TEACHING WITH COMMENTARIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective training of deep neural networks can be challenging, and there remain many open questions on how to best learn these models. Recently developed methods to improve neural network training examine *teaching*: providing learned information during the training process to improve downstream model performance. In this paper, we take steps towards extending the scope of teaching. We propose a flexible teaching framework using *commentaries*, meta-learned information helpful for training on a particular task or dataset. We present an efficient and scalable gradient-based method to learn commentaries, leveraging recent work on implicit differentiation. We explore diverse applications of commentaries, from learning weights for individual training examples, to parameterising label-dependent data augmentation policies, to representing attention masks that highlight salient image regions. In these settings, we find that commentaries can improve training speed and/or performance and also provide fundamental insights about the dataset and training process.

1 INTRODUCTION

Training, regularising, and understanding complex neural network models is challenging. There remain central open questions on making training faster and more data-efficient (Kornblith et al., 2019; Raghu et al., 2019a;b), ensuring better generalisation (Zhang et al., 2016) and improving transparency and robustness (Bau et al., 2017; Madry et al., 2017). A promising approach for addressing these questions is *learning-to-teach* (Zhu, 2015), in which learned auxiliary information about a task is provided to a neural network to inform the training process and help downstream objectives. Examples include providing auxiliary training targets (Liu et al., 2019; Navon et al., 2020; Pham et al., 2020) and reweighting training examples to emphasize important datapoints (Fan et al., 2020; Jiang et al., 2018; Ren et al., 2018; Shu et al., 2019).

Learning-to-teach approaches have achieved promising results in vision and language applications (Jiang et al., 2018; Ren et al., 2018; Shu et al., 2019) using a handful of specific modifications to the training process. In this paper, we take steps towards generalising these approaches, introducing a flexible and effective learning-to-teach framework using *commentaries*. Commentaries represent metalearned information helpful for learning about a particular task, dataset or even datapoint. We demonstrate that commentaries can effectively be used for applications ranging from speeding up training to gaining insights into the neural network model. Specifically, our contributions are:

1. We formalise the notion of commentaries to help provide a unifying framework for learning meta-information to improve network training and examine model learning.
2. We present a gradient-based method to learn commentaries by optimising a student network's validation loss, leveraging recent work in implicit differentiation to scale to larger models.
3. We use commentaries to define example-weighting curricula, a common method of teaching neural networks. We show that these learned commentaries hold interpretable insights, lead to speedups in training, and improve performance on few-shot learning tasks.
4. We investigate label-dependent commentaries to define a data augmentation policy, and obtain insights into the design of effective augmentation strategies and improved performance on benchmark tasks as compared to baselines.
5. We parameterise commentaries as attention masks to find important regions of images. Through qualitative and quantitative evaluation, we show these masks identify salient image regions and can be used to improve the robustness of neural networks to spurious background correlations.

2 TEACHING WITH COMMENTARIES

Definition: We define a *commentary* to be information helpful for training a model on a particular task, dataset or even datapoint. We propose that commentaries be stored with their datasets for use on future downstream tasks, or to provide insight about the dataset or task.

Let $t(x, y, i; \phi)$ denote a commentary that is a function of a data point, prediction target, and iteration of training, with parameters ϕ. The commentary may be represented in a tabular fashion for every combination of input arguments, or using a neural network that takes these arguments as inputs. This commentary influences the training of a student network with parameters θ, denoted $n(x; \theta)$.

2.1 LEARNING COMMENTARIES

Denoting the training set as $\mathcal{D}_T$ and the validation set as $\mathcal{D}_V$, we aim to find commentary parameters $\hat{\phi}$ that optimize the student network's validation loss $\mathcal{L}_V$. The validation loss depends on the commentary parameters implicitly through the student network parameters $\hat{\theta}$ obtained by optimising the training loss, $\mathcal{L}_T$. For a loss function $\mathcal{L}$ (e.g., cross entropy), we find:

$$\hat{\phi} = \texttt{optimize}(\phi)\left[\mathcal{L}_V(\phi)\right] = \texttt{optimize}(\phi)\left[\mathbb{E}_{x,y\sim\mathcal{D}_V}\left[\mathcal{L}\left(n(x;\hat{\theta}\left(\phi\right)), y\right)\right]\right], \tag{1}$$

where we define $\hat{\theta}(\phi)$, the trained parameters of the student network, as follows:

$$\hat{\theta}(\phi) = \texttt{optimize}(\theta)\left[\mathcal{L}_T(\theta, \phi)\right] = \texttt{optimize}(\theta)\left[\mathbb{E}_{x,y\sim\mathcal{D}_T}\left[\tilde{\mathcal{L}}\left(n\left(x;\theta\right), t\left(\cdot\,;\phi\right), y\right)\right]\right], \tag{2}$$

where $\tilde{L}$ is an adjusted loss function incorporating the commentary. We refer to equation 1 as the *outer optimisation*, and equation 2 as the *inner optimisation* (or inner loop). For both inner and outer optimisation, we use stochastic gradient descent.

The outer optimisation usually requires estimating $\frac{\partial\hat{\theta}}{\partial\phi}$, which is expensive for large models with many training iterations. We use one of two approaches depending on model size and training steps:

Backpropagating through training: When the inner optimisation has a small number of gradient steps, we obtain gradients for the commentary parameters by exactly backpropagating through the inner optimisation. For each step of the outer optimisation, we initialise a student with parameters θ_0, train for N steps of SGD to obtain $\hat{\theta}(\phi)$, compute the validation loss, and the gradients of this loss w.r.t. ϕ by backpropagating through N steps of training, and then update ϕ.

Implicit function theorem: When large models are trained with many iterations, computing $\frac{\partial\hat{\theta}}{\partial\phi}$ using backpropagation is too memory expensive. We therefore use the Implicit Function Theorem (IFT) and efficient inverse Hessian approximations, explored in Lorraine et al. (2020). Under fixed-point and regularity assumptions on student and commentary parameters $\left(\hat{\theta}(\phi), \phi\right)$, the IFT allows expressing the above derivative as:

$$\frac{\partial\hat{\theta}}{\partial\phi} = -\left[\frac{\partial^2\mathcal{L}_T}{\partial\theta\,\partial\theta^T}\right]^{-1} \times \frac{\partial^2\mathcal{L}_T}{\partial\theta\,\partial\phi^T}\Big|_{\hat{\theta}(\phi)}, \tag{3}$$

i.e., a product of an inverse Hessian and a matrix of mixed partial derivatives. Following Lorraine et al. (2020), we approximate this product using a truncated Neumann series and implicit vector-Jacobian products. In practice, a single term in the Neumann series is sufficient, which has similar time complexity to a single iteration of training. This approach scales to millions of commentary parameters, making gradient-based commentary learning practical on large models. Further methodological details are in Appendix A, and an illustrative example of commentaries is in Appendix B.

3 COMMENTARIES FOR EXAMPLE WEIGHTING CURRICULA

In this section, we explore commentaries that encode a separate weight for each training example at each training iteration. We specify these weights using a *commentary neural network (or teacher network)* $t(x, i; \phi) \rightarrow [0, 1]$ that produces a weight for every training example at every iteration of training of the student network. When training a student network, using the notation of §2.1, the commentary is incorporated in the training loss as: $\tilde{\mathcal{L}}_T = t(x, i; \phi) \cdot \mathcal{L}\big(n(x; \theta), y\big)$, where $\mathcal{L}(\cdot)$ is the original loss function for the task. The validation loss is unweighted.

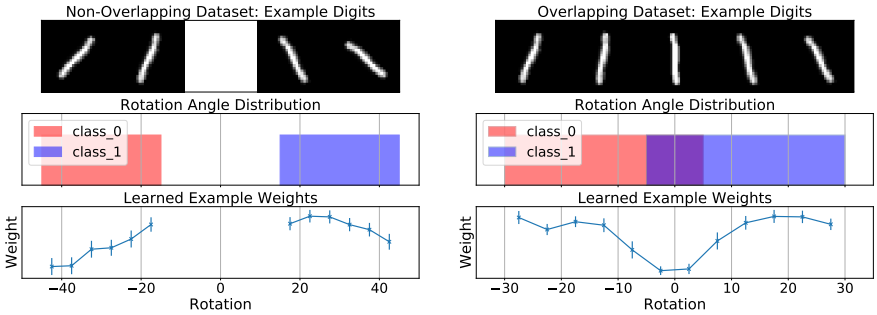

Figure 1: Learned per-example training weights on Rotated MNIST. We visualise rotated digits and rotation distributions from both the datasets. We plot the learned example weights at iteration 500 of student training as a function of rotation after discretising based on digit rotation. In the non-overlapping case (left), examples closer to the decision boundary are weighted most. When classes overlap (right), the more representative examples with greater rotation are weighted highly and examples in the overlap region are downweighted, which is sensible as examples in the overlap region are less indicative of the class label.

3.1 Synthetic Example: Rotated MNIST Digits

We first learn example weight curriculum commentaries on a synthetic MNIST binary classification problem. Each example in the dataset is a rotated MNIST digit '1', with variable rotation angle that defines the class. We generate two datasets: the **non-overlapping dataset** and the **overlapping dataset**. In the non-overlapping dataset, the rotation angle for each example from class 1 and class 0 is drawn from non-overlapping distributions Uniform$[15, 45]$ and Uniform$[-45, -15]$ respectively. In the overlapping dataset, the rotation angles are drawn from overlapping distributions Uniform$[-5, 30]$ and Uniform$[-30, 5]$ respectively (Figure 1).

We use two block CNNs as both the commentary neural network and student network. The commentary network takes as input the image and the iteration of student training, and outputs a weight for each example in the batch. We use 500 gradient steps for inner optimisation, and then compute the validation loss and commentary parameter gradients by backpropagating through the inner optimisation using the `higher` library (Grefenstette et al., 2019). Further details in Appendix C.1.

Results: Figure 1 visualises the two datasets and plots the learned example weights as a function of rotation at iteration 500 of the student training. When classes do not overlap (left), the example weights are highest for those examples near to the decision boundary (small rotation magnitude). When the classes do overlap (right), the more representative examples further from the boundary are upweighted and ambiguous examples in the overlap region are downweighted: a sensible result. We perform further analysis of the learned example weighting curriculum in Appendix C.1, demonstrating that the learned curricula in both cases are meaningful. Overall, these results demonstrate that the learned commentaries capture interesting and intuitive structure.

3.2 Commentaries for CIFAR10 and CIFAR100

We now learn example weighting curriculum commentaries on CIFAR10 and CIFAR100. The commentary network is again the two block CNN architecture, and when training the commentary network, the student network is also a two block CNN and is trained for 1500 inner loop gradient steps to obtain gradients on the validation set. For evaluation, the trained commentary network is used to produce example weights for (i) 2 block CNN (ii) ResNet-18 (iii) ResNet-34 student networks, all trained for 25000 steps, considering 3 random initialisations. Further details in Appendix C.2.

Example weighting commentaries improve learning speed. Figure 2 shows accuracy curves on the test sets of CIFAR10/100 for the two block CNN student with example weight curricula (orange line), a baseline (green line, no example weights) and an ablation (blue line, example weights without curriculum structure). On both datasets, the networks trained using the curriculum commentaries obtain better performance than the baseline and ablation over approximately 25000 steps of training (10 epochs), and have superior learning curves.

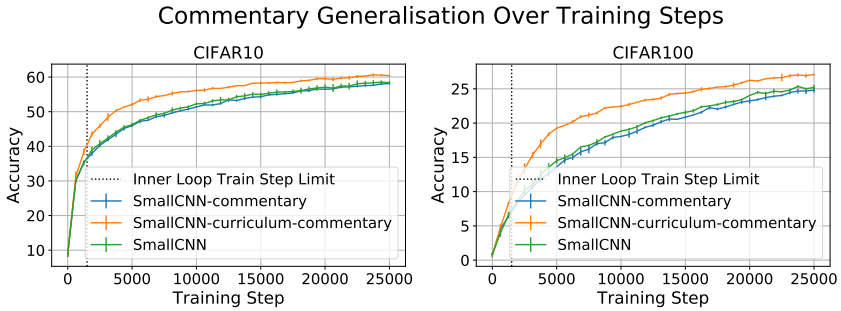

Figure 2: Example weight curricula can speed up training. Test-set accuracy curves on CIFAR10/100 when using curriculum commentaries, non-curriculum commentaries, and no commentaries, during student network training. The learned curriculum commentary network which generates per-iteration example weights results in learning speed improvements. This learning speed improvement holds when the student network is trained for many more steps than the number of inner loop update steps used during commentary network training (1500 steps). This demonstrates that the commentaries generalise to longer training times.

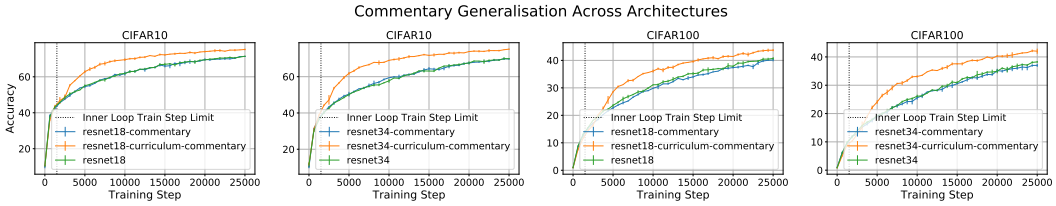

Figure 3: Example weight curricula generalise across network architectures. Using a learned curriculum commentary network trained with a simple 2 block CNN student, we apply these example weights to train two ResNet architectures. This gives improved test set accuracy curves for ResNet students also, indicating that the commentaries generalise across architectures.

Example weighting commentaries generalise to longer training times and across architectures. At training time, the commentary network was learned to produce example weights for the two block CNN student for 1500 inner loop update steps. We see in Figure 2 that the learned example weights lead to student network learning speedups well-beyond this point, suggesting generalisibility of the commentaries to longer training times. In addition, when the same commentary network is used to produce example weights for ResNet-18/34 students (Figure 3), we also observe a learning speedup, suggesting that the commentaries are generalisable across architectures.

3.3 Commentaries for Few-Shot Learning

Finally, we use example weight commentaries for few-shot learning. We start with the MAML algorithm (Finn et al., 2017), which learns a student parameter initialisation that allows fast adaptation on a new learning task using a support set of examples for that task. To incorporate example weighting in MAML, at training time, we jointly learn the MAML initialisation and a commentary network to provide per-example weights for each example in the support set, as a function of the inner loop step number. At test time, we follow the standard MAML procedure, and also incorporate the example weights when computing the support set loss and the resulting updates. Both networks use the 4-conv backbone structure from the original MAML paper. Details are in Appendix C.3.

We evaluate a standard MAML baseline and our commentary variant on standard few-shot learning benchmarks: (i) training/testing on MiniImageNet (MIN); and (ii) training on MIN and testing on CUB-200-2011 (CUB). Results are shown in Table 1, specifying the experimental setting (N-way K-shot), and the dataset used for training/testing. In all experiments, incorporating example weighting can improve on the MAML baseline, suggesting the utility of these commentaries in few-shot learning. Further experiments on other benchmark datasets (CIFAR-FS/SVHN) showing similar trends are in the appendix (Table C.1).

Mode: Train Data → Test Data	MAML Baseline	Example weighting + MAML
5-way 1-shot: MIN→MIN	43.64 ± 0.61	45.07 ± 0.61
5-way 1-shot: MIN→CUB	37.45 ± 0.55	38.00 ± 0.54
5-way 5-shot: MIN→MIN	61.85 ± 0.52	62.99 ± 0.52
5-way 5-shot: MIN→CUB	57.75 ± 0.54	59.06 ± 0.55

Table 1: Mean accuracy and 95% confidence interval across 1000 test-time tasks. On common few-shot learning benchmarks, using example weighting for the support set improves the MAML baseline's accuracy. Improvements are also observed when evaluating on out-of-distribution datasets.

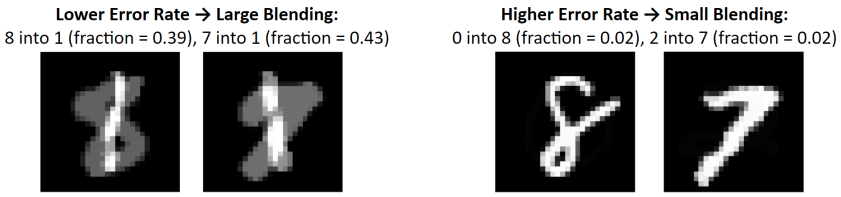

Figure 4: The learned blending augmentation is related to the class error rate. On digit 1s, a class with low average error rate, the learned augmentation blends in large amounts of other digits so as to maximise the learning signal. On digits with higher average error rates, such as 7s and 8s, the level of blending is very low so as to focus on classifying the class alone (rather than making the task more complex by adding another digit).

4 Learning to Blend Training Examples

We next investigate learning commentaries that define *label-dependent* data augmentation policies. We first consider an augmentation scheme where pairs of images are blended together with a proportion dependent on the classes of the two examples. At each training iteration, we:

- Sample two examples and their labels, (x_1, y_1) and (x_2, y_2) from the training set.
- Obtain the blending proportion $\lambda = t(y_1, y_2; \phi)$, and form a new image $x_m = \lambda x_1 + (1 - \lambda)x_2$, and class y_m equivalently.
- Use this blended example-label pair (x_m, y_m) when computing the training loss.

To compute the validation loss, use only unblended examples from the validation set.

For classification problems with N classes, the teacher $t(y_1, y_2; \phi)$ outputs an $N \times N$ matrix. This augmentation scheme is inspired by *mixup* (Zhang et al., 2018). However, we blend with a deterministic proportion, depending on pairs of labels, rather than drawing a blending factor from a Beta distribution. In doing so, we more finely control the augmentation policy.

Augmentation Commentaries on MNIST: We learn an augmentation commentary model t on MNIST by backpropagating through the inner optimisation of a 2-block CNN student network. In the learned augmentation, the error rate on class i is correlated (Pearson correlation$= -0.54$) with the degree of blending of other digits into an example of class i: lower error on class i implies that other digits are blended more heavily into it. On MNIST, this means that the class that has on average the lowest error rate (class 1) has other digits blended into it significantly (Figure 4 left); classes that have on average higher error rate (e.g., class 7, class 8) rarely have other digits blended in (Figure 4 right). Further details in Appendix D.1.

4.1 Augmentation Commentaries for CIFAR10 and CIFAR100

We next learn and evaluate augmentation commentaries on CIFAR10 and CIFAR100. We evaluate the effect of these augmentation commentaries on improving generalisation performance; since this represents a longer training process, we use the IFT method (described in § 2.1) to ensure computational tractability. We learn the commentaries jointly with a ResNet18 student network (as in Lorraine et al. (2020)). Once the commentary is learned, we fix it and train three new students with different random initialisations to evaluate the commentary's efficacy.

	CIFAR10		CIFAR100	
	Test Accuracy	Test Loss	Test Accuracy	Test Loss
No commentary	93.84 ± 0.22	0.239 ± 0.005	74.79 ± 0.39	1.05 ± 0.03
Random commentary	93.84 ± 0.30	0.385 ± 0.007	73.39 ± 0.58	1.14 ± 0.01
mixup (Zhang et al., 2018)	94.42 ± 0.12	0.313 ± 0.010	75.89 ± 0.55	1.01 ± 0.02
Learned commentary	94.12 ± 0.19	0.225 ± 0.007	76.25 ± 0.11	0.97 ± 0.01

Table 2: Learned augmentation commentaries result in competitive or improved model accuracy and loss on CIFAR10/100. Using the learned, label-dependent augmentation commentary during training results in models that are competitive with *mixup* and superior to other baselines. We show mean/standard deviation over three different initialisations.

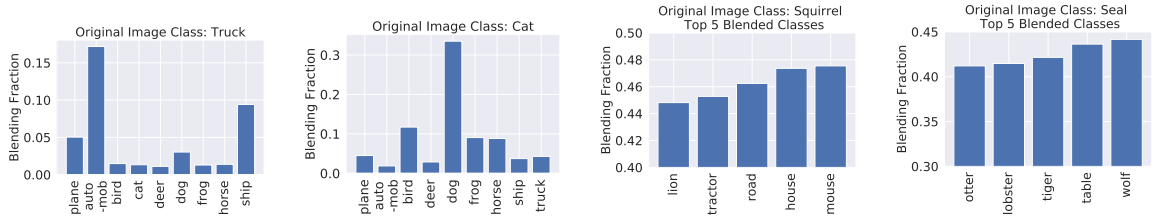

Figure 5: Blending proportions for selected CIFAR10 and CIFAR100 classes reveal interesting insights. Visualising the learned blending proportions on two CIFAR10 classes (left), we see that both classes are most blended with others that are visually similar (truck-automobile, and cat-dog), which may help the network differentiate between them. On CIFAR100 (right), considering the top 5 blended classes in two cases, we observe again the presence of visually similar classes that may be confused (seal-otter, and squirrel-mouse), but also visually unrelated classes. These may provide extra learning signal with each example.

Results: Table 2 shows test accuracy for different augmentation policies on CIFAR10 and 100. We compare the learned commentary to using only standard data augmentations for CIFAR10/100 (No commentary) and a random initialisation for the commentary matrix (Random commentary). We also compare to *mixup* (Zhang et al., 2018). We observe that the learned commentary is competitive with *mixup* and improves on other baselines across both tasks. In the appendix, we compare to an ablation that destroys the structure of the learned commentary grid by shuffling it, and find that the unshuffled commentary does better (Table D.1).

Further Analysis: In Figure 5, we visualise: (i) for two CIFAR10 classes, the blending fractions (defined as $(1 - \lambda)$) associated with the other classes (left); and (ii) for two CIFAR100 classes, the blending fractions associated with the five most highly blended classes. For CIFAR10, we see that other classes that are visually similar and therefore sources of misclassification are blended more significantly. On CIFAR100, we see that within the top 5 blended classes, there are classes that are visually similar, but there are also blended classes that have far less visual similarity, which could be blended in to obtain more learning signal per-example. Further analysis in Appendix D.2.

5 Attention Mask Commentaries for Insights and Robustness

We now turn to a challenging task: studying whether commentaries can learn to identify key (robust) features in the data. We formalise this problem as one of learning commentaries which act as attention masks. We perform a qualitative empirical study, learning commentary attention masks on a variety of image datasets: an MNIST variant, CIFAR10/100, medical chest X-rays, and Caltech-UCSD Birds (CUB)-200-2011, where we find that the learned commentaries identify salient image regions. We perform quantitative experiments, which illustrate the effectiveness of attention mask commentaries over baselines and in ensuring robustness to spurious correlations.

Formally, we learn a commentary network $t(x; \phi) \rightarrow [i, j]$ to output the centre of a 2D Gaussian that is then computed and used (with predefined standard deviation depending on the input image size, see Appendix E) as a pixelwise mask for the input image before feeding it through a student network. We denote the mask based on $t(x; \phi)$ as $m(x, t)$. Our goal is to learn masks that highlight

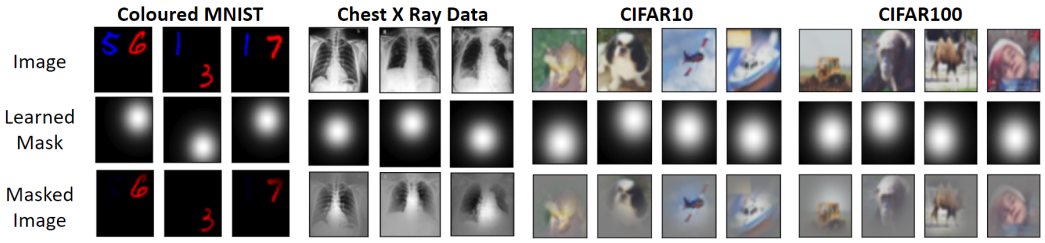

Figure 6: Learned attention masks highlight salient image regions for classification. We learn a commentary network to produce image attention masks on several image datasets, and the masks are qualitatively sensible in all cases. On Coloured MNIST, where the image label is determined by the red digit, the masks focus on the red digit. On a dataset of chest X-rays, the masks focus on the chest cavity, which is the appropriate reason for detecting the condition in question (cardiomegaly). On CIFAR10/100, the masks are focused on important regions, such as the faces of animals, the hump of the camel, and the bodies of vehicles.

the most important regions of the image for training and testing,so the masks are used both at train time and test time. We therefore have that $\tilde{\mathcal{L}} = \mathcal{L} = \text{x-ent}\left(n\left(x \odot m\left(x, t\right); \theta\right), y\right)$. The commentary network is a UNet (Ronneberger et al., 2015) with an output layer from KeypointNet (Suwajanakorn et al., 2018), learned using the IFT method, and the student network is a ResNet18.

5.1 Qualitative and Quantitative Analysis On Image Datasets

Masks for Coloured MNIST: We learn masks on a dataset where each image has two MNIST digits, coloured red and blue, with the red digit determining the label of the image. As seen in Figure 6 left, the commentary selectively attends to the red digit and not the blue digit.

Masks for Chest X-rays: Using a dataset of chest X rays (Irvin et al., 2019), we train a student network to detect cardiomegaly, a condition where an individual's heart is enlarged. Learned masks are centered on the chest cavity, around the location of the heart (Figure 6), which is a clinically relevant region. These masks could be used in medical imaging problems to prevent models relying on spurious background features (Badgeley et al., 2019; Winkler et al., 2019).

Masks for CIFAR10/100: The learned masks on CIFAR10/100 (Figure 6) attend to important image regions that define the class, such as the faces of animals/the baby, wheels/body of vehicles, and the humps of the camel. In the appendix (Table E.1) we show quantitatively that the learned masks are superior to other masking baselines, and also provide further qualitative examples.

5.2 Mask Commentaries for Robustness to Background Correlations

Using the task introduced in Koh et al. (2020), we demonstrate that mask commentaries can provide robustness to spurious background correlations. We take the CUB-200-2011 dataset (Welinder et al., 2010) and the associated fine-grained classification problem to classify the image as one of 200 bird species. Using the provided segmentation masks with the dataset, the background of each image is replaced with a background from the Places dataset (Zhou et al., 2017). For the training and validation sets, there is a specific mapping between the CUB class and the Places class for the background, but for the testing set, this mapping is permuted, so that the background features are now spuriously correlated with the actual image class.

We first learn an attention mask commentary network with the IFT method, and for evaluation, use this learned network when training a ResNet18 student (pretrained on ImageNet, as in Koh et al. (2020)). We assess student performance on the validation and test sets (with the same and different background mappings as the training set, respectively), considering three random seeds.

Results: Learned masks are shown in Figure 7; the masks are mostly focused on the bird in the image (though some also contain parts of the background). In a quantitative evaluation (Table 3), we see that using the masks helps model performance significantly on the test set as compared to a baseline that does not use masks. This suggests that the masks are indeed helping networks to rely on more robust features in the images. Note that the accuracy drop on the validation set is expected as we are limiting the region of the image that the model gets as input.

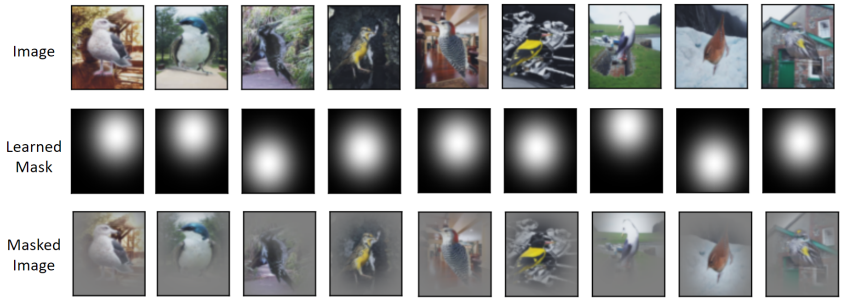

Figure 7: Learned masks on CUB dataset are primarily focused on the bird, rather than the background.

	Validation Set Accuracy		Test Set Accuracy (Distribution Shift)	
	Top 1	Top 5	Top 1	Top 5
Baseline (No masks)	78.82 ± 1.19	93.95 ± 0.28	25.81 ± 0.60	56.98 ± 0.69
With Masking Network	74.73 ± 0.76	92.45 ± 0.23	30.46 ± 0.27	61.57 ± 0.70

Table 3: Learned attention masks offer improved robustness to spurious background correlation. We train a commentary network to produce attention masks on a version of the CUB-200-2011 dataset that contains spurious background correlations, with the correlations present in the training/validation set *not* present in the test set (the test set has a distribution shift). On both top 1 and top 5 accuracy, using the masks results in improved model performance on the shifted test set.

6 Related Work

Learning to Teach: Using teaching in neural network training was explored at a high level in Zhu (2015), and the benefit of curriculum structure in learning was discussed in Bengio et al. (2009). Several recent works have explored teaching methods including: selecting what training examples to use (Fan et al., 2018; Liu et al., 2017); adjusting loss functions (Wu et al., 2018); metalearning auxiliary tasks/labels for use in training (Liu et al., 2019; Navon et al., 2020; Pham et al., 2020); and example weighting (Fan et al., 2020; Jiang et al., 2018; Ren et al., 2018; Shu et al., 2019). These works mainly focused on developing a single given method of teaching to improve downstream model performance. In contrast, in our work, we propose *commentaries* as more general framework for teaching. Example weighting is readily represented with commentaries, and we explore two other, distinct modes of teaching. We also examine learned commentaries to obtain insights that could more generally inform network training.

Learning with Hypergradients: We use *hypergradients* — derivatives of a model's validation loss with respect to training hyperparameters — to learn commentaries. Prior work on computing hypergradients includes memory-efficient exact computation in Maclaurin et al. (2015), using stored auxiliary information, and efficient approximate computation in Lorraine et al. (2020), using the implicit function theorem (IFT) and efficient approximate Hessian matrix inversion (which we adopt). Prior uses of the IFT in metalearning include Navon et al. (2020); Rajeswaran et al. (2019).

7 Conclusion

In this paper, we considered a general framing for teaching neural networks using *commentaries*, defined as meta-information learned from the dataset/task. We described a gradient-based method to learn commentaries and three distinct methods of applying commentaries to assist learning: example weight curricula, data augmentation, and attention masks. Empirically, we show that the commentaries can provide insights and result in improved learning speed and/or performance on a variety of datasets. Teaching with commentaries is a proof-of-concept idea, and we hope that this approach will inspire related ways of automatically re-using training insights across tasks and datasets.

REFERENCES

Marcus A Badgeley, John R Zech, Luke Oakden-Rayner, Benjamin S Glicksberg, Manway Liu, William Gale, Michael V McConnell, Bethany Percha, Thomas M Snyder, and Joel T Dudley. Deep learning predicts hip fracture using confounding patient and healthcare variables. *NPJ digital medicine*, 2(1):1–10, 2019.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Yang Fan, Fei Tian, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. Learning to teach. *arXiv preprint arXiv:1805.03643*, 2018.

Yang Fan, Yingce Xia, Lijun Wu, Shufang Xie, Weiqing Liu, Jiang Bian, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. Learning to teach with deep interactions. *arXiv preprint arXiv:2007.04649*, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

Edward Grefenstette, Brandon Amos, Denis Yarats, Phu Mon Htut, Artem Molchanov, Franziska Meier, Douwe Kiela, Kyunghyun Cho, and Soumith Chintala. Generalized inner loop meta-learning. *arXiv preprint arXiv:1910.01727*, 2019.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 590–597, 2019.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pp. 2304–2313, 2018.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. *arXiv preprint arXiv:2007.04612*, 2020.

Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2661–2671, 2019.

Hae Beom Lee, Hayeon Lee, Donghyun Na, Saehoon Kim, Minseop Park, Eunho Yang, and Sung Ju Hwang. Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. In *International Conference on Learning Representations*, 2019.

Shikun Liu, Andrew Davison, and Edward Johns. Self-supervised generalisation with meta auxiliary learning. In *Advances in Neural Information Processing Systems*, pp. 1679–1689, 2019.

Weiyang Liu, Bo Dai, Ahmad Humayun, Charlene Tay, Chen Yu, Linda B Smith, James M Rehg, and Le Song. Iterative machine teaching. In *International Conference on Machine Learning*, pp. 2149–2158, 2017.

Liangqu Long. Maml-pytorch implementation. `https://github.com/dragen1860/MAML-Pytorch`, 2018.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pp. 1540–1552, 2020.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International Conference on Machine Learning*, pp. 2113–2122, 2015.

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.

Aviv Navon, Idan Achituve, Haggai Maron, Gal Chechik, and Ethan Fetaya. Auxiliary learning by implicit differentiation. 2020.

Hieu Pham, Qizhe Xie, Zihang Dai, and Quoc V Le. Meta pseudo labels. *arXiv preprint arXiv:2003.10580*, 2020.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *International Conference on Learning Representations*, 2019a.

Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. In *Advances in neural information processing systems*, pp. 3347–3357, 2019b.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, pp. 113–124, 2019.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*, 2018.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, pp. 1919–1930, 2019.

Supasorn Suwajanakorn, Noah Snavely, Jonathan J Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Advances in neural information processing systems*, pp. 2059–2070, 2018.

Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.

Julia K Winkler, Christine Fink, Ferdinand Toberer, Alexander Enk, Teresa Deinlein, Rainer Hofmann-Wellenhof, Luc Thomas, Aimilios Lallas, Andreas Blum, Wilhelm Stolz, et al. Association between surgical skin markings in dermoscopic images and diagnostic performance of a deep learning convolutional neural network for melanoma recognition. *JAMA dermatology*, 155 (10):1135–1141, 2019.

Lijun Wu, Fei Tian, Yingce Xia, Yang Fan, Tao Qin, Lai Jian-Huang, and Tie-Yan Liu. Learning to teach with dynamic loss functions. In *Advances in Neural Information Processing Systems*, pp. 6466–6477, 2018.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

Xiaojin Zhu. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. 2015.