Estimating Example Difficulty using Variance of Gradients

Chirag Agarwal
MDSR Lab, Adobe
chiragagarwall12@gmail.com

Daniel D'souza
ML Collective
ddsouza@umich.edu

Sara Hooker
Google Research
shooker@google.com

Abstract

In machine learning, a question of great interest is understanding what examples are challenging for a model to classify. Identifying atypical examples ensures the safe deployment of models, isolates samples that require further human inspection and provides interpretability into model behavior. In this work, we propose Variance of Gradients (VoG) as a valuable and efficient metric to rank data by difficulty and to surface a tractable subset of the most challenging examples for human-in-the-loop auditing. We show that data points with high VoG scores are far more difficult for the model to learn and over-index on corrupted or memorized examples. Further, restricting the evaluation to the test set instances with the lowest VoG improves the model's generalization performance. Finally, we show that VoG is a valuable and efficient ranking for out-of-distribution detection.

1. Introduction

Over the past decade, machine learning models are increasingly deployed to high-stake decision applications such as healthcare [4, 20, 52, 70], self-driving cars [51] and finance [53]. For gaining trust from stakeholders and model practitioners, it is important for deep neural networks (DNNs) to make decisions that are interpretable to both researchers and end-users. To this end, for sensitive domains, there is an urgent need for auditing tools which are scalable and help domain experts audit models.

Reasoning about model behavior is often easier when presented with a subset of data points that are relatively more difficult for a model to learn. Besides aiding interpretability through case-based reasoning [11, 30, 39], it can also be used to surface a tractable subset of atypical examples for further human auditing [46, 73], for active learning to inform model improvements, and to choose not to classify some instances when the model is uncertain [7, 14, 21]. One of the biggest bottlenecks for human auditing is the large scale of modern datasets and the cost of annotating individual features [3, 38, 68]. Methods which automatically surface a subset of relatively more challenging examples for human inspection help prioritize limited human annotation and auditing time. Despite the urgency of this use-case, ranking examples by difficulty has had limited treatment in the context of deep neural networks due to the computational cost of ranking a high dimensional feature space.

Present work. A popular interpretability tool is saliency maps, where each of the features of the input data are scored based on their contribution to the final output [64]. However, these explanations are typically for a single prediction and generated after the model is trained. Our goal is to leverage these explanations to automatically surface a subset of relatively more challenging examples for human inspection to help prioritize limited human annotation and auditing time. To this end, we propose a ranking method across all examples that instead measures the per-example change in explanations over training. Examples that are difficult for a model to learn will exhibit higher variance in gradient updates throughout training. On the other hand, the backpropagated gradients of the samples that are *relatively easier* will exhibit lower variance because the loss from these examples does not consistently dominate the model training.

We term this class normalized ranking mechanism *Variance of Gradients* (VoG) and demonstrate that VoG is a meaningful way for ranking data by difficulty and surfacing a tractable subset of the most challenging examples for human-in-the-loop auditing across a variety of large-scale datasets. VoG assigns higher scores to test set examples that are more challenging for the model to classify and proves to be an efficient tool for detecting out-of-distribution (OoD) samples. VoG is model and domain-agnostic as all that is required is the backpropagated gradients from the model.

Contributions. We demonstrate consistent results across two architectures and three datasets – Cifar-10, Cifar-100 [43] and ImageNet [61]. Our contributions can be enumerated as follows:

1. We present Variance of Gradients (VoG) – a class-normalized gradient variance score for determining the relative ease of learning data samples within a given

class (Sec. 2). VoG identifies clusters of images with clearly distinct semantic properties, where images with low VoG scores feature far less cluttered backgrounds and more prototypical vantage points of the object (Fig. 4). In contrast, images with high VoG scores over-index on images with cluttered backgrounds and atypical vantage points of the object of interest.

2. VoG effectively surfaces memorized examples, *i.e.* it allocates higher scores to images that require *memorization* (Sec. 4). Further, VoG aids in understanding the model behavior at different training stages and provides insight into the learning cycle of the model.

3. We show the reliability of VoG as an OoD detection technique and compare its performance to 9 existing OoD methods, where it outperforms several methods, such as PCA [24] and KDE [15, 54]. VoG presents an overall improvement of 9.26% in precision compared to all other methods.

2. VoG Framework

We consider a supervised classification problem where a DNN is trained to approximate the function $\mathcal{F}$ that maps an input variable $\mathbf{X}$ to an output variable $\mathbf{Y}$, formally $\mathcal{F} : \mathbf{X} \mapsto \mathbf{Y}$, where $\mathbf{Y}$ is a discrete label vector associated with each input $\mathbf{X}$ and $y \in \mathbf{Y}$ corresponds to one of C categories or classes in the dataset.

A given input image $\mathbf{X}$ can be decomposed into a set of pixels x_i, where $i = \{1, \ldots, N\}$ and N is the total number of pixels in the image. For a given image, we compute the gradient of the activation A_p^l with respect to each pixel x_i, where l designates the pre-softmax layer of the network and p is the index of either the true or predicted class probability. We would like to note that the pre-softmax layer is responsible for connecting activations from previous layers in the network to individual class scores. Hence, computing the gradients w.r.t. this class indexed score measures the contribution of features to the final class prediction [64].

Note our goal is to rank examples, so for each example, we compute the pre-softmax activation gradient indexed at predicted/true label with respect to the input. This is far more computationally efficient than computing the full Jacobian matrix with individual layers.

Let $\mathbf{S}$ be a matrix that represents the gradient of A_p^l with respect to individual pixels x_i, *i.e.* for an image of size $3 \times 32 \times 32$, the gradient matrix $\mathbf{S}$ will be of dimensions $3 \times 32 \times 32$.

$$\mathbf{S} = \frac{\partial A_p^l}{\partial x_i} \tag{1}$$

This formulation may feel familiar as it is often computed based upon the weights of a trained model and visualized as a image heatmap for interpretability purposes [5, 31, 63, 64, 64–66]. In contrast to saliency maps which are inherently local explanation tools, we are leveraging relative changes in gradients across training to rank all examples globally.

Following several seminal papers in explainability literature [31, 63–66], we take the average over the color channels to arrive at a gradient matrix [63–66] where $\mathbf{S} \in \mathbb{R}^{32\times32}$. For a given set of K checkpoints, we generate the above gradient matrix $\mathbf{S}$ for all individual checkpoints, *i.e.*, $\{\mathbf{S}_1, \ldots, \mathbf{S}_K\}$. We then calculate the mean gradient μ by taking the average of the K gradient matrices. Note, μ is the mean across different checkpoints and is of the same size as the gradient matrix $\mathbf{S}$. We then calculate the variance of gradients across each pixel as:

$$\mu = \frac{1}{K}\sum_{t=1}^{K}\mathbf{S}_t. \tag{2}$$

$$\text{VoG}_p = \sqrt{\frac{1}{K}\sum_{t=1}^{K}(\mathbf{S}_t - \mu)^2}. \tag{3}$$

We average the pixel-wise variance of gradients to compute a scalar VoG score for the given input image:

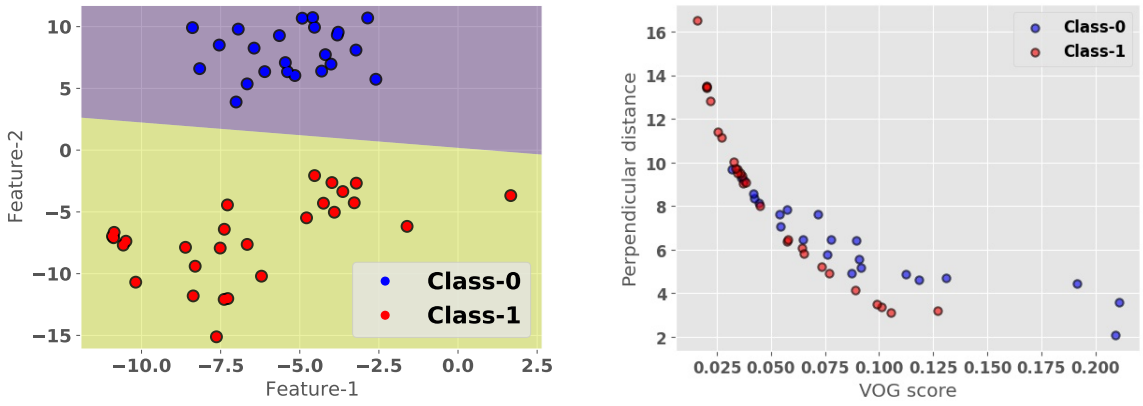
$$\text{VoG} = \frac{1}{N}\sum_{t=1}^{N}(\text{VoG}_p), \tag{4}$$

where N is the total number of pixels in a given image. First calculating the pixel-wise variance (Eqn. 3) and then average over the pixels (Eqn. 4) is consistent with previous XAI works where the gradients of an input image are computed independently for each pixel in an image [64–66].

In order to account for inherent differences in variance between classes, we normalize the absolute VoG score by class-level VoG mean and standard deviation. This amounts to asking: *What is the variance of gradients for a given image with respect to all other exemplars of this class category?*

2.1. Validating the behavior of VoG on synthetic data

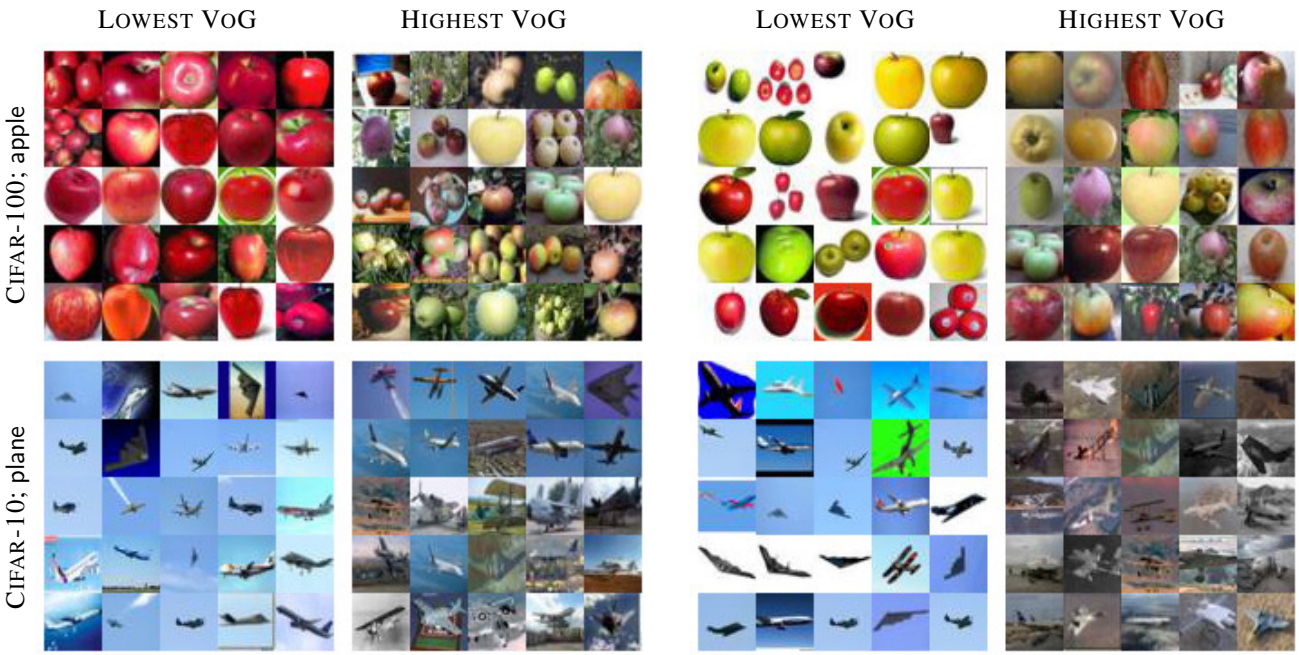
In Fig. 1a, we illustrate the principle and effectiveness of VoG in a controlled toy example setting. The data was generated using two separate isotropic Gaussian clusters. In such a simple low dimensional problem, the most challenging examples for the model to classify can be quantified by distance to the decision boundary. In Fig. 1a, we visualize the trained decision boundary of a multiple layer perceptron (MLP) with a single hidden layer trained for 15 epochs. We compute VoG for each training data point and plot final VoG score for each point against the distance to the trained boundary. In Fig. 1b, we can see that VoG successfully ranks highest the examples closest to the decision



(a) Toy dataset trained decision boundary

(b) Distance vs. VoG score

Figure 1. **Left:** Variance of Gradients (VoG) for each testing data point in the two-dimensional toy problem. **Right:** VoG accords higher scores to the most challenging examples closest to the decision boundary (as measured by the perpendicular distance).



(a) Early-stage training

(b) Late-stage training

Figure 2. The 5×5 grid shows the top-25 Cifar-10 and Cifar-100 training-set images with the lowest and highest VoG scores in the *Early* (a) and *Late* (b) training stage respectively of two randomly chosen classes. Lower VoG images evidence uncluttered backgrounds (for both apple and plane) in the *Late* training stage. VoG also appears to capture a color bias present during the *Early* training stage for both apple (red). The VoG images in *Late* training stage present unusual vantage points, with images where the frame is zoomed in on the object of interest.

boundary. The most challenging examples exhibit the greatest variance in gradient updates over the course of the training process. In the following sections, we will scale this toy problem and show consistent results across multiple architectures and datasets.

2.2. Experimental Setup

Datasets. We evaluate our methodology on Cifar-10 and Cifar-100 [43], and ImageNet [61] datasets. For all datasets, we compute VoG for both training and test sets.

Cifar Training. We use a ResNet-18 network [25] for both Cifar-10 and Cifar-100. For each dataset, we train the model for 350 epochs using stochastic gradient descent (SGD) and compute the input gradients for each sample every 10 epochs. We implemented standard data augmentation by applying cropping and horizontal flips of input images. We use a base learning rate schedule of 0.1 and adaptively change to 0.01 at 150^{th} and 0.001 at 250^{th} training epochs. The top-1 test set accuracy for Cifar-10 and Cifar-100 were 89.57% and 66.86% respectively.

ImageNet Training. We use a ResNet-50 [25] model for training on ImageNet. The network was trained with batch normalization [35], weight decay, decreasing learning rate schedules, and augmented training data. We train for $32,000$ steps (approximately 90 epochs) on ImageNet with a batch size of 1024. We store 32 checkpoints over the course of training, but in practice observe that VoG ranking is very stable computed with as few as 3 checkpoints. Our model achieves a top-1 accuracy of 76.68% and top-5 accuracy of 93.29%.

Number of checkpoints. The number of checkpoints used to compute VoG balances efficiency for practitioners to use with the robustness of ranking. This can be set by the practitioner, and we note that in practice the last 3 checkpoints are sufficient for a robust VoG ranking (minimal difference when restricting to the last 3 in Figs. 5b,8b,11b vs. evaluating on all checkpoints in Fig. 4). For all experiments, VoG(*early*-stage) is computed using checkpoints from the first 3 epochs and VoG(*late*-stage) is computed using checkpoints from the last 3 epochs. The test set accuracy at the *early*-stage is 44.65%, 14.16%, and 51.87% for Cifar-10, Cifar-100, and ImageNet, respectively. In the *late*-stage it is 89.57%, 66.86%, and 76.68% for Cifar-10, Cifar-100, and ImageNet, respectively.

3. Utility of VoG as an Auditing Tool

In this section, we evaluate the merits of VoG as an auditing tool. Specifically, we (1) present the qualitative properties of images at both ends of the VoG spectrum, (2) measure how discriminative VoG is at separating easy examples from difficult, (3) quantify the stability of the VoG ranking, (4) use VoG as an auditing tool for test dataset, and (5) leverage VoG to understand the training dynamics of a DNN.

1) Qualitative inspection of ranking. A qualitative inspection of examples with high and low VoG scores shows that there are distinct semantic properties to the images at either end of the ranking. We visualize 25 images ranked lowest and highest according to VoG for both the entire dataset (visualized for ImageNet in Fig. 7) and for specific classes (visualized for ImageNet in Fig. 3 and for Cifar-10 and Cifar-100 in Fig. 2). Images with *low* VoG score tend to have uncluttered and often white backgrounds with the object of interest centered clearly in the frame. Images with the *high* VoG scores have cluttered backgrounds and the object of interest is not easily distinguishable from the background. We also note that images with high VoG scores tend to feature atypical vantage points of the objects such as highly zoomed frames, side profiles of the object or shots taken from above. Often, the object of interest is partially occluded or there are image corruptions present such as heavy blur.

2) Test set error and VoG. A valuable property of an auditing tool is to effectively discriminate between easy and challenging examples. In Fig. 4, we plot the test set error of examples bucketed by VoG decile. Note that we plot error, so lower is better. We show that examples at the lowest percentiles of VoG have low error rates, and misclassification increases with an increase in VoG scores. Our results are consistent across all datasets, yet the trend is more pronounced for more complex datasets such as Cifar-100 and ImageNet. We ascribe this to differences in underlying model complexity. Furthermore, in Fig. 10, we observe that test set error on the lowest VoG scored images are lower than the baseline test set performance.

3) Stability of VoG ranking. To build trust with an end-user, a key desirable property of any auditing tool is consistency in performance. We would expect a consistent method to produce a ranking with a closely bounded distribution of scores across independently trained runs for a given model and dataset. To measure the consistency of the VoG ranking, we train five Cifar-10 networks from random initialization following the training methodology described in Sec. 2.2. Empirically, Fig. 6 shows that VoG rankings evidence a consistent distribution of test-error at each percentile given the same model and dataset. For completeness, we also measure instance-wise VoG stability by computing the standard deviation of VoG scores for 50k Cifar-10 samples across 10 independent initializations. The standard deviation of the VoG scores is negligible with a mean deviation of $3.81e^{-9}$ across all samples. In addition, we find similar results for Cifar-100 dataset where the output VoG scores are stable (mean std of $9.6e{-}6$) across different model initializations. Finally, we extend our stability experiments to understand the effect of different training hyperparameter settings (e.g., batch size) on the VoG scores. Here, we train 5 Cifar-10 models using different batch sizes, i.e., $\{128, 256, 384, 512, 640\}$, and find that the mean VoG standard deviation across 50k Cifar-10 samples was $1.9e{-}5$.

4) VoG as an unsupervised auditing tool. Many auditing tools used to evaluate and understand possible model bias require the presence of labels for protected attributes and underlying variables. However, this is highly infeasible in real-world settings [68]. For image and language datasets, the high dimensionality of the problem makes it hard to

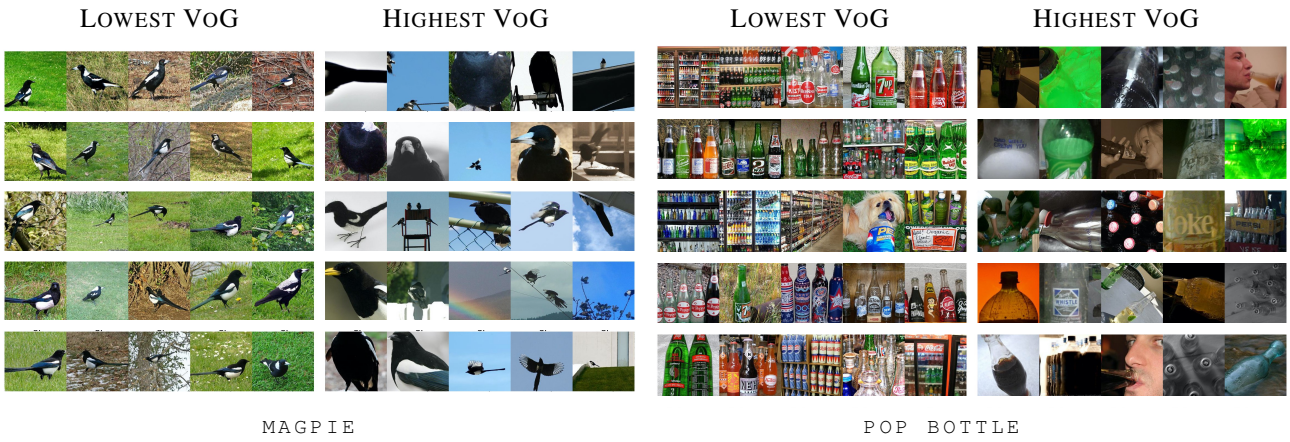
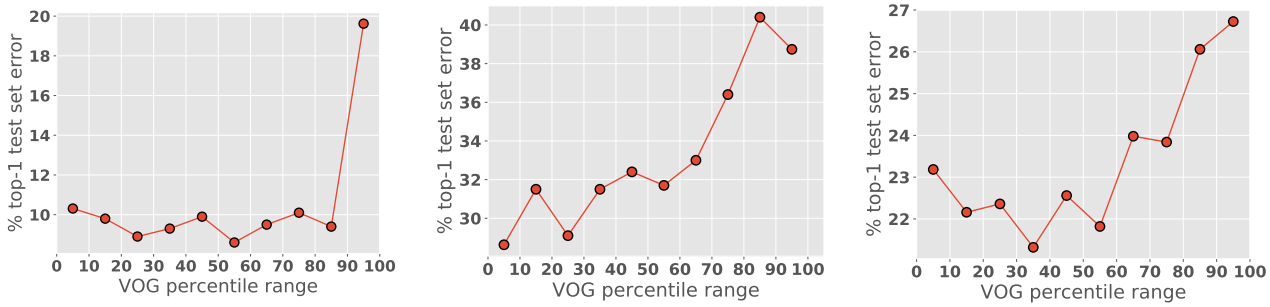


Figure 3. Each 5×5 grid shows the top-25 ImageNet training-set images with the lowest and highest VoG scores for the class `magpie` and `pop bottle`. Training set images with higher VoG scores tend to feature zoomed-in images with atypical color schemes and vantage points.



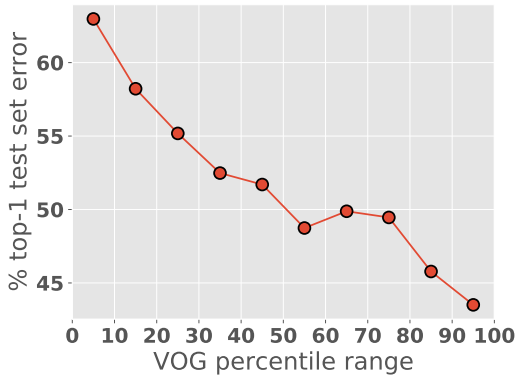
(a) Cifar-10 (b) Cifar-100 (c) ImageNet

Figure 4. The mean top-1 test set error (y-axis) for the examples thresholded by VoG score percentile (x-axis). Across Cifar-10, Cifar-100 and ImageNet, mis-classification increases with an increase in VoG scores. Across all datasets the group of samples in the top-10 percentile VoG scores have the highest error rate, *i.e.* contains most number of misclassified samples.

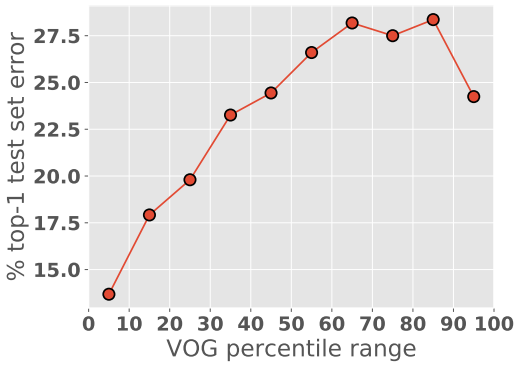
identify a priori what underlying variables one needs to be aware of. Even acquiring the labels for a limited number of attributes protected by law (gender, race) is expensive and/or may be perceived as intrusive, leading to noisy or incomplete labels [2, 29]. This means that ranking techniques which do not require labels at test time are very valuable.

One key advantage of VoG is that we show it continues to produce a reliable ranking even when the gradients are computed *w.r.t.* the predicted label. In Fig. 7, we include the top and bottom 25 VoG ImageNet test images using predicted labels from the model. Finally, we also computed the mean test-error for the predicted VoG distribution, and find that it also effectively discriminates between top-10 and bottom-10 examples, respectively (Fig. 12a).

5) VoG understands early and late training dynamics. Recent works have shown that there are distinct stages to training in deep neural networks [1, 17, 36, 49]. To this end, we investigate whether VoG rankings are sensitive to the stage of the training process. We compute VoG separately for two different stages of the training process: (i) the *Early*-stage (first three epochs) and (ii) the *Late*-stage (last three epochs). We plot VoG scores against the test set error at each decile in early- and late-stage and find a flipping behavior across all datasets and networks (Fig. 5 for ImageNet, Fig. 8 for Cifar-100, and Fig. 11 for Cifar-10). In the early training stage, samples having higher VoG scores have a lower average error rate as the gradient updates hinge on easy examples. This phenomenon reverses during the late-stage of the training, where, across all datasets, high VoG scores in the late-stage have the highest error rates as updates to the challenging examples dominate the computation of variance. Further, we note a noticeable visual difference between the image ranking computed for *early*- and *late*-stages of training. As seen in Fig. 2, for some classes such as *apple*, it appears that VoG scores also



(a) Early-stage training



(b) Late-stage training

Figure 5. The mean top-1 test set error (y-axis) for the examples thresholded by VoG score percentile (x-axis) in ImageNet validation set. The Early (a) and Late (b) stage VoG analysis shows inverse behavior where the role of VoG flips as the training progresses.

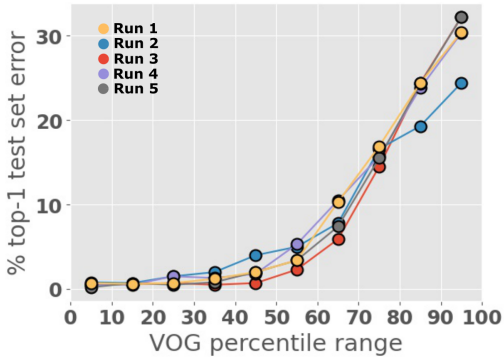


Figure 6. The VoG top-1 test set error for five ResNet-18 networks independently trained on Cifar-10 from random initialization. The plot shows that VoG produces a stable ranking with a similar distribution of error in each percentile across all images

capture the network's color bias during the *early* training stage, where images with the lowest VoG scores over-index on red-colored apples.

4. Relationship between VoG Scores and Memorized/OoD Examples

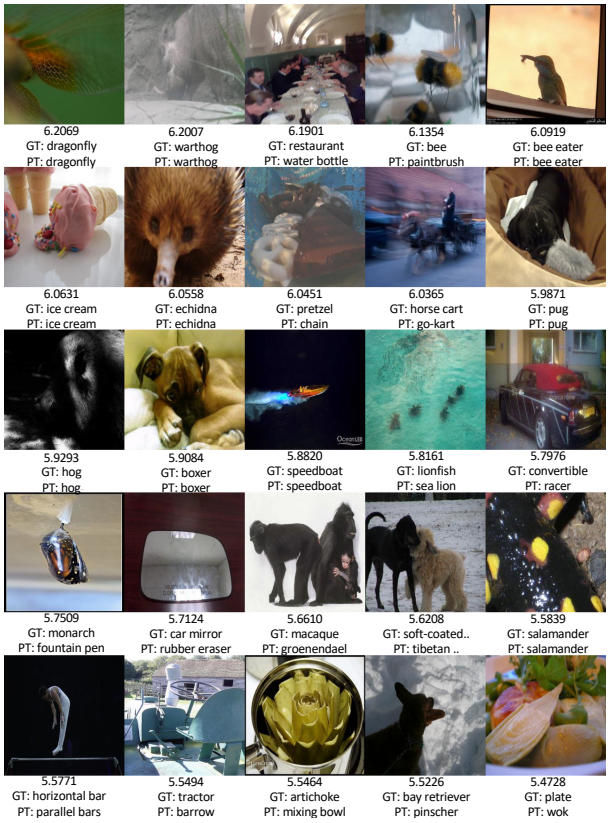
Recent works have highlighted that DNNs produce uncalibrated output probabilities that cannot be interpreted as a measure of certainty [22, 26, 37, 44]. To this end, we argue that if VoG is a reliable auditing tool, it should capture model uncertainty even when it's not reflected in the output probabilities. We consider VoG rankings on a task where the network produces highly confident predictions for incorrect/out-of-distribution inputs and evaluate VoG on two separate tasks: (1) identifying examples memorized by the model and (2) detecting out-of-distribution examples.

4.1. Surfacing examples that require memorization

Overparameterized networks have been shown to achieve zero training error by memorizing examples [19, 32, 72]. We explore whether VoG can distinguish between examples that require memorization and the rest of the dataset. To do this, we replicate the general experiment setup of Zhang et al. [72] and replace 20% of all labels in the training set with randomly shuffled labels. We re-train the model from random initialization and compute VoG scores *across training* for all examples in the training set. Our network achieves 0% training error which would only be possible given successful memorization of the noisy examples with shuffled labels. We now answer the question: *Is VoG able to discriminate between these memorized examples and the rest of the dataset?*

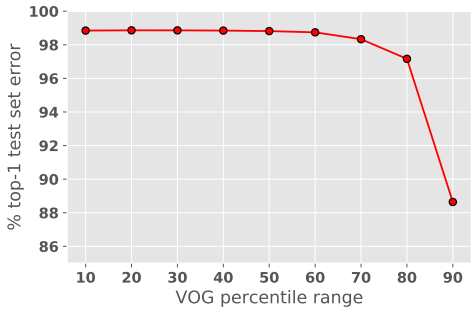
We perform a two-sample t-test with unequal variances [69] and show that this difference is statistically significant at a p-value of 0.001, *i.e.* shuffled labels have a different VoG distribution than the non-shuffled dataset. Intuitively, the two-sample t-test produces a p-value that can be used to decide whether there is evidence of a significant difference between the two distributions of VoG scores. The p-value represents the probability that the difference between the sample means is large, *i.e.* the smaller the p-value, the stronger is the evidence that the two populations have different means. For both Cifar-10 and Cifar-100, we find a statistically significant difference in VoG scores for each population (p-value is < 0.001), which shows that VoG is discriminative at distinguishing between memorized and non-memorized examples. We include more details about the statistical testing in Sec. C.

(a) Lowest VoG

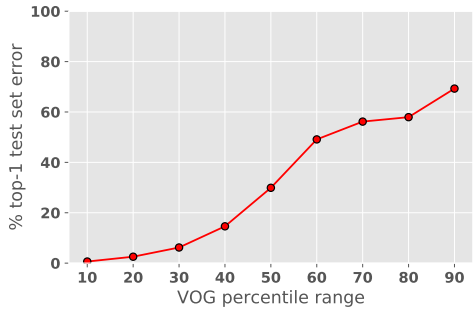


(b) Highest VoG

Figure 7. Each 5×5 grid shows the top-25 ImageNet test set images with the lowest and highest VoG scores for the top-1 predicted class. Test set images with higher VoG scores tend to feature zoomed-in images and are misclassified more as compared to the lower VoG images which tend to feature more prototypical vantage points of objects.



(a) Early-stage training



(b) Late-stage training

Figure 8. The mean top-1 test set error (y-axis) for the exemplars thresholded by VoG score percentile (x-axis) in Cifar-100 testing set. The early (a) and late (b) stage VoG analysis shows inverse behavior where the role of VoG flips as the training progresses. Results for Cifar-10 are shown in Appendix Fig. 11.

4.2. Out-of-Distribution detection

We have already established that VoG is very effective at distinguishing between easy and challenging examples (Fig. 10). Here, we ask whether this makes VoG an effective out of distribution (OoD) detection tool. It also gives us a setting in which to compare VoG as a ranking mechanism to other methods

Ruff et al. [59] benchmark a variety of OoD detection techniques on MNIST-C [50]. For completeness, we repli-

cate this precise setup by using a trained LeNet model and evaluate VoG on MNIST-C against 9 other methods [12, 41, 45, 56–58, 60, 62, 67].

Evaluation metrics. We evaluate OoD detection performance using the following metrics:

i) AUROC. The Area Under the Receiver Operator Characteristic (AUROC) curve can be interpreted as the probability that a positive example is assigned a higher detection score than a negative example [18].

ii) AUPR (In). The Area Under the Precision Recall (AUPR) curve computes the precision-recall pairs for different probability thresholds by considering the in-distribution examples as the positive class.

iii) AUPR (Out). AUPR (Out) is AUPR as described above, but calculated considering the OoD examples as the positive class. We treat this outlier class as positive by multiplying the VoG scores by -1 and labelling them positive when calculating AUPR (Out).

Table 1. Comparison of VoG to 9 existing OoD detection methods. Shown are average values of metrics and standard deviations across 15 corruptions in the MNIST-C datasets. Arrows (↑) indicate the direction of better metric performance. VoG outperforms most baselines by a large margin.

OoD methods	AUROC (↑)	AUPR OUT (↑)
KDE [57]	57.46±32.09	62.56±24.16
MVE [58]	62.84±21.92	61.42±19.1
DOCC [60]	69.16±28.35	70.37±23.25
kPCA [12]	72.12±31.00.	75.39±26.37
SVDD [67]	74.01±21.39	73.33±21.98
PCA [56]	77.71±30.90	80.86±25.2
Gaussian [45]	80.57±29.71	84.51±22.62
VoG	85.42±10.28.	84.96±9.61
AE [41]	89.89±18.52	89.99±18.19
AEGAN [62]	95.93±7.90.	95.40±9.46

Findings. In Table 1, we observe that VoG outperforms all methods except AutoEncoders (AE) and AutoEncoder GAN (AEGAN). In stark contrast to VoG, AE and AEGAN require complex training of auxiliary models and do not feasibly scale beyond small-scale datasets like MNIST. Given these limitations, VoG remains a valuable and scalable OoD detection method as it can be used for large-scale datasets (*e.g.* ImageNet) and networks (*e.g.* ResNet-50). Unlike generative models, VoG does not require an uncorrupted training dataset for learning image distributions. Further, VoG only leverages data from training itself, is computed from checkpoints already stored over the course of training, and does not require the true label to rank.

5. Related Work

Our work proposes a method to rank training and testing data by estimating example difficulty. Given the size of current datasets, this can be a powerful interpretability tool to isolate a tractable subset of examples for human-in-the-loop auditing and aid in curriculum learning [8] or distinguishing between sources of uncertainty [16, 33]. While prior works have proposed different notions of what subset merits surfacing, introduced the concept of prototypes and quintessential examples in the dataset, but did not focus on large-scale deep neural networks models [9, 13, 39, 40, 73].

Unlike previous works, we propose a measure that can be extended to rank the entire dataset by estimating example difficulty (rather than surfacing a prototypical subset). In addition, VoG is far more efficient than other global rankings like [42] and [23].

VoG also does not require modifying the architecture or making any assumptions about the statistics of the input distribution. In particular, works such as [39] require assumptions about the statistics of the input distribution and [47] requires modifying the architecture to prefix an autoencoder to surface a set of prototypes, [55] leverages pruning of the model to identify difficult examples and [6] requires the addition of an auxiliary k-nn model after each layer.

Our work is complementary to recent works by [36] that proposes a c-score to rank examples by aligning them with training instances, [30] that classifies examples as outliers according to sensitivity to varying model capacity, and [10] that considers different measures to isolate prototypes for ranking the entire dataset. We note that the c-score method proposed by [36] is considerably more computationally intensive to compute than VoG as it requires training up to 20,000 network replications per dataset. Several of the prototype methods considered by [10] require training ensembles of models, as does the compression sensitivity measure proposed by [30]. Finally, our proposed VoG is both different in the formulation and can be computed using a small number of existing checkpoints saved over the course of training.

6. Conclusion and Future Work

In this work, we proposed VoG as a valuable and efficient way to rank data by difficulty and surface a tractable subset of the most challenging examples for human-in-the-loop auditing. High VoG samples are challenging to classify for algorithm and surfaces clusters of images with distinct visual properties. Moreover, VoG is domain agnostic as it uses only the vanilla gradient explanation from the model, and can be used to rank both training and test examples. We show that it is also a useful unsupervised protocol, as it can effectively rank examples using the predicted label.

References

[1] Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep networks. In *ICLR*, 2019. 5

[2] McKane Andrus, Elena Spitzer, Jeffrey Brown, and Alice Xiang. "what we can't measure, we can't understand": Challenges to demographic data procurement in the pursuit of fairness. *CoRR*, abs/2011.02282, 2020. 5

[3] McKane Andrus, Elena Spitzer, Jeffrey Brown, and Alice Xiang. What we can't measure, we can't understand: Challenges to demographic data procurement in the pursuit of fairness. In *FAccT*, 2021. 1

[4] 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. In *NPJ Digital Medicine*, 2019. 1

[5] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. How to explain individual classification decisions. In *JMLR*, 2010. 2

[6] Robert J. N. Baldock, Hartmut Maennel, and Behnam Neyshabur. Deep learning through the lens of example difficulty. *CoRR*, abs/2106.09647, 2021. 8

[7] Peter L. Bartlett and Marten H. Wegkamp. Classification with a reject option using a hinge loss. In *JMLR*, 2008. 1

[8] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009. 8

[9] Jacob Bien and Robert Tibshirani. Prototype selection for interpretable classification. In *The Annals of Applied Statistics*, 2011. 8

[10] Nicholas Carlini, Ulfar Erlingsson, and Nicolas Papernot. Distribution density, tails, and outliers in machine learning: Metrics and applications. *arXiv*, 2019. 8

[11] Rich Caruana. Case-based explanation for artificial neural nets. In *Artificial Neural Networks in Medicine and Biology*, 2000. 1

[12] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Robust, deep and inductive anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 36–51. Springer, 2017. 8

[13] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *NeurIPS*, 2017. 8

[14] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Boosting with abstention. In *NeurIPS*, 2016. 1

[15] Richard A Davis, Keh-Shin Lii, and Dimitris N Politis. Remarks on some nonparametric estimates of a density function. In *Selected Works of Murray Rosenblatt*. Springer, 2011. 2

[16] Daniel D'souza, Zach Nussbaum, Chirag Agarwal, and Sara Hooker. A tale of two long tails, 2021. 8

[17] Fartash Faghri, David Duvenaud, David J Fleet, and Jimmy Ba. A study of gradient variance in deep learning. *arXiv*, 2020. 5

[18] Tom Fawcett. An introduction to roc analysis. In *Pattern recognition letters*, 2006. 8

[19] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *ACM SIGACT Symposium on Theory of Computing*, 2020. 6

[20] Ross Gruetzemacher, Ashish Gupta, and David B. Paradice. 3d deep learning for detecting pulmonary nodules in ct scans. In *JAMIA*, 2018. 1

[21] Abhijit Guha Roy, Jie Ren, Shekoofeh Azizi, Aaron Loh, Vivek Natarajan, Basil Mustafa, Nick Pawlowski, Jan Freyberg, Yuan Liu, Zach Beaver, Nam Vo, Peggy Bui, Samantha Winter, Patricia MacWilliams, Greg S. Corrado, Umesh Telang, Yun Liu, Taylan Cemgil, Alan Karthikesalingam, Balaji Lakshminarayanan, and Jim Winkens. Does your dermatology classifier know what it doesn't know? detecting the long-tail of unseen conditions. *Medical Image Analysis*, 75:102274, 2022. 1

[22] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017. 6

[23] Hrayr Harutyunyan, Alessandro Achille, Giovanni Paolini, Orchid Majumder, Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Estimating informativeness of samples with smooth unique information. In *ICLR*, 2021. 8

[24] Douglas M. Hawkins. The detection of errors in multivariate data using principal components. In *Journal of the American Statistical Association*, 1974. 2

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 14

[26] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR*, 2017. 6, 14

[27] Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained Transformers Improve Out-of-Distribution Robustness. page arXiv, Apr. 2020. 14

[28] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021. 13

[29] Sara Hooker. Moving beyond "algorithmic bias is a data problem". *Patterns*, 2(4):100241, 2021. 5

[30] Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget? *arXiv*, 2019. 1, 8

[31] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *NeurIPS*, 2019. 2

[32] Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. Characterising bias in compressed models. *arXiv*, 2020. 6

[33] Niel Teng Hu, Xinyu Hu, Rosanne Liu, Sara Hooker, and Jason Yosinski. When does loss-based prioritization fail?, 2021. 8

[34] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 14

[35] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 4

[36] Ziheng Jiang, Chiyuan Zhang, Kunal Talwar, and Michael C Mozer. Characterizing structural regularities of labeled data in overparameterized models. In *ICML*, 2021. 5, 8

[37] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017. 6

[38] Zaid Khan and Yun Fu. One label, one billion faces. In *FAccT*, 2021. 1

[39] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *NeurIPS*, 2016. 1, 8

[40] Been Kim, Cynthia Rudin, and Julie A Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *NeurIPS*, 2014. 8

[41] Ki Hyun Kim, Sangwoo Shim, Yongsub Lim, Jongseob Jeon, Jeongwoo Choi, Byungchan Kim, and Andre S Yoon. Rapp: Novelty detection with reconstruction along projection pathway. In *International Conference on Learning Representations*, 2019. 8

[42] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017. 8

[43] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1, 3

[44] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017. 6

[45] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018. 8

[46] Christian Leibig, Vaneeda Allken, Murat Seçkin Ayhan, Philipp Berens, and Siegfried Wahl. Leveraging uncertainty information from deep neural networks for disease detection. In *Scientific reports*, 2017. 1

[47] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *AAAI*, 2018. 8

[48] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018. 14

[49] Karttikeya Mangalam and Vinay Uday Prabhu. Do deep neural networks learn shallow learnable examples first? In *ICML Workshop on Deep Phenomena*, 2019. 5

[50] Norman Mu and Justin Gilmer. Mnist-c: A robustness benchmark for computer vision. In *ICML Workshop on Uncertainty and Robustness in Deep Learning*, 2019. 7

[51] NHTSA. Technical report, U.S. Department of Transportation, National Highway Traffic, Tesla Crash Preliminary Evaluation Report Safety Administration. *PE 16-007*, Jan 2017. 1

[52] Luke Oakden-Rayner, Jared Dunnmon, Gustavo Carneiro, and Christopher Ré. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. In *ACM conference on Health, Inference, and Learning*, 2020. 1

[53] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. Deep learning for financial applications: A survey. In *Applied Soft Computing*, 2020. 1

[54] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of mathematical statistics*, 1962. 2

[55] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training, 2021. 8

[56] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901. 8

[57] M Rosenblatt. Remarks on some nonparametric estimates of a density function. annals of mathematical statistics. 1956. 8

[58] Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*, volume 589. John wiley & sons, 2005. 8

[59] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. In *Proceedings of the IEEE*, 2021. 7

[60] Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. *arXiv preprint arXiv:1906.02694*, 2019. 8

[61] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. In *IJCV*, 2015. 1, 3

[62] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017. 8

[63] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *ICML*, 2017. 2

[64] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at ICLR*, 2014. 1, 2

[65] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. In *ICML Workshop on Visualization for Deep Learning*, 2017. 2

[66] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, 2017. 2

[67] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004. 8

[68] Michael Veale and Reuben Binns. Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data. In *Big Data & Society*, 2017. 1, 4

[69] Bernard L Welch. The generalization of 'student's' problem when several different population variances are involved. In *Biometrika*, 1947. 6

[70] Hongtao Xie, Dongbao Yang, Nannan Sun, Zhineng Chen, and Yongdong Zhang. Automated pulmonary nodule detection in ct images using deep convolutional neural networks. In *Pattern Recognition*, 2019. 1

[71] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 14

[72] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017. 6

[73] Jianping Zhang. Selecting typical instances in instance-based learning. In *Machine Learning Proceedings*. Elsevier, 1992. 1, 8