Machine Teaching with Generative Models for Human Learning

Michael Doron [1] **Hussein Mozannar** [2] **David Sontag** [2] **Juan C. Caicedo** [1]

Abstract

Experimental scientists face an increasingly difficult challenge: while technological advances allow for the collection of larger and higher quality datasets, computational methods to better understand and make new discoveries in the data lag behind. Existing explainable AI and interpretability methods for machine learning focus on better understanding model decisions, rather than understanding the data itself. In this work, we tackle a specific task that can aid experimental scientists in the era of big data: given a large dataset of annotated samples divided into different classes, how can we best teach human researchers what is the difference between the classes? To accomplish this, we develop a new framework combining machine teaching and generative models that generates a small set of synthetic teaching examples for each class. This set will aim to contain all the information necessary to distinguish between the classes. To validate our framework, we perform a human study in which human subjects learn how to classify various datasets using a small teaching set generated by our framework as well as several subset selection algorithms. We show that while generated samples succeed in teaching humans better than chance, subset selection methods (such as k-centers or forgettable events) succeed better in this task, suggesting that real samples might be better suited than realistic generative samples. We suggest several ideas for improving human teaching using machine learning.

1. Introduction

Scientific research often necessitates the ability to analyze labeled data and identify the differences that exist in it. However, this task is often challenging due to the high dimensional nature of the data and the large number of examples. For example, consider cell microscopy images from sick and healthy individuals where each class is known to be sampled from a separate population, but where the exact nature of the differences between the healthy and sick cells is unknown. Encouragingly, machine learning (ML) classifiers often succeed in separating these samples into their respective groups, indicating that even though scientists have not yet found the distinction between them, there exists a set of features that separates the classes.

This motivates the possibility of developing a teaching setup where machine learning methods can find and highlight features that will teach humans to better separate the groups. Our teaching setup consists of showing human participants a small compressed dataset that contains the necessary information to learn the classification problem. Using a combination of machine teaching and generative models, we optimize a synthetic realistic teaching set that succeeds in teaching various human-proxy student models how to classify different datasets better than subset-selection methods.

We further compare the performance of real human participants who were trained on these realistic synthetic teaching sets to humans who were trained on real subset selection sets. We focus on biological datasets, where the problem of making sense of data which is simultaneously annotated and unexplained is prevalent. Our preliminary results suggest that generated realistic images are better than real images in the task of teaching ML models how to classify, but that subset selection methods (such as k-centers (Sener & Savarese, 2017) or forgettable events (Toneva et al., 2018)) might be just as good, or better, than generated realistic images in teaching humans how to classify.

Our main contributions are:

- We suggest a method for generating a realistic synthetic teaching set that is able to teach both ML classifiers and human learners.
- We perform an empirical comparison between different teaching sets, both synthetic and real, and study how well they teach human participants.

[1] Broad Institute of MIT and Harvard, Cambridge, MA, USA [2] Massachusetts Institute of Technology, Cambridge, MA, USA. Correspondence to: Michael Doron <mdoron@broadinstitute.org>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

2. Related work

Our work is related to and is informed by several fields. *Subset selection* deals with finding a subset of samples $D_s \subset D$ s.t. a student learning from D_s will perform as well as a student learning from D. While most works deal with specific student models (Singla et al., 2014; Chen et al., 2018; Aodha et al., 2018; Pinsler et al., 2020), recent ones attempt to teach deep learning models (Fan et al., 2018; Sener & Savarese, 2018; Nguyen et al., 2020; Coleman et al., 2019). Although these selected samples are better at teaching compared to randomly selected samples, our hypothesis was that D may not always contain the perfect teaching examples, and that a smaller generated dataset $\hat{D}$ might be better suited than D_s. Sharing our approach of generating teaching samples is the relatively new field of *dataset distillation*, which deals with generating a new set of samples that will teach the student model best (Wang et al., 2018; Lorraine et al., 2019; Raghu et al., 2020). These papers, while similar to us in using machine teaching in a bi-level optimization fashion, directly optimize the pixels of the generated images. We, on the other hand, suggest viewing the teaching process as searching the latent space of a generative model. The use of a generative model has two main potential advantages: 1. Generative teaching samples may contain useful teaching properties such as compression of information from several real instances or highlighting of information that is crucial for the classification process. 2. Teaching samples that are generated using a pretrained generative model are realistic, hopefully making them easier to learn from for the human learners.

A recent paper, *Generative Teaching Networks* (GTN) (Such et al., 2020), attempted a similar approach of machine teaching, or bi-level optimization, combined with generative models, in order to generate a set of synthetic images that would teach a student model better than other teaching sets. While similar to our attempted approach, there are several major differences between the studies: 1. GTN trains the generator from scratch during the teaching phase, causing the images to be unrealistic, while we first train the generator to produce realistic images and only then begin the teaching optimization process. 2. The main application of GTN is neural architecture search (NAS), while we focus on teaching humans. 3. Because of its goal, GTN uses ML student models that have different inductive biases and learning strategies than human learners, while we use human-proxy models that were designed to find the best teaching examples for human learners.

In addition to these works, it is worth pointing out several recent works that might aid us in the discussion of the next steps: (Singla et al., 2019) generates an "exaggeration" of semantic features of different instances, thus highlighting what makes them more or less fitting into different classes. Similarly, (Schut et al., 2021) generates counterfactual images that show the minimal *realistic* change necessary to convert an image from one class to another.

3. Problem Formulation

Assume a labeled data distribution $\mathcal{X} \times \mathcal{Y}$, from which we sample a dataset $D = \{x_i, y_i\}_{i=1}^n$, s.t. $x_i \in \mathcal{X}, y_i \in \mathcal{Y}$. Further assume a *student* $S(\mathcal{X} \times \mathcal{Y}) : \mathcal{X} \to \mathcal{Y}$ that receives a dataset D and produces a trained predictor $\mathcal{X} \to \mathcal{Y}$. We wish to produce a teaching set $D_T = \{x_i, y_i\}_{i=1}^b, b \ll n$, such that $S(D_T)$ will reach low prediction error when tested on D. We assume that the student has never seen any instances from D before training on instances from D_T. Note that D_T is not restricted to be a subset of D, so that we move away from the usual paradigm of *subset selection* in the standard teaching literature to that of *subset generation*. In the experiments shown in this paper, $Y \subset \mathbb{N}$ and $\mathcal{X} \subset \mathbb{R}^{p \times p}$ (i.e., $\mathcal{Y}$ is discrete and $\mathcal{X}$ contains images), but the framework is relevant to other data types as well. We measure the performance of the predictor S via the expectation of the loss $l(y, \hat{y}) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$:

$$L_D(S(D_T)) = \frac{1}{n} \sum_{i \in [n]} l\left(S(D_T)(x_i), y_i\right). \tag{1}$$

4. Algorithm

Next, we will go over our proposed solution to the problem: the Machine Teaching with Generative Models framework. In this framework, the teaching set D_T is generated by a *teacher* $T : \{\mathcal{X} \times \mathcal{Y}\}^n \to \{\mathcal{X} \times \mathcal{Y}\}^b$ that takes in D, a dataset of real samples, and generates D_T, a smaller dataset that will minimize the error of $S(D_T)$ over D. We assume the teacher T has full access to the dataset D. In addition, we assume T has no access to the student S, but that it has full access to the performance of S, and in the case where S is a differentiable function, to the gradients of S during training. During training, T will solve the following optimization problem:

$$T(D) := \arg \min_{D_T \in \{\mathcal{X} \times \mathcal{Y}\}^b} L_D\left(S(D_T)\right). \tag{2}$$

This optimization process is a form of machine teaching that uses bi-level optimization.

4.1. Machine teaching.

In the beginning of each teacher training iteration, or step 1 in Figure 1, T produces an annotated set of samples D_T and passes it to student S for training. Inside the student loop at step 2, S trains a predictor $\mathcal{X} \to \mathcal{Y}$ to minimize $L_{D_T}\left(S(D_T)\right)$ over several student epochs. Finally, in step

3, the predictor $S(D_T)$ is tested on the real dataset D, producing a loss term $L_D\left(S(D_T)\right)$. This loss term backpropagates back to T through the optimization process in step 2, allowing T to change its parameters and produce better samples in the next teacher loop iteration. We found that T produces better teaching examples when S is reset at the beginning of each teacher loop; when S can continue its training (as done in (Lorraine et al., 2019)), S retains information from earlier iterations of D_T, preventing T from creating an isolated set D_T that will contain all information necessary to teach S how to predict D.

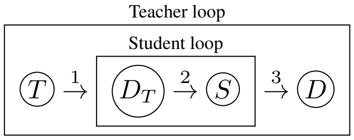

Figure 1. **Machine Teaching framework.** The teacher generates teaching set D_T using the generator by feeding it latent code vectors D_z (step 1). The student learns how to classify D_T over many iterations (step 2). Finally, the student is being tested on the real dataset D, and its performance is backpropagated though the learning process in step 2 to the teacher, allowing the teacher to optimize D_z and generate better teaching examples in future iterations of the teacher loop (step 3).

4.2. Teacher: Generative models

To produce a set of teaching examples D_T that would best teach S how to solve D, we propose that T should have access to a *pretrained* generative model $G : \mathcal{Z} \to \mathcal{X}$ that was trained on D to produce realistic samples. Over training, T will learn a set of latent space vectors $D_z = \{z_i, y_i\}_{i=1}^{b}$, s.t. $z_i \in \mathcal{Z}, y_i \in \mathcal{Y}$. These latent codes will be fed to G, producing D_T. Note that in our solution, the label of each sample in D_z is known in advance, allowing the teacher to focus on generating the best teaching examples for each class. Note also that unlike (Such et al., 2020), we use a pretrained generative model. This has the benefit of requiring the teacher to only learn how to sample from an existing latent space, instead of learning the dual task of generating both realistic samples and teachable samples.

4.3. Optimizing $L\left(S(D_T)\right)$

To perform machine teaching and generate an optimal teaching set, we need to optimize the samples generated by the teacher based on the performance of the student on the real dataset. This bi-level optimization process (see Figure 1) is demanding, as it requires keeping the gradients of the students throughout the training phase. We followed a recent approach (Lorraine et al., 2019) that suggests estimating the inner loop gradients (step 2 in Figure 1) using implicit function theorem, thus allowing us to train the student for long (500+ student epochs) training phases without keeping the gradients in memory.

4.4. Student: Human-proxies

Because humans require orders of magnitude more time for each training epoch (i.e., viewing training samples, testing their classification performance on testing samples, receiving feedback and receiving the next batch of training samples), we use a human-proxy ML student. The teacher learns how to generate samples that teach the human-proxy student how to predict the real dataset, with the ultimate goal of giving this dataset to a human student who will perform better than it would have given a different teaching dataset.

We divide the identity of the student model in two: The feature extractor and the classifier head. In both human-proxy student models in this study we used an extractor taken from a residual convolutional neural network (He et al., 2016) pretrained on ImageNet (Deng et al., 2009). While the metric space learned by a residual network on ImageNet is not necessarily the same as the metric space used by humans to identify the distance between images, we assume that its training on thousands of classes gives it a good approximation of human metric learning.

Unlike the feature extractor that is frozen and shared between the models, the student models vary in their classifier head that receives the features: Either a linear fully connected layer (*Linear Classifier*, or *LC*) or a nearest neighbor classifier (*Nearest neighbor*, or *NN*). In all student models, the weights of the feature extractor were frozen, and only the classifier head is updated during the student training loop. The following subsections present the main relevant details of each student model evaluated in our work, and how they relate to human learning.

4.4.1. Linear Classifier (LC)

For this student, we used a fully connected layer after the feature extractor, with the number of outputs as the number of classes. The loss function used for this classifier is

$$l_{LC}(x,y) = -\sum_{i}^{c} \mathbb{1}_{y=c} log\left(\frac{exp(\theta_i(x))}{\sum_j exp(\theta_j(x))}\right), \tag{3}$$

where $\theta_i(x)$ is the logit output of the residual network feature extractor for class i. The *LC* student was meant to mimic the decision bound theory of human classification learning (Trabasso, 1975), in which humans learn by projecting the sample onto a feature space and learning a decision boundary in that space.

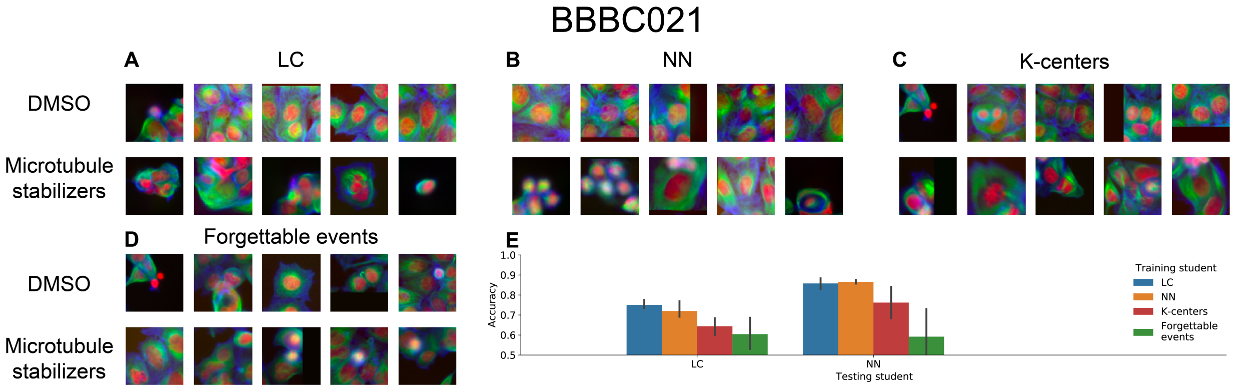

Figure 2. **Generative teaching examples perform better than subset selection examples when teaching synthetic models.** A-B. Generative teaching sets learned using the *LC* and *NN* human-proxy students. C-D. Teaching sets selected from the real BBBC021 dataset using the K-centers algorithm and forgettable events algorithm. All teaching sets are divided into the DMSO class (upper row) and Microtubule stabilizers class (lower row). E. The performance of new *LC* and *NN* students trained using teaching sets created similarly to those in A-D.

4.4.2. Nearest Neighbor (NN)

For this student, we use the teaching examples D_T as prototypes in a way similar to a prototypical network (Snell et al., 2017), classifying each new sample extracted from D based on its closest prototype in the pretrained metric space. We use a similarity function based on the pretrained metric space $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$. The decision of the NN student can be formulated as

$$l_{NN}(x) = -\sum_{i}^{c} \mathbb{1}_{y=c} log\left(\frac{exp(K(x,x_i))}{\sum_j exp(K(x,x_j))}\right), \quad (4)$$

Where x_i is the prototype for class i.

For this student, we forgo the bi-level optimization framework, and inject the teaching examples D_T directly as prototypes; the only learning is done by the teacher who chooses the prototypes based on the loss of the student over D. The *NN* student was meant to mimic the prototype theory of human classification learning (Rosch, 1975), in which humans learn prototypes for each category, and compare new samples to the nearest prototype for classification.

5. Experiments

To test the machine teaching with generative model framework, we trained the teacher on two datasets from biological domains (BBBC021 and Retina, described in Section 6). For each dataset, we compared two different student models (*LC* and *NN*), as well as two baselines: K-centers (Sener & Savarese, 2017) and Forgettable events (Toneva et al., 2018). Each model was used to create a different teaching set that was later used to teach either a new *NN* or a *LC* student on the real dataset. In the Retina dataset we also compared the ability of the teaching sets to teach human students. The images from the teaching set were removed from the final testing set. The baselines we compare against are as follows:

- **K-centers (Sener & Savarese, 2017).** This method selects k prototypes from D that cover D as much as possible in the metric space.
- **Forgettable events (Toneva et al., 2018).** This method trains an artificial neural network to classify D, and sorts the samples by how often the network switched from classifying them one way or another. The samples that “forget” their label the most are hypothesized to have the most amount of information about the classes.

5.1. Synthetic experiments

In this experiment, we tested the ability of the teaching sets to teach the human-proxy ML models described in Section 4. For each of the two datasets, and for each of the human-proxy and subset selection models, we trained a separate teaching set. We then used these teaching sets to train the *LC* and *NN* human-proxy ML students, with a new randomly initialized model for each of the teaching sets. Each teaching set and testing human-proxy student experiment was repeated five times. The *LC* student model was trained for 50 epochs, while the *NN* model was tested based on the prototypes given by the teaching set. The models were tested on their accuracy in correctly classifying the entire dataset D.

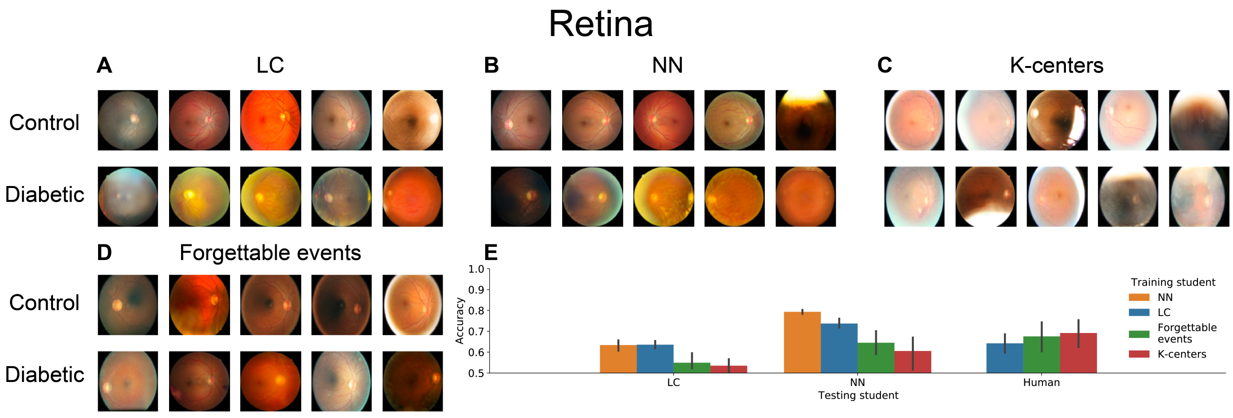

Figure 3. **While generated teaching sets outperform selected teaching sets in teaching synthetic models, human participants learned better using real images compared to generated ones.** A-D. As in Figure 2, with the Retinopathy Detection dataset. All teaching sets are divided into the Control class (upper row) and Diabetic class (lower row). E. The performance of new *LC*, *NN* and *Human* students trained using teaching sets created similarly to those in A-D.

5.2. Human experiments

In this experiment, we tested the ability of the generated teaching sets to teach real human learners compared to selected teaching sets. We asked a small sample of 12 people, 4 per condition, to assist in a pilot experiment. The experiment worked as follows: First, the participants saw all ten images of the teaching set, including labels. After confirming that they understood the teaching set, the participants moved on to the next stage, where they received each image of the teaching set separately without labels, and were asked to classify them. After classifying all images from the teaching set, the participants moved on to the final testing stage, where they saw 200 randomly selected unlabeled images from the real dataset, and were asked to classify them by clicking on the name of the class they belong to. In addition, at this stage, the participants could click a button at will to look at the teaching set, and then return to the testing set to continue the classification.

6. Results

6.1. Dataset #1 – BBBC021

We tested whether our framework can generate realistic teaching sets that can train new student learners on cellular microscopy data. We used the BBBC021 dataset (Caie et al., 2010), a dataset of human breast cancer cellular images. The cells were treated with 113 compounds at 8 different doses and sorted into 12 mechanisms of action (MOA). In this particular dataset, the training set was composed of 1,400 images equally distributed between the control (i.e., DMSO) and the Microtubule stabilizer MOA images. For the teacher, we first fine-tuned a pretrained StyleGAN2 generative model (Karras et al., 2020) that was originally trained on the FFHQ (Karras et al., 2019) dataset for 550,000 epochs. This model was fine-tuned on the 1,400 BBBC021 images for 90,000 epochs to generate realistic images of BBBC021 cells. Then, during the teacher training, we optimized ten latent vectors – five per class – to generate the prototypes in D_T.

We trained two generative student models (*LC* and *NN*) and used the Machine Teaching framework described in Section 4 to generate teaching sets for each student model (See Figure 2A-B). Each teacher was trained for 1,000 teacher epochs, where inside each teacher epoch the student trained for 50 student epochs (see Figure 1). The students were reset to the same initialized state at the beginning of each teacher epoch.

For the evaluation, we trained new *LC* and *NN* models with new random parameter initializations on the teaching sets produced in Figure 2A-D. The results can be seen in Figure 2E. The results show that both the *LC* and the *NN* students performed best when trained with generative teaching sets, followed by the subset selection methods. Each experiment was repeated five times (i.e., each teaching set was generated five times with different random seeds, and each testing student was randomly initialized to train on the datasets).

6.2. Dataset #2 – Retina

Similar to the BBBC021 dataset, we tested how well our framework performs with a more complex dataset, the Diabetic Retinopathy Detection dataset (Graham, 2015) that consists of images of retinas with different stages of diabetic retinopathy, annotated by experts. We subsampled the retina dataset to get 1,600 images divided equally from the control (no diabetic retinophaty) and the most severe (proliferative diabetic retinopathy) cases, and fine-tuned a pretrained

StyleGan2 generative model that was trained for 550,000 epochs on the FFHQ dataset. This model was fine-tuned on the subsampled retina dataset for 10,000 epochs to produce realistic images. The teacher and student framework was the same as in Section 6.1, and the generated teaching sets, as well as the teaching sets selected by the subset selection baselines, can be seen in Figure 3. Based on the better performance of the *LC* student in the BBBC021 dataset, we tested the human students on the teaching set produced by the *LC* student.

As can be seen in Figure 3, the human participants succeeded in classifying the images from the testing set better than chance for both the subset selection methods and the generated teaching sets. However, while the *LC* and *NN* student performed best when trained on generated teaching sets, the human students performed best on the real samples chosen by subset selection methods, in a reverse order to the performance of the human-proxy student models.

7. Discussion

In this work, we tested the possibility of using machine teaching in combination with generative models to generate a small set of images that will contain all the information necessary to teach humans how to predict the class of samples in a real dataset. We found that while this method indeed succeeds in producing realistic images that teach a *human-proxy* model well, and while these realistic synthetic teaching sets succeed in teaching real humans better than chance, we don't yet succeed in producing a teaching set that will teach human students better than simpler methods for choosing prototypes from the real dataset using subset selection.

These preliminary results raise interesting questions that can be addressed in future work: How come realistic synthetic teaching sets, that succeed in teaching humans better than chance, still teach less well than subset selection methods? First, it may be that the human proxy students do not mimic the human category learning well enough, preventing them from creating a good teaching set that will be optimal for both machines and humans (see for example the reverse order of performance between the *LC* student and the human students in Figure 3H). This would mean that better human proxy students are needed. Second, it may be that although generative models may generate good teaching samples, the real dataset contains better teaching samples that are harder to find in the large latent space (at least using our approach). Third, the bi-level optimization method may require better teacher models than a simple search in the latent space of the generative model, e.g., regularization of the distance between the latent codes to prevent teaching samples that are too similar to each other.

In addition to these directions, the results show mediocre results for the human learners, reaching only 69.1% accuracy after learning with the best performing method, while the best ML method reached 79.3% using the same number of realistic samples. While humans are notoriously better than machines in few-shot learning, it may be that these complex biological datasets require auxiliary aids in addition to static teaching sets to enhance learning. For example, new methods such as (Singla et al., 2019) and (Schut et al., 2021), which focus on producing realistic counterfactuals to different samples, may help in providing causal explanations for why does a sample belong to one class or another, possibly assisting the teaching process. Another possible auxiliary aid is providing interpretable visual explanations as for why a sample belongs to a specific class (Aodha et al., 2018). It would be interesting to examine how the performance of human learners changes with additional training images compared to ML method learning curves. This endeavor of finding human-proxy learners is exciting on its own – creating a ML method that could mimic the human learning process, even if not human performance, will allow us to tailor teaching sets to best teach humans properties of the data. This may be done by directly optimizing human-proxy models using human participant and ML method learning data over the same datasets.

We believe the question of how can we best use ML to assist humans in learning about their data is timely: we have both the need, due to large annotated datasets in experimental sciences, and the tools, thanks to the recent bloom of interpretable machine learning methods. Our work here shows that these methods can indeed help in teaching humans, although more work needs to be done to teach humans better than more classical methods.

References

Aodha, O. M., Su, S., Chen, Y., Perona, P., and Yue, Y. Teaching categories to human learners with visual explanations. *CoRR*, abs/1802.06924, 2018. URL http://arxiv.org/abs/1802.06924.

Caie, P. D., Walls, R. E., Ingleston-Orme, A., Daya, S., Houslay, T., Eagle, R., Roberts, M. E., and Carragher, N. O. High-content phenotypic profiling of drug response signatures across distinct cancer cells. *Molecular cancer therapeutics*, 9(6):1913–1926, 2010.

Chen, Y., Aodha, O. M., Su, S., Perona, P., and Yue, Y. Near-optimal machine teaching via explanatory teaching sets. In Storkey, A. and Perez-Cruz, F. (eds.), *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1970–1978. PMLR, 09–11 Apr 2018.

Coleman, C., Yeh, C., Mussmann, S., Mirzasoleiman, B., Bailis, P., Liang, P., Leskovec, J., and Zaharia, M. Selection via proxy: Efficient data selection for deep learning. *CoRR*, abs/1906.11829, 2019. URL http://arxiv.org/abs/1906.11829.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Fan, Y., Tian, F., Qin, T., Li, X., and Liu, T. Learning to teach. *CoRR*, abs/1805.03643, 2018. URL http://arxiv.org/abs/1805.03643.

Graham, B. Kaggle diabetic retinopathy detection competition report, 2015. URL https://www.kaggle.com/c/diabetic-retinopathy-detection.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119, 2020.

Lorraine, J., Vicol, P., and Duvenaud, D. Optimizing millions of hyperparameters by implicit differentiation. *CoRR*, abs/1911.02590, 2019. URL http://arxiv.org/abs/1911.02590.

Nguyen, T. A., Andreis, B., Lee, J., Yang, E., and Hwang, S. J. Stochastic subset selection, 2020.

Pinsler, R., Gordon, J., Nalisnick, E., and Hernández-Lobato, J. M. Bayesian batch active learning as sparse subset approximation, 2020.

Raghu, A., Raghu, M., Kornblith, S., Duvenaud, D., and Hinton, G. Teaching with commentaries, 2020.

Rosch, E. Cognitive representations of semantic categories. *Journal of experimental psychology: General*, 104(3): 192, 1975.

Schut, L., Key, O., Mc Grath, R., Costabello, L., Sacaleanu, B., Gal, Y., et al. Generating interpretable counterfactual explanations by implicit minimisation of epistemic and aleatoric uncertainties. In *International Conference on Artificial Intelligence and Statistics*, pp. 1756–1764. PMLR, 2021.

Sener, O. and Savarese, S. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

Sener, O. and Savarese, S. Active learning for convolutional neural networks: A core-set approach, 2018.

Singla, A., Bogunovic, I., Bartók, G., Karbasi, A., and Krause, A. Near-optimally teaching the crowd to classify. *CoRR*, abs/1402.2092, 2014. URL http://arxiv.org/abs/1402.2092.

Singla, S., Pollack, B., Chen, J., and Batmanghelich, K. Explanation by progressive exaggeration. *arXiv preprint arXiv:1911.00483*, 2019.

Snell, J., Swersky, K., and Zemel, R. S. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

Such, F. P., Rawal, A., Lehman, J., Stanley, K., and Clune, J. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. In *International Conference on Machine Learning*, pp. 9206–9216. PMLR, 2020.

Toneva, M., Sordoni, A., Combes, R. T. d., Trischler, A., Bengio, Y., and Gordon, G. J. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

Trabasso, T. *Attention in learning : theory and research*. R.E. Krieger Pub. Co, Huntington, N.Y, 1975. ISBN 0882752316.

Wang, T., Zhu, J., Torralba, A., and Efros, A. A. Dataset distillation. *CoRR*, abs/1811.10959, 2018. URL http://arxiv.org/abs/1811.10959.