noise has been shown to improve the training and performance of DNNs [17, 32, 50, 56, 58, 63]. Furthermore, a family of randomization techniques that inject noise in the model both during training and inference time is proven to be effective to the adversarial attacks [12, 34, 46, 57]. Randomized smoothing transforms any classifier into a new smooth classifier that has certifiable l_2-norm robustness guarantees [9, 31]. Label smoothing improves the performance of deep neural networks across a range of tasks [44, 52].

Knowledge Distillation [23] has proven to be an effective framework for training compact models with the help

[1] We use the term *target variability* to refer to the random label corruption which we are introducing intentionally for each batch during training. Whereas, *noisy labels* refers to the inherent corruption in the labels which comes from incorrect annotations.



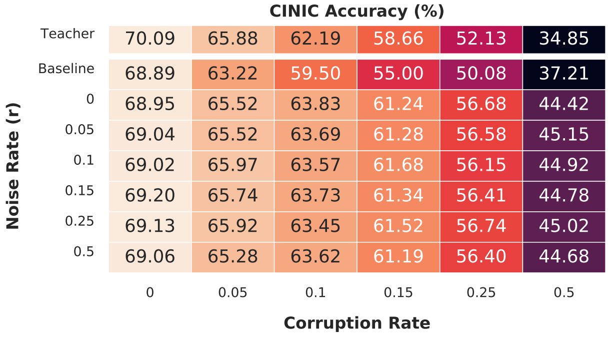
CIFAR-10 Accuracy (%)

Noise Rate (r) \ Corruption Rate	0	0.05	0.1	0.15	0.25	0.5
Teacher	95.11	92.61	89.37	86.56	78.94	55.53
Baseline	93.95	90.51	86.91	83.51	76.07	56.61
0	94.26	92.67	90.69	89.36	85.32	67.97
0.05	94.39	92.57	90.88	89.36	85.30	68.75
0.1	94.41	92.53	90.73	89.33	85.22	68.25
0.15	94.18	92.52	90.95	89.54	85.15	68.27
0.25	94.32	92.60	90.97	89.44	84.96	68.05
0.5	94.40	92.62	90.96	89.47	85.46	68.54

SVHN Accuracy (%)

Noise Rate (r) \ Corruption Rate	0	0.05	0.1	0.15	0.25	0.5
Teacher	96.78	94.70	93.13	91.53	88.13	78.50
Baseline	96.11	93.92	92.78	91.58	89.01	81.56
0	96.81	96.29	95.94	95.61	94.80	91.93
0.05	96.77	96.30	95.98	95.61	94.79	91.74
0.1	96.72	96.34	95.95	95.63	94.73	91.74
0.15	96.80	96.31	95.99	95.65	94.77	91.73
0.25	96.77	96.31	95.93	95.66	94.67	91.52
0.5	96.81	96.27	95.90	95.59	94.59	91.05

Figure 4. Generalization performance of Messy Collaboration (MC) with varying noise rates trained on corrupted labels. Knowledge distillation provides considerable generation gains compared to the Baseline and Teacher. MC, further, improves the generalization. For standard deviations, see Supplementary Material Tables S2 and S3.



CINIC Accuracy (%)

Noise Rate (r) \ Corruption Rate	0	0.05	0.1	0.15	0.25	0.5
Teacher	70.09	65.88	62.19	58.66	52.13	34.85
Baseline	68.89	63.22	59.50	55.00	50.08	37.21
0	68.95	65.52	63.83	61.24	56.68	44.42
0.05	69.04	65.52	63.69	61.28	56.58	45.15
0.1	69.02	65.97	63.57	61.68	56.15	44.92
0.15	69.20	65.74	63.73	61.34	56.41	44.78
0.25	69.13	65.92	63.45	61.52	56.74	45.02
0.5	69.06	65.28	63.62	61.19	56.40	44.68

Figure 5. In addition to in-distribution generalization, Messy Collaboration also improves the out-of-distribution generalization over the Teacher and vanilla knowledge distillation under corrupted labels. For standard deviations, see Supplementary Material Table S4.

of additional supervision signals from a larger static pretrained model. A number of modifications have been proposed to the original formulation. AT [59] proposed attention, defined as a set of spatial maps, as a mechanism for transferring knowledge to the student. RKD [43] encourages the student to form the same relational structure in the output representation space with that of the teacher using two potential functions: RKD-D measures the euclidean distance between two data samples while RKD-A measures the angle formed by the three data samples. RKD-DA is a combination of both losses. SP [55] preserves the pairwise similarities so that similar/dissimilar activations in the teacher produce similar/dissimilar activations in the student. While different ways of distilling knowledge to the student have been extensively studied [47], the role of noise in the knowledge distillation framework is not well studied. Interestingly, Muller *et al*. [40] report that label smoothing impairs knowledge distillation. On the contrary, we show that our biologically inspired technique of injecting noise into knowledge distillation, *Fickle Teacher*, consistently improves the generalization of the student. Furthermore, *Fickle Teacher* differs from the works of Bulo *et al*. [7] and Gurau *et al*. [19] in that instead of using the soft target distribution obtained by averaging Monte Carlo samples, we use the logits of the teacher model with dropout active directly as a source of uncertainty encoding noise for distilling knowledge to a compact student.

6. Conclusion

We demonstrated that noise can be used as a resource for learning in the knowledge distillation framework by injecting noise at multiple levels and extensively evaluating its effects on the performance of the model. Inspired by trial-to-trial variability in the brain which can result from multiple noise sources, we introduced *Fickle Teacher* which exposes the student to its uncertainty using dropout. We show that the variability in the supervision signal improves both in-distribution and out-of-distribution generalization. We further proposed *Soft Randomization* which utilizes input noise into the training of the student model. It involves matching the output distribution of the student on noisy data to the output distribution of the teacher on clean data. This considerably increases the capacity of the student to learn robust features and provides considerably higher adversarial robustness compared to the model trained alone with Gaussian data augmentation for lower noise intensities while also keeping the loss in generalization minimal. Finally, *Messy Collaboration* employs target variability to improve the effectiveness of the model to learn under noisy labels. Our empirical results suggest that the use of noise in a collaborative learning framework is a promising direction and warrants further investigation.

References

[1] Guozhong An. The effects of adding noise during backpropagation training on a generalization performance. *Neural*

computation, 8(3):643–674, 1996.

[2] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 233–242. JMLR. org, 2017.

[3] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[4] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

[5] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

[6] Léon Bottou. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8):12, 1991.

[7] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. Dropout distillation. In *International Conference on Machine Learning*, pages 99–107, 2016.

[8] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[9] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019.

[10] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[12] Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.

[13] A Aldo Faisal, Luc PJ Selen, and Daniel M Wolpert. Noise in the nervous system. *Nature reviews neuroscience*, 9(4):292, 2008.

[14] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

[15] Y Gal and Z Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. arxiv, 2015.

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[17] Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011.

[18] Keren Gu, Brandon Yang, Jiquan Ngiam, Quoc Le, and Jonathan Shlens. Using videos to evaluate image model robustness. *arXiv preprint arXiv:1904.10076*, 2019.

[19] Corina Gurau, Alex Bewley, and Ingmar Posner. Dropout distillation for efficiently estimating model confidence. *arXiv preprint arXiv:1809.10562*, 2018.

[20] K He, X Zhang, S Ren, and J Sun. Deep residual learning for image recognition. computer vision and pattern recognition (cvpr). In *2016 IEEE Conference on*, volume 5, page 6, 2015.

[21] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[22] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019.

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

[24] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

[25] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 2017.

[26] Robert Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does sgd escape local minima? *arXiv preprint arXiv:1802.06175*, 2018.

[27] Johan E Korteling, Anne-Marie Brouwer, and Alexander Toet. A neural network framework for cognitive bias. *Frontiers in psychology*, 9, 2018.

[28] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[29] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[30] Alex Lamb, Vikas Verma, Juho Kannala, and Yoshua Bengio. Interpolated adversarial training: Achieving robust neural networks without sacrificing accuracy. *arXiv preprint arXiv:1906.06784*, 2019.

[31] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. *arXiv preprint arXiv:1802.03471*, 2018.

[32] Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems*, pages 597–607, 2017.

[33] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[34] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–385, 2018.

[35] Wolfgang Maass. Noise as a resource for computation and learning in networks of spiking neurons. *Proceedings of the IEEE*, 102(5):860–880, 2014.

[36] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[37] Mark D McDonnell and Lawrence M Ward. The benefits of noise in neural systems: bridging theory and experiment. *Nature Reviews Neuroscience*, 12(7):415–425, 2011.

[38] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[39] Jose G Moreno-Torres, Troy Raeder, RocíO Alaiz-RodríGuez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012.

[40] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? *arXiv preprint arXiv:1906.02629*, 2019.

[41] Arvind Neelakantan, Luke Vilnis, Quoc V Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*, 2015.

[42] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *In NeurIPS workshop on deep learning and unsupervised feature learning*, 2011.

[43] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[44] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

[45] Rafael Pinot, Laurent Meunier, Alexandre Araujo, Hisashi Kashima, Florian Yger, Cédric Gouy-Pailler, and Jamal Atif. Theoretical evidence for adversarial robustness through randomization: the case of the exponential family. *arXiv preprint arXiv:1902.01148*, 2019.

[46] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. *arXiv preprint arXiv:1811.09310*, 2018.

[47] Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. Knowledge distillation beyond model compression. *arXiv preprint arXiv:2007.01922*, 2020.

[48] Alessandro Scaglione, Karen A Moxon, Juan Aguilar, and Guglielmo Foffani. Trial-to-trial variability in the responses of neurons carries information about stimulus location in the rat whisker thalamus. *Proceedings of the National Academy of Sciences*, 108(36):14956–14961, 2011.

[49] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[50] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[51] Mark Steijvers and Peter Grünwald. A recurrent network that performs a context-sensitive prediction task. In *Proceedings of the 18th annual conference of the cognitive science society*, pages 335–339, 1996.

[52] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[53] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[54] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[55] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. *arXiv preprint arXiv:1907.09682*, 2019.

[56] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066, 2013.

[57] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

[58] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017.

[59] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[60] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[61] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*, 2019.

[62] Mo Zhou, Tianyi Liu, Yan Li, Dachao Lin, Enlu Zhou, and Tuo Zhao. Towards understanding the importance of noise in training neural networks. *arXiv preprint arXiv:1909.03172*, 2019.

[63] Zhengyuan Zhou, Panayotis Mertikopoulos, Nicholas Bambos, Stephen Boyd, and Peter W Glynn. Stochastic mirror descent in variationally coherent optimization problems. In *Advances in Neural Information Processing Systems*, pages 7040–7049, 2017.