Improving Uncertainty based dataset pruning with Density Estimation for Noisy data

1st Mike Soricelli
Computer & Information Science
University of Massachusetts Dartmouth
Dartmouth, USA
msoricelli@umassd.edu

2nd Youchou Chang
Computer & Information Science
University of Massachusetts Dartmouth
Dartmouth, USA
ychang1@umassd.edu

3rd Christopher J Hixenbaugh
NUWC Division Newport
Naval Sea Systems Command
Newport, USA
christopher.j.hixenbaugh2.civ@us.navy.mil

***Abstract*—With advancements in machine learning methods for image classification, models are getting larger and the need for large datasets is increasing. Although increased sizes of models and datasets lead to improved classification accuracy, it also comes at the cost of increased computational cost and energy consumption to train the classification model. Data-centric approaches have been proposed to address this problem. An area of data-centric approaches focuses on pruning datasets to minimize the size of training data while maintaining model performance. Current State of the Art prediction uncertainty-based dataset pruning methods are effective, but they struggle when the target dataset contains noisy images. In this work, we adapt prior works that use data uncertainty calculations over the course of training to prune easy-to-learn data points from a large scale dataset and add density estimation utilizing normalizing flow models to encourage pruning of outlier data from the dataset to create a novel dataset pruning method. Using density estimation allows our method to perform uncertainty-based pruning, while also removing noisy images from the target dataset. Experimental results show improved accuracy up to 1.11% with moderate noise injection and 4.00% with high levels of noise injection With the weighted score-based approach.**

***Index Terms*—Dataset Pruning, Normalizing Flows, prediction uncertainty**

I. Introduction

Deep learning models for image classification have grown substantially in both parameter count and dataset size, improving accuracy but also increasing the computational resources required for training. Data-centric approaches have been proposed to address this problem. One of these approaches is considered dataset pruning. The goal of data pruning is to remove less informative training samples from the dataset in an effort to maximize the training efficiency of the data samples in our dataset while also reducing the size of the training set. Less informative samples are typically categorized as easy-to-learn samples or hard/noisy samples. One of the state-of-the-art methods in dataset pruning, Dyn-Unc [4], focuses on dynamic prediction uncertainty based calculations to gauge the usefulness of a given data example. Although this method performs well, we hypothesize that there may be shortcomings when it comes to pruning datasets with a lot of noisy samples due to the focus on prediction uncertainty.

To address this potential shortcoming, we develop a data pruning method that utilizes prediction uncertainty and density estimation. More specifically, we adapt prior works that use prediction uncertainty calculations over the course of training to prune easy-to-learn data points from a large-scale dataset and add density estimation to encourage pruning of outlier data from the dataset. To perform density estimation, we leverage normalizing flow models which are able to efficiently perform density estimation through their invertible architecture. With both of these metrics, we are able to develop a score for all of our data by combining the uncertainty and the approximate probability for all our data.

II. Related Works

A. Dataset pruning

Dataset pruning defines a group of methods that aim to reduce the size of a dataset in an effort to reduce training costs. More specifically, the objective of dataset pruning methods is to remove samples that are less efficient for learning. This area of research has resulted in a variety of methods that cut down the size of datasets while maintaining reasonable model performance. [9] attempt to identify forgettable examples during training dynamics, such that removal of forgettable samples does not affect the model performance. Scoring-based methods have been leveraged to measure the relative importance of training examples during the initial training dynamics [7]. Optimization-based sample selection methods were used in [10], with the aim of maintaining data samples which improve the models ability to generalize well. More recent developments in the literature include methods like dynamic uncertainty-based pruning [4]. This method uses variance of sample outputs to gauge prediction uncertainty to differentiate easy-to-learn samples from hard-to-learn samples. Temporal Dual Depth Scoring [12] implements a two-level scoring system that considers both sample contribution and variability of sample contribution.

B. Normalizing Flows

Normalizing Flows are a class of generative models that learns invertible transformations from a basic distribution to a target distribution. This basic distribution is typically a standard normal distribution. Due to this invertible constraint,

The work is supported by the Office of Naval Research (ONR) grant N000142312123

normalizing flows are able to perform efficient density estimation by computing likelihoods in the space of the basic distribution and converting the likelihoods into the space of the target distribution by using the invertible architecture and a change of variables formula. One of the first breakthroughs for normalizing flow models was in 2015 with the publication of the RealNVP method which presented a scalable and efficient method for normalizing flows [2]. In 2017, masked autoregressive flows were introduced in [6], these flow models use autoregressive neural networks to represent the transformations in the flow model. In 2018, the Glow model was published [**?**]. The Glow model introduced an invertible 1x1 convolution which improved expressiveness over the RealNVP model. Neural Spline Flows [3] utilize monotonic rational-quadratic splines, which allow for more expressive capabilities as opposed to affine affine coupling and autoregressive layers, while maintaining invertibility and tractable Jacobians. In more recent years, Residual flows were proposed [1] which use residual networks, as well as transformer based flows [11], which utilize transformer based blocks for flow architecture.

III. Methods

The goal of this method is to improve the capabilities of uncertainty-based pruning by leveraging density estimation to improve training accuracy when pruning datasets with outlier or corrupted data. In this work, we adapt the uncertainty-based pruning technique established from [4] to calculate prediction uncertainty of our data and we use a normalizing flow model to compute likelihoods of data. We combine both of these calculations in a weighted sum to form a score that we can use to rank our data samples.

A. prediction uncertainty

To calculate prediction uncertainty according to [4], we train a model on the target dataset and analyze the training dynamics to estimate prediction uncertainty. More specifically, we track the class probabilities for each data sample at each epoch and then calculate how much variance there is in prediction outcomes for each sample. Instead of calculating this uncertainty for the class predictions for every epoch at once, the uncertainty is calculated over sliding windows (equation 1), dividing out the entire training dynamics into separate subsets. The uncertainty for all windows is added up to calculate the uncertainty for a given data sample, which can be seen in equation 2.

While using prediction uncertainty scores is effective for pruning easy-to-learn samples, it may struggle with pruning samples with considerable amounts of feature noise. As the training data becomes corrupted with noise the features extracted from the network have less significance for the corrupted data samples. This can lead the model to generate similar and ambiguous features for corrupted samples. These corrupted samples also have different class labels. Since these samples produce similar features with varying class labels, the model will struggle to produce a stable mapping from features to class labels for corrupted data samples, leading to prediction uncertainty throughout training for corrupted samples. Because of these considerations, the assumption can be made that prediction uncertainty will be high for corrupted data samples, which will lead the dynamic uncertainty pruning method to maintain corrupted data samples, which would detrimental for classification performance when training with the pruned dataset. With this consideration, we propose using likelihood estimation to create a score-based pruning method that mixes prediction uncertainty and likelihood estimations.

$$U_k(x) = \sqrt{\frac{1}{J}\sum_{j=0}^{J-1}\left(P_j(y_i|x_i) - \frac{1}{J}\sum_{j=0}^{J-1}P_j(y_i|x_i)\right)^2} \qquad (1)$$

$$\mathrm{U}(x) = \frac{1}{K}\sum_{n=0}^{N-1}U_k(x_i) \qquad (2)$$

B. likelihood estimation

To perform likelihood estimation, we train a normalizing flow model on a clean dataset and utilize it to calculate likelihoods of our data samples. Normalizing Flows are a powerful class of generative models that have the ability to compute exact likelihoods by imposing constraints on the architecture. These constraints include that the layers perform invertible transformations and that they have tractable determinants of their Jacobian matrices. Typically, a flow model will be designed to transform the data distribution to a simple distribution such as a standard normal distribution through the invertible transformations. Considering this mapping and the constraints, we can use normalizing flow models to compute likelihoods of a given data sample. We can accomplish this by transforming our data sample into a standard normal distribution and calculating the likelihood according to the standard Normal which is easy to compute. Once this is computed, we can calculate the likelihood according the modeled data distribution by transforming our data back from the standard normal to the target distribution and performing a change of variables formula, which can be seen in equation 3 where X is the target distribution and Z is a standard normal distribution. For the change of variables formula we need to calculate the determinant of the Jacobian for each respective layer. It is worth reiterating that the flow model is constrained to have tractable determinants of the Jacobian matrices by restricting the Jacobian matrices of each layer to be triangular, allowing the determinant to be computed by just considering the product of the diagonal elements.

$$\log p_X(x) = \log p_Z(f(x)) + \log\left|\det\frac{\partial f(x)}{\partial x}\right| \qquad (3)$$

C. pruning

To perform pruning, we consider both calculated prediction uncertainty for each sample and the likelihood of each data sample using a pretrained flow model. To decide which samples to prune, we use a weighted sum of the two metrics to generate a pruning score. The weighted sum can be seen in equation 4, where λ is a hyperparameter between 0-1. With this pruning score, we can prune our dataset at our desired ratio, by sorting our samples by the pruning score in ascending order and removing the first N*R samples where N is the size of the original dataset and R is the pruning ratio. The entire pruning procedure can be seen in algorithm 1.

$$\text{score} = \lambda \cdot \text{uncertainty} + (1 - \lambda) \cdot \text{density} \tag{4}$$

Algorithm 1 Dataset Pruning with Uncertainty and Flow Probability

```
Target dataset $\mathcal{D}$, pruning ratio $R$, normalizing flow model $F$, uncertainty estimator $U$
for each data sample $x_i$ in $\mathcal{D}$ do
    Calculate uncertainty $u_i \leftarrow U(x_i)$
    Calculate probability $p_i \leftarrow F(x_i)$
end for
for each data sample $x_i$ in $\mathcal{D}$ do
    Compute score $s_i \leftarrow \lambda \cdot u_i + (1-\lambda) \cdot p_i$ {$\lambda$ is a weighting coefficients}
end for
Sort all samples in $\mathcal{D}$ by score $s_i$ (ascending or descending as appropriate)
Prune the lowest (or highest) $R$ proportion of samples from $\mathcal{D}$
```

IV. Experiments

A. Experimental Settings

To test our method, we prune a corrupted variant of 2 benchmark image data set (CIFAR10, CIFAR100 [5] and then train a classification model and evaluate each pruning method based on classification accuracy of the trained models. The corrupted variants of these datasets are created by adding random gaussian noise to half of the images from the original dataset. Pruning is done using algorithm 1. We first evaluate prediction uncertainty for each data sample, then we evaluate the likelihoods of each data sample by using a normalizing flow model that is trained on the clean datasets with no noise added.

To perform these experiments, we utilize a standard ResNet9 architecture for our image classification model. We also utilize a Glow normalizing Flow model to perform likelihood estimation developed using the normFlows python package [8]. For classification training settings we use Adam optimizer with a constant learning rate of 0.001. We train each model for 50 epochs with a batch size of 256, data augmentations were also used during training.

Experiments were conducted with code developed using the PyTorch framework. To run our experiments, we utilizing the Google Colab environment with a T4 GPU and we also run test using the Nvidia Jetson AGX Orin developer kit.

B. Performance

To evaluate performance of our method, we perform pruning with the proposed method and compare the performance to the dynamic uncertainty pruning method [4]. For these tests, we set the weighting parameter λ = 0.5 which we find to be a good value in experimentation. We perform tests using 2 different pruning ratios (0.5, 0.25). For this experiment half of the data samples have random gaussian noise added, for each test run the corrupted data samples are randomized. The average results over multiple runs can be seen in table 1. The results show that our proposed method outperforms dynamic uncertainty when there is a moderate amount of noise added for both of these datasets. For CIFAR10, we achieve a 0.50% accuracy increase and a 1.55% accuracy increase with pruning ratios of 0.5 and 0.25 respectively. For CIFAR100, we achieve a 0.99% increase and a 0.87% increase with the same pruning ratios.

TABLE I

Dataset	**Method**	**r = 0.5**	**r = 0.25**
CIFAR-10	dyn-Unc	0.8074	0.7370
	Ours	0.8124	0.7525
CIFAR-100	dyn-Unc	0.5133	0.4385
	Ours	0.5232	0.4472

C. Pruning with different noise levels

We perform more tests to see the effect of different using different standard deviation levels of noise to corrupt the data. Table 2 shows the results these experiments, where we test using low, medium, and high standard deviations of noise(0.1, 0.25, and 0.5) when training on CIFAR10. Figures for the different noise levels can be seen in figure 1. These results show that using likelihood scores is more important with large amounts of noise, while dynamic uncertainty-based pruning outperforms our hybrid approach with low amounts of noise. This behavior is expected, as larger amounts of noise would lead to stronger accuracy degradation if corrupted samples are not pruned, thus using likelihoods to prune more corrupted samples would lead to better performance. We are able to achieve an improved accuracy by 0.45% with moderate noise injection and 2.54% with high levels of noise injection for a pruning ratio of 0.5. We see even more benefits with a smaller pruning ratio of 0.25, achieving 1.11% increase with moderate noise and a 4.00% increase with high levels of noise.

TABLE II

Noise Level (Std)	**Method**	**r = 0.5**	**r = 0.25**
0.5	dyn-Unc	0.7992	0.7264
	Ours	0.8246	0.7664
0.25	dyn-Unc	0.8077	0.7362
	Ours	0.8122	0.7473
0.1	dyn-Unc	0.8191	0.7445
	Ours	0.7821	0.7413

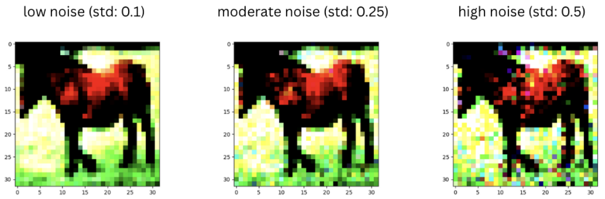

Fig. 1.

V. Discussion

Our results show that our proposed method outperforms the dynamic uncertainty-based pruning approach when the noise levels of training data are moderate to high. Using likelihoods estimation allows our method to filter out noisy images while also considering prediction uncertainty to achieve a high validation accuracy with a pruned dataset in these settings.

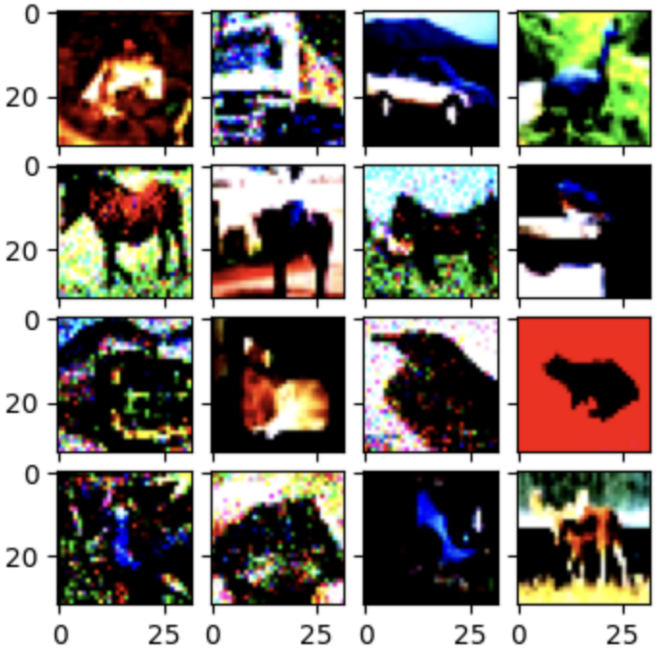

Fig. 2.

A. analysis

For further analysis, we looked directly at how uncertainty-based pruning handles noisy data by measuring the percentage of noisy images that are in the pruned dataset. What we find is that uncertainty-based pruning yields a dataset that contains 54.8% noisy images. This is an interesting result, because this is actually higher than the percentage of noisy images in the original full data set (50%). This result highlights the difficulties that arise with using uncertainty-based pruning with noisy datasets and provides empirical evidence for our hypothesis that noisy images will yield high prediction uncertainty throughout training. Figure 2 shows a subset of images that results from dynamic uncertainty-based pruning, which contains many noisy images.

B. future work

Future work considerations include the following:

hyperparameter learning: For future work, we will look at setting our weighting parameter λ to be a learnable parameter. To accomplish this, we will setup a bi-level optimization scheme with a differentiable soft pruning probabilistic pruning approach, such that we are learning the optimal λ according to validation performance of our model.

improved density estimation: To further enhance this work we look to experiment with some of the latest state-of-the-art density estimators to see if we can improve our performance with improved distribution modeling for better likelihood estimation.

Comparative analysis: To provide stronger empirical evidence for using the proposed method, our objective is to perform comparative tests with other state-of-the-art data pruning methods in our experimental settings.

VI. conclusion

In this work, we proposed an improved method for performing uncertainty-based data set pruning in noisy training settings by leveraging normalizing flow models to perform density estimation. Uncertainty-based pruning is a valuable pruning approach, but it suffers from accuracy degradation when moderate to high noise levels are introduced to the training data. With our proposed weighted score-based approach, we are able to improve accuracy to 1.11% with moderate noise injection and 4.00% with high levels of noise injection. Our results indicate that the proposed method can be a useful dataset pruning method for datasets with moderate to high amounts of noise. Although there are future areas of improvement for this method, the proposed work does present a way to improve uncertainty-based pruning for noisy data.

References

[1] Ricky T. Q. Chen, Jens Behrmann, David Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling, 2020.

[2] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp, 2017.

[3] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows, 2019.

[4] Muyang He, Shuo Yang, Tiejun Huang, and Bo Zhao. Large-scale dataset pruning with dynamic uncertainty. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 7713–7722, 2023.

[5] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[6] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation, 2018.

[7] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[8] Vincent Stimper, David Liu, Andrew Campbell, Vincent Berenz, Lukas Ryll, Bernhard Schölkopf, and José Miguel Hernández-Lobato. normflows: A pytorch package for normalizing flows. *Journal of Open Source Software*, 8(86).

[9] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning, 2019.

[10] Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence, 2023.

[11] Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Navdeep Jaitly, and Josh Susskind. Normalizing flows are capable generative models, 2025.

[12] Xin Zhang, Jiawei Du, Yunsong Li, Weiying Xie, and Joey Tianyi Zhou. Spanning training progress: Temporal dual-depth scoring (tdds) for enhanced dataset pruning, 2024.