Hydra: Preserving Ensemble Diversity for Model Distillation

Linh Tran [1 *] **Bastiaan S. Veeling** [2 3 *] **Kevin Roth** [4 *] **Jakub Świątkowski** [5 *] **Joshua V. Dillon** [3]
Stephan Mandt [6 *] **Jasper Snoek** [3] **Tim Salimans** [3] **Sebastian Nowozin** [3] **Rodolphe Jenatton** [3]

Abstract

Recent research has focused on *distilling* ensembles into a single compact model, reducing the burden of computation and memory of the ensemble while trying to preserve its predictive behavior. Most existing distillation formulations summarize the ensemble by capturing its *average* predictions. As a consequence, the *diversity* of the ensemble predictions is lost and the distilled model cannot provide a measure of uncertainty comparable to that of the original ensemble. To retain the diversity of the ensemble more faithfully, we propose a distillation method based on a single multi-headed neural network which we refer to as *Hydra*. We demonstrate that with a slight increase in parameter count, Hydra improves distillation performance on classification and regression settings while capturing the uncertainty behavior of the original ensemble over both in-domain and out-of-distribution tasks.

1. Introduction

Deep neural networks have achieved impressive performance, however, they tend to make over-confident predictions and poorly quantify uncertainty (Lakshminarayanan et al., 2017). It has been demonstrated that ensembles of models improve predictive performance and offer higher quality uncertainty quantification (Dietterich, 2000; Lakshminarayanan et al., 2017; Ovadia et al., 2019). A fundamental limitation of ensembles is the cost of computation and memory at evaluation time. A popular solution is to distill an ensemble of models into a single compact network by attempting to match the average predictions of the original ensemble. This idea goes back to the foundational work of Hinton et al. (2015), itself inspired by earlier ideas developed by (Bucilua et al., 2006). While this process has led to simple and well-performing algorithms, it fails to take into account the intrinsic diversity of the predictions of the ensemble, as represented by the individual predictions of each of its members. In particular, this diversity is all the more important in tasks that hinge on the uncertainty output of the ensemble, e.g., in out-of-distribution scenarios (Lakshminarayanan et al., 2017; Ovadia et al., 2019).

Inspired by multi-headed architectures already widely applied in various applications (Szegedy et al., 2015; Sercu et al., 2016; Osband et al., 2016; Song & Chai, 2018), we propose a multi-headed model to distill ensembles. Our multi-headed approach—which we name *Hydra*—can be seen as an interpolation between the full ensemble of models and the knowledge distillation proposed by (Hinton et al., 2015). Our distillation model is comprised of (1) a single *body* and (2) as many *heads* as there are members in the original ensemble. The heads share the same body network whose role is to provide a common feature representation. Each head is assigned to an ensemble member and tries to mimic the *individual predictions* of this ensemble member, as illustrated in Figure 1. The design of the body and the heads makes it possible to trade off the computational and memory efficiency against the fidelity with which the diversity of the ensemble is retained. We show through experimental evaluation that Hydra outperforms existing distillation methods for both classification and regression tasks w.r.t. predictive test performance. Further, we investigate Hydra's behavior in terms of in-domain and out-of-distribution data and demonstrate that Hydra comes closest to the ensemble behavior in comparison to existing distillation methods.

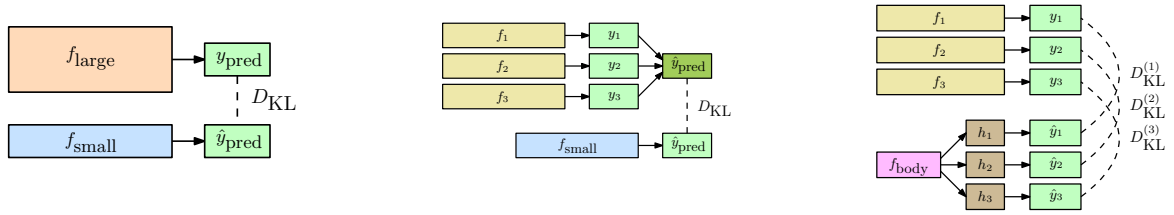
To the best of our knowledge, our approach is the first to employ a multi-headed architecture in the context of ensemble distillation. It is simple to implement, does not make strong parametric assumptions, requires few modifications to the distilled ensemble model and works well in practice, thereby making it attractive to apply to a wide range of ensemble models and tasks.

2. Hydra: A Multi-Headed Approach

With a focus on offline distillation, our goal is to train a student network to match the predictive distri-

[*]Work done while at Google [1]Imperial College London [2]University of Amsterdam [3]Google Research [4]ETH Zurich [5]University of Warsaw [6]University of California, Irvine. Correspondence to: Linh Tran <linh.tran@imperial.ac.uk>.

Presented at the ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning.



(a) Knowledge distillation (b) Distillation of ensembles (c) Hydra distillation

Figure 1. Existing distillation methods compared to Hydra. *Knowledge distillation*, (Hinton et al., 2015), trains a distillation network to imitate the prediction of a larger network. Applying knowledge distillation to ensemble models (Hinton et al., 2015) train a network to imitate the average ensemble prediction. *Hydra* instead learns to distill the individual predictions of each ensemble member into separate light-weight head models while amortizing the computation through a shared heavy-weight body network. This retains the diversity of ensemble member predictions which is otherwise lost in knowledge distillation.

bution of the teacher models, which is an ensemble of (deep) neural networks. Formally, given a dataset $\mathcal{D} = \{(\mathrm{x}^{(i)}, \mathrm{y}^{(i)})\}_{i=1}^{N}$, we consider an ensemble of M models' parameters $\theta_{\text{ens}} = \{\theta_{\text{ens},m}\}_{m=1}^{M}$ and prediction outputs $\{p(\mathrm{y}^{(i)}|\mathrm{x}^{(i)}; \theta_{\text{ens},m})\}_{m=1}^{M}$.

Hydra builds upon the approach of knowledge distillation and extends it to a multi-headed student model. Hydra is defined as a (deep) neural network with a single body and M heads, and has as many heads as there are ensemble members. The distillation model is parametrized by $\theta_{\text{hydra}} = \{\theta_{\text{body}}, \{\theta_{\text{head},m}\}_{m=1}^{M}\}$ in which the body, θ_{body}, is shared among all heads $\{\theta_{\text{head},m}\}_{m=1}^{M}$. The objective is to minimize the average KL divergence between each head m and corresponding ensemble member m. We differentiate between two tasks, classification and regression.

Classification. For classification tasks, the ensemble of models has access to $\mathcal{D}$ during training, with each x belonging to one of C classes, i.e., $\mathrm{y} \in \{1, 2, \ldots, C\}$. Assuming $\mathrm{z}_m = f_{\theta_{\text{head},m}}(f_{\theta_{\text{body}}}(\mathrm{x}))$ corresponds to the logits, the categorical distribution for a sample x over a class c is computed as $p(c|\mathrm{x}) = \frac{\exp(\mathrm{z}_{m,c}/T)}{\sum_{j=1}^{C}\exp(\mathrm{z}_{m,j}/T)}$, where T is a temperature re-scaling the logits. As discussed in (Hinton et al., 2015; Malinin et al., 2019) the distribution of the teacher network is often "sharp", which can limit the common support between the output distribution of the model and the target empirical distribution. To alleviate this issue, we follow the common practice (Hinton et al., 2015; Song & Chai, 2018; Lan et al., 2018) to use temperature to "heat up" both distributions and increase common support during training. The soft probability distributions are used to match the teacher ensemble of models by minimizing the average KL divergence between each head m and ensemble model m:

$$\mathcal{L} = \frac{T^2}{M}\sum_{m=1}^{M}\mathrm{KL}\big(p(\mathrm{y}|\mathrm{x};\theta_{\text{ens},m}) \parallel p(\mathrm{y}|\mathrm{x};\theta_{\text{body}},\theta_{\text{head},m})\big) \tag{1}$$

$$= -\frac{T^2}{M}\sum_{m=1}^{M} p(\mathrm{y}|\mathrm{x};\theta_{\text{ens},m})\log p(\mathrm{y}|\mathrm{x};\theta_{\text{body}},\theta_{\text{head},m}), \tag{2}$$

where the final line is reduced to the standard cross entropy loss by ignoring the constant entropy terms. We scale our objective by T^2 as the gradient magnitudes produced by the soft targets are scaled by $1/T^2$. By multiplying the loss term by a factor of T^2 we ensure that the relative contributions to additional regularization losses remain roughly unchanged (Song & Chai, 2018; Lan et al., 2018).

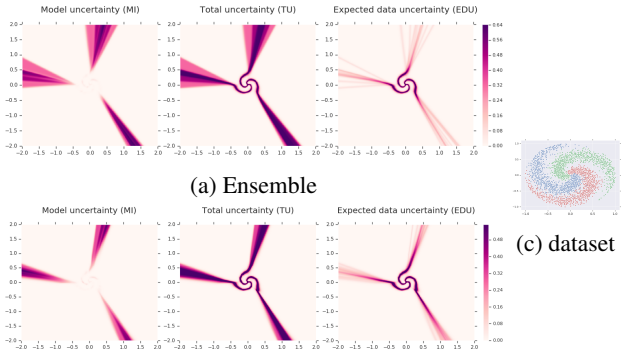
Regression. We focus on heteroscedastic regression tasks where each ensemble member m outputs a mean μ_m and σ_m^2 given an input x. The output is modeled as $p(\mathrm{y}|x,\theta_m) = \mathcal{N}(\mu_m, \sigma_m^2)$ for a given head m and the ensemble of models are trained by minimizing the negative log-likelihood. Traditional knowledge distillation matches a single Gaussian ("student") outputting μ_{distill} and $\sigma^2_{\text{distill}}$ to a mixture of Gaussians (a "teacher" ensemble). With Hydra, each head m outputs a mean $\mu_{\text{hydra},m}$ and variance $\sigma^2_{\text{hydra},m}$ and optimizes the KL divergence between each head output and corresponding ensemble member output:

$$\mathcal{L} = \frac{1}{M}\sum_{m=1}^{M}\mathrm{KL}\Big(\mathcal{N}(\mu_m,\sigma_m^{2,(i)}) \parallel \mathcal{N}(\mu_{\text{hydra},m},\sigma^2_{\text{hydra},m})\Big) \tag{3}$$

$$= -\frac{1}{M}\sum_{m=1}^{M}\frac{\sigma_m^2 + (\mu_m - \mu_{\text{distill}})^2}{2\sigma^2_{\text{distill}}} + \frac{1}{2}\log(2\pi\sigma^2_{\text{distill}}), \tag{4}$$

where the final line uses the fact that each KL term has an analytical solution.

Training with multi-head growth. Hydra is trained in two phases. First, Hydra mimics knowledge distillation in that it is trained until convergence with a single head—the "Hinton head"—to match the average predictions of the ensemble. Hydra is then extended by $M-1$ heads, all of which initialized with the parameter values of the "Hinton head". The resulting M heads are finally further trained to match the individual predictions of the M ensemble members. In practice, we sometimes experienced difficulties for Hydra to converge in absence of this initialization scheme and this two-phase training scheme led to overall quicker conver-



(a) Ensemble

(c) dataset

(b) Hydra

Figure 2. Model uncertainty, total uncertainty and expected data uncertainty applied on in-domain and out-of-distribution data for both (a) ensemble and (b). The original dataset is visualized (c), where each color corresponds to a single class.

gence.

3. Evaluation

We demonstrate that Hydra not only best matches the behavior of the teacher ensemble in terms of uncertainty quantification but also improves the predictive performance compared to existing distillation approaches over both classification and regression tasks.

Experimental settings We use a spiral toy dataset for visualizing and explaining model uncertainty, MNIST and CIFAR10 for classification and the standard regression datasets from the UCI repository (Asuncion & Newman, 2007) for regression. For evaluating CIFAR10, we also used cyclic translated test data. For the toy dataset we trained a 10-ensemble multi-layer perceptron (MLP) model. Following the settings of (Ovadia et al., 2019), we trained ensembles of 50 MLP for MNIST and ensembles of 50 ResNet-20 V1 models for CIFAR10. For all regression tasks, we used an ensemble of 50 MLPs. We compare our work with two core distillation approaches, Knowledge Distillation (Hinton et al., 2015) and Prior Networks (Malinin et al., 2019; Malinin & Gales, 2018). All baseline models have the same architecture as the ensemble one for distillation. For MNIST, Hydra uses the original ensemble member architecture and adds an MLP with two hidden layers of 100 units each as head. For CIFAR10, the original Resnet20 V1 model without the last residual block was used as body. For both classification and regression, we evaluate the negative log likelihood (NLL), Brier score and model uncertainty (MU) (Depeweg et al., 2017; Malinin et al., 2019).

Model capacity and efficiency The ensemble for MNIST has $9,960,500$ parameters and $9.0 \cdot 10^7$ floating point operations (FLOPs). Both Knowledge Distillation and Prior Networks use distillation models with $199,210$ parameters and $1.8 \cdot 10^8$ FLOPs, which amount to 2% of the ensemble model for both parameter count and FLOPs. Hydra's model

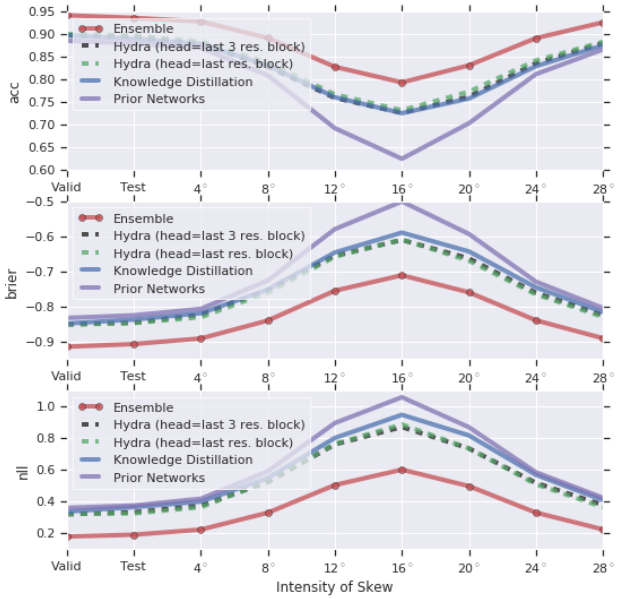


Figure 3. Model evaluation metrics plotted against intensity of distributional shift for CIFAR10. The plots contain all distillation models as well as the original ensemble of models. We observe that Hydra consistently more accurately captures the uncertainty of the ensemble.

has $1,757,700$ parameters and requires $2.5 \cdot 10^7$ FLOPs, reducing the original ensemble model to 17.6% of its parameters and 27.7% of its FLOPs. For CIFAR10, the ensemble model has $13,722,100$ parameters and requires $4.09 \cdot 10^{10}$ FLOPs. Both Knowledge Distillation and Prior Networks models take up to 2% of both parameter count ($274,442$) and FLOPs ($8.18 \cdot 10^8$). In contrast, Hydra has a higher parameter count of $3,950,324$ (28.7%) and a higher FLOPs count of $5.45 \cdot 10^9$ (13.3%).

Uncertainty Quantification We assess Hydra's ability to distill uncertainty metrics from an ensemble on classification tasks with the toy spiral dataset and CIFAR10. One way to quantify uncertainty is through model uncertainty (Depeweg et al., 2017; Malinin et al., 2019) which measures the spread or disagreement of an ensemble. It can be expressed as the difference of the total uncertainty and the expected data uncertainty, where total uncertainty is the entropy of the expected predictive distribution and expected data uncertainty is the expected entropy of individual predictive distribution. The total uncertainty will be high whenever the model is uncertain – both in regions of severe class overlap and out-of-distribution. However, for out-of-distribution data the estimates of expected data uncertainty are poor, resulting in high model uncertainty. Figure 2 visualizes the model uncertainty and its decomposition for the spiral toy dataset. Hydra successfully models uncertainty and its decomposition, although with a slight decrease in scale. We observe, as expected, a low model uncertainty where classes overlap due to both high total uncertainty and expected data uncertainty, and high model uncertainty where at the border

Model	ACC ↑	NLL ↓	BS ↓	MU
Ensemble ($M = 50$)	0.9851	0.0439	-0.9780	$9.97 \cdot 10^{-6}$
Prior Networks ((Malinin et al., 2019))	0.9842	0.0521	-0.9285	0.1158
Knowledge distillation ((Hinton et al., 2015))	0.9843	0.0497	-0.9764	N/A
Hydra (head = [100,100,10])	**0.9857**	**0.0465**	**-0.9776**	$\mathbf{2.28 \cdot 10^{-5}}$

(a) MNIST

Model	ACC ↑	NLL ↓	BS ↓	MU
Ensemble ($M = 50$)	0.9226	0.2392	-0.9033	0.1055
Prior Networks ((Malinin et al., 2019))	0.8731	0.4392	-0.8231	**0.0280**
Knowledge distillation ((Hinton et al., 2015))	0.8933	0.3598	-0.8373	N/A
Hydra (head = last res. block)	**0.8992**	**0.3179**	**-0.8468**	0.0074

(b) CIFAR10

Table 1. Average ($n = 3$) test performance for different baselines and Hydra for MNIST and CIFAR10. For all models, classification accuracy (ACC), negative log-likelihood (NLL), Brier score (BS) and model uncertainty (MU) are reported. Bold numbers represent best performance w.r.t. the specific evaluation metric (columns) across distillation models (rows) except for MU, where we report the model uncertainty closest to the ensemble one.

of in-domain and out-of-distribution data.

We plot all evaluation metrics against the intensity of skew of cyclic translated CIFAR10 in Figure 3 to evaluate Hydra w.r.t. in-domain and out-of-distribution behavior. Figure 3 shows that Hydra best matches the behavior of the ensemble in terms of accuracy, Brier score and NLL. As expected, Hydra with a larger head configuration even improves on overall performance compared to a smaller head-sized Hydra and the baseline comparisons.

Classification Performance on MNIST and CIFAR10. We report all metrics for MNIST in Table 1a and for CIFAR10 in Table 1b. For MNIST all distillation methods match the accuracy of the target ensemble with Hydra outperforming both knowledge distillation and prior networks in terms of capturing the ensemble uncertainty, almost matching the ensemble predictive NLL and Brier score. For CIFAR10, all distillation methods retain a gap in NLL performance compared to the ensemble, but Hydra has the smallest accuracy gap and a significantly smaller NLL compared to Knowledge distillation and Prior networks. In-distribution model uncertainty (MU) is comparable for both Prior Networks and Hydra but smaller compared to target ensemble MU, meaning it is possible to improve uncertainty quantification in all distillation methods tested.

Regression Performance on UCI Regression Datasets. We trained both Knowledge Distillation (Hinton et al., 2015) and Hydra on standard regression UCI datasets shown in Table 2. Here, Prior Network is not applicable because for probabilistic regression we cannot take averages of distributions. For regression Hydra outperforms knowledge distillation w.r.t. predictive performance (NLL) because

Dataset	Ensemble	Prior Network (Malinin et al., 2019)	Knowledge distillation (Hinton et al., 2015)	Hydra (head = [10, 1])
bost	2.3780	N/A	2.3893	**2.3805**
concr	3.0585	N/A	3.1231	**3.0982**
ener	1.2756	N/A	1.5236	**1.4402**
kin8	-1.2977	N/A	-1.2343	**-1.2555**
nava	-7.5983	N/A	-6.4340	**-7.1987**
powe	2.8861	N/A	2.8940	**2.8921**
prot	2.8272	N/A	2.8970	**2.8829**
wine	0.9111	N/A	**0.9112**	0.9113
yach	-0.1640	N/A	0.3837	**0.3489**

Table 2. UCI regression benchmark (Dua & Taniskidou, 2017). Average ($n = 3$) test negative log-likelihood (NLL) of the 9 different datasets considered. As Prior Networks (Malinin et al., 2019) cannot be applied to regression tasks, we denote this with "N/A".

Hydra produces a more flexible output in the form of a Gaussian mixture model, whereas Knowledge Distillation can produce only a single Gaussian component.

4. Related Work

In (Hinton et al., 2015), a “student" network is trained to match the *average* predictions of the “teacher" network(s). This methodology has been later successfully applied to the distillation of Bayesian ensembles (Balan et al., 2015). A parallel line of research has focused on *co-distillation*, also known as *online distillation*, to further reduce overall training cost (Zhang et al., 2018; Anil et al., 2018; Lan et al., 2018). Distillation has recently been the topic of theoretical analysis to better explain its empirical success (Lopez-Paz et al., 2015; Phuong & Lampert, 2019). Closest to our approach is the work of Lan et al. (2018) which consists in training multiple student models whose combined predictions induce an ensemble teacher model. While we share conceptual similarities with their work, we depart from their formulations in several ways. First, we focus on the offline ensemble setting (Hinton et al., 2015) where we start from a pre-defined ensemble whose training may be difficult to replicate inside a co-distillation process. Second, our approach follows a different goal: we consider multiple branches to *individually match the behavior of each teacher model*. Third, our methodology has a conceivably simpler design, as reflected by our single-component objective function and the absence of a learned gating mechanism.

5. Conclusion

We presented *Hydra*, a simple and effective approach to distillation for ensemble models. Hydra preserves diversity in ensemble member predictions and we have demonstrated on standard models that capturing this information translates into improved performance and better uncertainty quantification. While Hydra improves on previous approaches we believe that we can further improve distillation performance by leveraging techniques from fields studying sets of related learning such as meta-learning and domain adaptation.

References

Anil, R., Pereyra, G., Passos, A., Ormandi, R., Dahl, G. E., and Hinton, G. E. Large scale distributed neural network training through online distillation. *preprint arXiv:1804.03235*, 2018.

Asuncion, A. and Newman, D. Uci machine learning repository, 2007.

Balan, A. K., Rathod, V., Murphy, K. P., and Welling, M. Bayesian dark knowledge. In *Advances in Neural Information Processing Systems*, pp. 3438–3446, 2015.

Bucilua, C., Caruana, R., and Niculescu-Mizil, A. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541. ACM, 2006.

Depeweg, S., Hernández-Lobato, J. M., Doshi-Velez, F., and Udluft, S. Decomposition of uncertainty for active learning and reliable reinforcement learning in stochastic systems. *stat*, 1050:11, 2017.

Dietterich, T. G. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pp. 1–15. Springer, 2000.

Dua, D. and Taniskidou, E. K. UCI machine learning repository. *University of California, Irvine, School of Information and Computer Sciences*, 2017.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *preprint arXiv:1503.02531*, 2015.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

Lan, X., Zhu, X., and Gong, S. Knowledge distillation by on-the-fly native ensemble. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7528–7538, 2018.

Lopez-Paz, D., Bottou, L., Schölkopf, B., and Vapnik, V. Unifying distillation and privileged information. *preprint arXiv:1511.03643*, 2015.

Malinin, A. and Gales, M. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, pp. 7047–7058, 2018.

Malinin, A., Mlodozeniec, B., and Gales, M. Ensemble distribution distillation. *preprint arXiv:1905.00076*, 2019.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems*, pp. 4026–4034, 2016.

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

Phuong, M. and Lampert, C. Towards understanding knowledge distillation. In *International Conference on Machine Learning*, pp. 5142–5151, 2019.

Sercu, T., Puhrsch, C., Kingsbury, B., and LeCun, Y. Very deep multilingual convolutional neural networks for lvcsr. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4955–4959. IEEE, 2016.

Song, G. and Chai, W. Collaborative learning for deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 1832–1841, 2018.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Zhang, Y., Xiang, T., Hospedales, T. M., and Lu, H. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4320–4328, 2018.