Leveraging Gradient Information for Out-of-Domain Performance Estimations

Ekaterina Khramtsova[1](✉), Mahsa Baktashmotlagh[1], Guido Zuccon[1], Xi Wang[2], and Mathieu Salzmann[3]

[1] The University of Queensland, Brisbane, Australia
{e.khramtsova, m.baktashmotlagh, g.zuccon}@uq.edu.au
[2] Neusoft, Shenyang, China wxi@neusoft.com
[3] École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
mathieu.salzmann@epfl.ch

Abstract. One of the limitations of applying machine learning methods in real-world scenarios is the existence of a domain shift between the source (i.e., training) and target (i.e., test) datasets, which typically entails a significant performance drop. This is further complicated by the lack of annotated data in the target domain, making it impossible to quantitatively assess the model performance. As such, there is a pressing need for methods able to estimate a model's performance on unlabeled target data. Most of the existing approaches addressing this train a linear performance predictor, taking as input either an activation-based or a performance-based metric. As we will show, however, the accuracy of such predictors strongly depends on the domain shift. Recent research highlights the significance of network weights in understanding model generalizability. The early work of [46] proposes a method to predict out-of-distribution error by comparing the weights of the original model and fine-tuned model on the target data. However, this process is computationally demanding, especially for large models and input sizes. To address this, we propose an an efficient approach for assessing model's performance on target dataset by leveraging the gradients and Hessian of model as indicators of weight differences. Our approach builds on the idea that lower norms of gradient and Hessian matrices signifies a flatter training landscape and better adaptability to new data. Our extensive experiments on standard object recognition benchmarks, using diverse network architectures, demonstrate the benefits of our method, outperforming both activation-based and performance-based baselines by a large margin. It also outperforms [46]'s weight-based approach in efficiency by avoiding parameter updates and effectively estimates out-of-domain performance. Our code is available in the following repository: https://github.com/khramtsova/hessian_performance_estimator/

Keywords: Performance Prediction · Generalisability Estimation.

1 Introduction

Being able to estimate how well a trained deep network would generalize to new target, unlabeled datasets would be a key asset in many real-world scenarios,

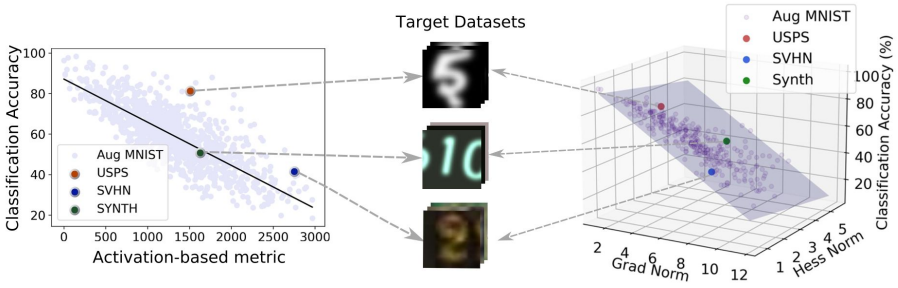

Fig. 1: Correlation between classification accuracy and different metrics: Hessian and Gradient norms (right, our method) and Fréchet Inseption Distance (left, [13]). Note that our method yields a more reliable performance estimator, as evidenced by the points corresponding to the target datasets lying closer to the decision boundary. The light-blue points correspond to sample sets from the meta-dataset.

where acquiring labels is too expensive or unfeasible. When the training and target data follow the same distribution, this can easily be achieved by setting aside a validation set from the training data. However, such a performance estimator fails in the presence of a domain shift, i.e., when the target data differs significantly from the source one.

Recent studies [13,11] address this by creating a meta-dataset incorporating multiple variations of the source data obtained by diverse augmentation techniques, such as background change, color variation, and geometric transformations, so as to mimic different domain shifts. Target datasets can then be sampled from this meta-dataset, and their ground-truth performance obtained by evaluating the source-trained network on them. In essence, this provides data to train a linear performance predictor, which in turn can be applied to the real target data.

The aforementioned studies differ in the quantities they use as input to this linear performance predictor. Specifically, Deng et al. [13] rely on the Fréchet distance between the network activations obtained from the source samples and the target ones, whereas the authors of [11] exploit the performance of the source network on the task of rotation prediction. Unfortunately, while the resulting linear predictors perform well within the meta-dataset, their generalization to some real target datasets remains unsatisfactory, depending on the gap between the source and real target data. This is illustrated by the left plot of Fig.1, where the red point indicating the true performance on USPS lies far from the activation-based linear predictor shown as a black line.

Recent studies show that the network weights provide valuable insights into model uncertainty [31], model complexity [40], model compressibility [2], and in-domain generalization [4,16,35,43]. The early work of [46] in the field of out-of-domain generalization analyzes the extent to which network weights change when

fine-tuned on target data with a supervised loss. It confirms that the greater the domain gap between source and target datasets, the more substantial the changes needed in the network to bridge this gap. Building on this concept, [46] proposes predicting out-of-distribution error by fine-tuning the model on the target data using a cross-entropy loss and then calculating the distance between the weights of the original and the fine-tuned models. Despite the more accurate estimation achieved with the proposed weight-based metric in [46], the process of fine-tuning and comparing weights can be computationally intensive and time-consuming, particularly for large models.

To take advantage of the weight-based metric while avoiding its high computational cost, we propose an efficient alternative that captures the key aspects of weight differences without needing to update network parameters. Our method focuses on examining the gradients and the Hessian of the network, acting as a substitute for weight changes. Our approach is based on the intuition that a smaller norm of the gradient and Hessian matrices indicates a flatter landscape of the training objective. This flatness is strongly correlated with better generalization, suggesting that networks with flatter landscapes are more likely to generalize effectively to new data. Our approach avoids the significant complexity, memory, and time demands usually involved in updating network weights. This makes it especially suitable for large models and large batch sizes.

Our results demonstrate that the proposed metric provides more reliable performance estimates than those based on activation and score. Compared to the weight-based approach of [46], our method is not only more efficient because it does not require parameter updates, but it is also effective in bridging the domain gap for estimating out-of-domain performance. This is illustrated in the right plot of Fig. 1, where the points corresponding to the three real target datasets all lie close to the linear predictor. While alternative, more complex measures may also be viable, our work shows that even a basic gradient-based approach surpasses other methods, which we evidence on several benchmark datasets and using different network architectures.

2 Related Work

Various methods have been proposed to estimate the performance of a model on an unlabeled dataset under a domain shift. We categorize the existing works into two main groups: Activation-based and performance-based methods.

Activation-based approaches aim to find a criteria for performance estimation based on network activations. For example, Garg et al. [18] propose Average Threshold Confidence (ATC) score based on the negative entropy of networks predictions. The authors acknowledge that ATC returns inconsistent estimates on certain types of distribution shifts. Another approach in this category [41] explores various statistics derived from a prediction score to estimate the accuracy on the target domain. Elsahar et al. [15] also provide a similar analysis for NLP tasks. An alternative entropy-based approach was proposed by Guillory et al. [19], who discover a correlation between the classification accuracy and the

difference of entropy between the network activations on the source and target data. However, the success of this approach to produce consistent estimations depends on how calibrated the network is.

Chen et al. [6] develop an evaluation framework, Mandoline, that adapts importance weighting to settings with distribution shifts between source and target domains. Their approach leverages prior knowledge about the nature of the shift, which can be a strength when such information is available, though it may limit applicability in fully unsupervised scenarios.

In contrast with the above-mentioned approaches that focus on the network output, Deng et al. [13] analyze feature representations. The authors create augmented source datasets and train a linear regression model to predict accuracy based on the Fréchet distance between source and augmented feature representations. In our experiments, while there is a strong linear correlation between accuracy on the augmented datasets and the Fréchet distance, real target datasets do not consistently follow this pattern, leading to poor accuracy estimates.

Performance-based approaches evaluate the classification accuracy of the network using its performance on self-supervised tasks. For instance, Deng et al. [11] propose to learn a correlation between the rotation prediction accuracy and the classification accuracy. The works of [26,8] show that test error can be estimated by training the same network multiple times on the source dataset and measuring the disagreement on the target dataset. Building on this work, Chen et al. [5] learn an ensemble of models to identify misclassified points from the target dataset based on the disagreement between the models, and use self-training to improve this ensemble.

The aforementioned methods require access to the model during training. For example, in the work of Deng et al. [11], the network architecture needs to be upgraded with the second head and trained on both tasks. The works of [26,8,5] require re-training of the source model to find the samples with disagreement. This might be undesirable for a large source dataset where training is time consuming. Note that our approach requires neither architecture alterations nor re-training on the source data.

In this work, we focus on analyzing the network weights and gradients, which was proven to be useful for various in-domain and out-of-domain tasks. For example Nagarajan et al. [35] show that the distance of trained weights from random initialization is implicitly regularized by SGD and has a negative correlation with the proportion of noisy labels in the data. Hu et al. [24] further use the distance of trained weights from random initialization as a regularization method for training with noisy labels. Yu et al. [46] introduce a projection norm and show its correlation with out-of-distribution error.

By contrast, here, we study the relationship between the first and second-order derivative of the network w.r.t an unsupervised loss function, and performance on the target data. Our approach compares favorably to the SOTA accuracy estimation methods. We emphasize that our method requires neither prior knowledge of the nature of the distribution shift, nor target labels.

3 Methodology

Let us now introduce our approach to estimating how well a model trained on a source dataset would generalize to a target dataset from a different domain, in the absence of target supervision. Instead of predicting performance from the activation difference between the source and target samples or from the network performance on a different task, we propose to exploit the model's weights perturbations from an unsupervised loss. Specifically, we consider the Gradient Norm and the Hessian Norm, obtained by differentiating the network with an unsupervised loss function calculated on the target dataset. We empirically show that these metrics display a strong linear correlation with the model performance on the target task. We therefore learn this correlation with a linear regressor trained on augmented versions of the source data, which we use to predict the target data performance.

3.1 Problem Definition

Let $\mathcal{P}^S$ and $\mathcal{Q}^T$ be the probability distributions of the source and target domains, respectively, $\mathcal{D}_S : \{x_s, y_s\}^{n_s} \sim \mathcal{P}^S$ be a labeled source dataset with n_s samples, and $\mathcal{D}_T : \{x_t\}^{n_t} \sim \mathcal{Q}^T$ be an unlabeled target dataset with n_t samples. A model f_θ is trained on the source dataset $\mathcal{D}_S$ to predict a correct label: $f_\theta : x_i \rightarrow \hat{y_i}; x_i \sim \mathcal{D}^S$. Our goal then is to estimate the accuracy of the trained model f_θ on the unlabeled target dataset $\mathcal{D}_T$.

3.2 Gradient-Based Performance Estimation

In this paper, we propose predicting model performance on target data by examining the norm of the first and second-order derivatives of the unsupervised loss function with respect to the target dataset. This is motivated by the intuition that large domain gaps would lead to larger gradient variations and also to lower accuracy than small domain gaps. Below, we first introduce the intuition behind analysing the weight dynamics for estimating the performance on unlabeled target dataset; we then propose an approach for approximating the the degree of the network changes via the gradient. Finally, we show how the proposed approximating can be used to build an effective and efficient accuracy predictor.

Due to the high-dimensionality of the network weight space, comparing the network weights and gradients is non-trivial and may suffer from the curse of dimensionality. The impact of backpropogation is not equally distributed across the network, with the last layers typically being affected more than the first ones [29]. Furthermore, computing the second-order derivative is both resource-intensive and time-consuming, making it less suitable for deep networks. Therefore, we limit our gradient calculations to the classifier part of the network, which includes only the final fully connected layer.

From magnitude of network updates to difference between weights In this section, we provide a detailed analysis of weight updates from the perspective of gradients.

Consider a network with parameters θ optimized by minimizing an unsupervised entropy loss L, using a learning rate α. We will use the following notations:

- $\theta^{(k)}$ - the weights of the network at step k,
- $g_i^{(k)} = \nabla L^{(i)}(\theta^{(k)})$ - the gradient of the loss L at step i w.r.t. $\theta^{(k)}$,
- $H_i^{(k)} = \nabla^2_\theta L^{(i)}(\theta^{(k)})$ - the Hessian at step i w.r.t. $\theta^{(k)}$.

The aim of this section is to express the dynamics of the network updates after k steps of fine-tuning (from $\theta^{(0)}$ to $\theta^{(k)}$) using the gradients and Hessians computed at $\theta^{(0)}$. This will allow us to assess the model's generalizability by calculating the derivatives only once.

Let us consider the change in the weights after one step of gradient descent. Naturally, it corresponds to the gradient at step 0:

$$\theta^{(0)} - \theta^{(1)} = \alpha g_0^{(0)} \tag{1}$$

After the second gradient descent step, the distance between the weights has an additional gradient $g_1^{(1)}$, calculated at step 1 w.r.t. the updated weights $\theta^{(1)}$:

$$\theta^{(0)} - \theta^{(2)} = \alpha g_0^{(0)} + \alpha g_1^{(1)} \tag{2}$$

Following the works of [37] and [42], we can approximate $g_1^{(1)}$ using First-order Taylor series as follows:

$$g_1^{(1)} = g_0^{(1)} - \alpha H_0^{(1)} g_0^{(0)} + \mathcal{O}(\alpha^2). \tag{3}$$

This allows us to approximate the gradients at $\theta^{(1)}$ in terms of the gradients and Hessians at $\theta^{(0)}$.This approximation is due to the fact that we are not updating the network, which makes $\theta^{(1)}$ unknown at step 1. Plugging 3 back into 2:

$$\theta^{(0)} - \theta^{(2)} = \alpha g_0^{(0)} + \alpha(g_0^{(1)} - \alpha H_0^{(1)} g_0^{(0)}) = \alpha(g_0^{(0)} + g_0^{(1)}) - \alpha^2 H_0^{(1)} g_0^{(0)} + \mathcal{O}(\alpha^3) \tag{4}$$

Finally, after k steps of gradient descent, the gradient is:

$$g_k^{(k)} = g_0^{(k)} - \alpha H_0^{(k)} \sum_{i=0}^{k-1} g_i^{(i)} + \mathcal{O}(\alpha^2) \tag{5}$$

The weight difference after k steps of fine-tuning is:

$$\theta^{(0)} - \theta^{(k)} = \alpha \sum_{i=0}^{k-1} g_i^{(i)} = \alpha \sum_{s=0}^{k-1} g_0^{(s)} - \alpha^2 \sum_{i=1}^{k-1} H_0^{(i)} \sum_{j=0}^{i-1} g_0^{(j)} + \mathcal{O}(\alpha^3) \tag{6}$$

The norm of the weight difference $||\theta^{(0)} - \theta^{(k)}||$ has been shown to correlate with model accuracy [46]. However, our approach diverges from this weight-based approximation, opting instead to work directly with gradients. By taking a norm of 6, we obtain:

$$||\theta^{(0)} - \theta^{(k)}|| \leq \alpha||\sum_{s=0}^{k-1} g_0^{(s)}|| + \alpha^2||\sum_{i=1}^{k-1} H_0^{(i)}|| \cdot ||\sum_{j=0}^{i-1} g_0^{(j)}|| \tag{7}$$

In Equation 7, two main components have emerged. We will now explain how these two components of the model updates, namely the magnitude and the curvature of the loss function, are related to model generalizability.

- *Magnitude of the network updates:* The magnitude of the network updates required to optimize an unsupervised loss function is encapsulated in its average gradient w.r.t. $\theta^{(0)}$, i.e., $||\sum_{s=0}^{k-1} g_0^{(s)}||$. The gradient's magnitude reflects the flatness of the loss function and can be regarded as an indicator of convergence [47].
- *Curvature of the loss surface.* In addition to evaluating the magnitude of network updates, we can assess the sensitivity of the loss function to changes in model parameters by examining the Hessian, which provides insights into the curvature of the loss surface. Large Hessian norm, $||\sum_{i=1}^{k-1} H_0^{(i)}||$, implies that slight modifications in the parameters θ might lead to substantial changes in loss, indicating a sharp minimum [27]. On the other hand, a Hessian with small values implies that the parameters reside in a flatter region.

Accuracy predictor. The right-hand side plot in Fig.1, corroborated by our experimental results, reveals a linear relationship between the newly proposed gradient-based metrics and the accuracy achieved on a target dataset. This suggests that the accuracy for a given target dataset can be effectively predicted using a linear regression model. Specifically, we compute both the norm of the cumulative Hessians and the norm of the cumulative gradients across the target dataset. These computed norms serve as inputs to the linear regression model, where w_0, w_1, and w_2 represent the model's learnable parameters.

To train these parameters, we follow [13] and create a meta-dataset consisting of a collection of datasets obtained by performing different augmentations of the source data. Specifically, a sample set $\hat{\mathcal{D}}_s^j$ in the meta-dataset is built as follows. First, a set of m possible transformations $T = \{T_1, T_2, .., T_m\}$, corresponding to background change, geometric transformations, or color variations, is created. Then, l images are randomly selected from the validation set $\{v_s\}$ of the source data, leading to a set $\{v_s^j\}^l \subset \{v_s\}$. A random selection of t transformations $\tau = \{T_i\}_{i=1}^t$ is then applied to these images, resulting in the sample set $\hat{\mathcal{D}}_s^j = \tau[v_s^j]$. By repeating this process k times, we create a collection of sample sets, which form the meta-dataset.

As each sample set originally comes from the source data, we can compute its true performance under model f_θ. Similarly, we can calculate the model's gradients on each sample set using the entropy loss function. Altogether, this

gives us supervised data, consisting of pairs of gradients and true accuracy, from which we can learn the weights w_0, w_1 and w_2 of the linear regressor.

3.3 Accuracy Prediction on Target Data

We can use the trained linear regressor to estimate the network performance on any unlabeled target dataset. Specifically, given a target dataset $\mathcal{D}_T : \{x_t\}^{n_t}$, we first split it into k subsets of size l,

$$\mathcal{D}_t = \{\mathcal{D}_t^1, \mathcal{D}_t^2, ..., \mathcal{D}_t^k\}, \quad k = \left\lfloor \frac{n_t}{l} \right\rfloor,$$

so that the size of each subset matches the size of the validation sample sets. This procedure standardizes gradient scales across varying dataset sizes.

Then, we calculate the gradient g and the Hessian H on $\mathcal{D}_t^j, \forall j \in [1, .., k]$ with our unsupervised loss, and estimate the change using a gradient-based measure. Given the obtained metrics g and H, we use the trained linear regressor to predict the accuracy of $\mathcal{D}_t^j$ as $acc_j = w_2 \cdot g + w_1 \cdot H + w_0$. The final accuracy for the target dataset is calculated as the average accuracy of its subsets.

4 Experiments

We conduct extensive experiments on both Single-Source and Multi- Source Datasets to evaluate the effectiveness of the proposed approach.

Single-Source Datasets include Digits, CIFAR10 and ImageNet.

Digits consists of one source domain, MNIST [33], with 60K training and 10K test images of handwritten digits from 10 classes, and three target datasets: USPS [14], SVHN [36], and SYNTH [17]. The target datasets also consist of digit images, but they differ in terms of colors, styles, and backgrounds. USPS and SVHN represent natural shifts, while SYNTH represents a synthetic shift.

CIFAR10 contains one source domain, CIFAR10 [30], with natural images from 10 classes, divided between 50K training samples and 10K test samples, and one target domain, CIFAR10.1 [38] with 2K test images from the TinyImages dataset [44]. *ImageNet.* The source domain is a large-scale dataset of natural images, Imagenet [10], with 1.2M training and 50K validation images. The target dataset ImageNetV2 [39] contains three test sets with 10K images each, representing a natural domain shift.

Multi-source Datasets For a more realistic evaluation, we include multi-source datasets, fMoW[7], Camelyon17 [1] and iWildCam [3], from the WILDs benchmark [28], and PACS and DomainNet from Domain Bed [21], where we employ leave-one-domain-out cross-validation.

fMoW includes satellite images categorized into 62 classes. The domains are categorized based on the year the image was taken and the geographical region.

Camelyon17 includes patches from 50 whole-slide images of a lymph node section from a patient with potentially metastatic breast cancer. The training set consists of 30 WSIs from 3 hospitals, with an OOD-validation split of 10 WSIs

from another hospital and a test split of 10 WSIs from the last hospital. The task is to predict whether the patch contains a tumor.
PACS consists of 4 domains, each representing a unique visual style; within each domain, there are 7 categories.
DomainNet has 6 domains, each separated into 345 object categories. These domains cover a diverse range of image types, including clipart, real-world photos, sketches, infographics, artistic paintings, and quickdraw drawings.
Networks We utilize the LeNet architecture [32] for the Digits setup. For the CIFAR10 dataset, we follow [46] and fine-tune the pretrained on Image-Net ResNet50 [22] model. For ImageNet, we utilize the ResNet50 model with the same hyperparameters described in [22]. For both the fMoW and Camelyon17 datasets, we adopt the Densenet-121 [25] architecture; for iWildCam we use the ResNet50 architecture; and follow the ERM training procedure from the WILDs benchmark [28]. For Pacs and DomainNet, we use ResNet50 [22] and follow the training procedure from DomainBed benchmark [21].

For multi-source datasets we adopt a standard domain generalization training approach, where training is conducted on all available domains except for the test domain.

4.1 Baselines and Metrics

The considered baselines can be divided into three groups: score-based, activation-based and weight-based.

Score-based methods rely on the validation set of the source data to establish a threshold on a certain metric, and evaluate each sample of the target data w.r.t. that threshold. The score-based methods are: Entropy Score, AC [23], DoC [20], COT [34] and Nuclear Norm [12]. Entropy score considers the prediction to be correct if its entropy is smaller than a certain threshold $\tau \in [0, 1]$. In other words, the prediction $\hat{y}$ is considered to be correct if $H(\hat{y}) \leq \tau * \log(C)$, where C is the number of classes. Average Confidence (AC), proposed by Hendrycks et al. [23], calculates the model's performance by determining the maximum confidence value from softmax probabilities on the target data. DOC [20] calculates the difference in probabilities between the source and target datasets. The final accuracy prediction for the target set is determined by subtracting the difference in confidences from the source accuracy. COT [34] uses the Earth Mover's Distance to measure the dissimilarity between the softmax probability distributions of samples from the source and target domains. Nuclear Norm [12] quantify the dispersity and confidence of the prediction matrix with Nuclear Norm.

Activation-based approaches analyze the hidden representations within the network. We consider the FID baseline [13], wherein the authors propose creating a collection of augmented source datasets. They further learn a linear regression model to predict accuracy on these sets based on the Fréchet distance between the source and augmented feature representations. We have also considered the Negative Dispersion Score [45], which analyzes feature separability. However, the results across all natural shifts were unsatisfactory; therefore, we have included it only for the CIFAR dataset in Figure 2.

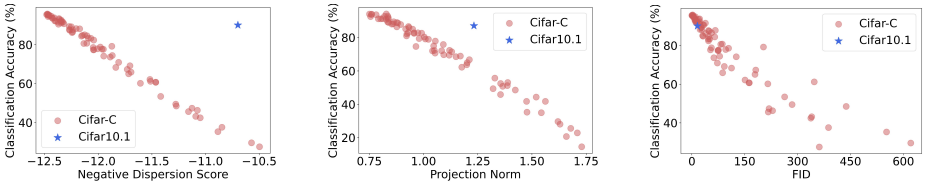

Fig. 2: Correlation between the accuracy of CIFAR-Corrupted and CIFAR10.1 across various metrics, including Negative Dispersion Score, Projection Norm, and FID. Note that while the Dispersion Score and Proj. Norm exhibit a stronger linear correlation on CIFAR-Corrupted compared to FID, they are less effective on CIFAR10.1, which represents natural shift.

Weight-based approaches rely on analyzing the dynamics of the network's weights. We refer to Projection Norm [46] as a weight-based baseline; this method involves fine-tuning the network using pseudo-labels and calculating the distance between the original and fine-tuned weights. The original implementation of Projection Norm does not support direct prediction of accuracy but instead provides a correlation between the norm of weights and accuracy. To adapt it to our task, we employ the meta-set pipeline used in [13] and our own method.

The construction of a Meta-set involves creating a collection of augmented sets from the validation split of the corresponding source dataset. For the comparison to the baselines to be fair, we use the same augmentation strategy for all the methods, which results in identical meta-datasets for every experimental setup. The data is augmented once, prior to the network updates.

As MNIST contains grayscale images, we create binary masks from the MNIST samples. We then select a test sample from the COCO dataset, and mine patches to match the size of the binary masks. Finally, we invert the values of the patches in the location of the MNIST binary masks.

For the COCO and CIFAR10 datasets, we use the RandAugment [9] automated augmentation strategy. For each sample set, we randomly select an augmentation magnitude and three transformations from the following pool of transformation types: `cutout`, `auto_contrast`, `contrast`, `brightness`, `equalize`, `sharpness`, `solarize`, `color`, `posterize`, `translate`. Differently from [13], we do not apply a computationally expensive background replacement on the COCO dataset. In fact, we show that even with these simple transformations, our approach is able to capture a variety of domain shifts.

It is worth noting that the applied augmentations primarily induce covariate shift, as they modify the input distribution P(X) without altering the label distribution P(Y|X). This limits our current setup to shifts that do not affect the label semantics (i.e., no concept shift). Incorporating other shift types—particularly concept shift, where the relationship between features and labels changes—could potentially improve robustness and help explain some of the method's observed limitations. We leave the exploration of such shift types for future work.

Table 1: Results on Single-Source Setups. Values Represent Absolute Error, MAE: Mean Absolute Error.

	Digits				ImageNet				CIFAR10
	SVHN	USPS	SYNTH	MAE ↓	MFreq	Thresh	TopIm	MAE ↓	CIFAR10.1
Ground Truth Accuracy	41.59	81.46	50.66		63.13	72.29	77.61		88.65
Entropy Score $_{\tau=0.1}$	39.82	27.85	30.46	32.71	13.16	13.01	14.28	13.48	3.25
Entropy Score $_{\tau=0.3}$	36.84	7.52	15.94	20.10	11.56	8.23	9.03	9.61	10.90
AC	9.41	5.03	13.17	9.2	5.41	0.15	1.61	2.39	2.61
DoC	9.77	4.67	12.82	9.08	4.72	0.54	0.92	2.06	1.94
COT	10.49	2.97	9.88	7.78	2.45	2.89	1.32	2.22	1.32
Nuc	0.08	4.46	20.37	8.3	11.33	4.01	6.34	7.23	4.3
FID	13.94	26.26	1.76	14.04	12.02	2.82	2.38	5.74	8.90
ProjNorm	14.87	33.99	6.83	18.56	13.74	4.65	2.23	6.87	26.66
Hess & Grad Norm	6.1	7.46	0.57	4.71	1.21	0.09	0.96	0.76	0.85

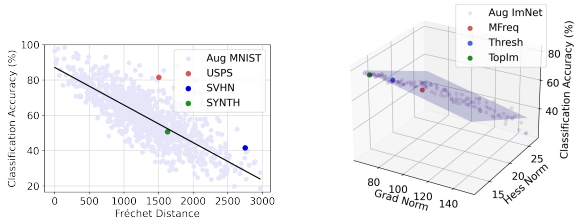

Fig. 3: Single-Source Datasets results. Correlation between classification accuracy and the FID measure for the Digits setup (Left), as well as our proposed gradient-based approach on ImageNet.

Results on Single-Source datasets. The results for Single-Source setup are summarized in Table 1. We start our analysis with the discussion of the criteria for assessing the effectiveness of accuracy prediction. Our findings indicate that relying solely on the correlation between an input metric (for instance, FID or Projection Norm, see Figure 2) and accuracy within a meta-dataset falls short of providing a comprehensive assessment. This limitation arises because this method of evaluation may not accurately represent the correlation with target datasets, particularly when faced with natural distribution shift.

We further highlight that applying a single threshold derived from entropy scores is not effective across various datasets. Specifically, a higher threshold improves the prediction for the Digits dataset, while a lower one is more suitable for CIFAR-10, suggesting a need for adjustments tailored to each dataset.

In single-source setups, we observed that score-based methods, that estimate the threshold based on the validation set of the source data (e.g. DoC, ATC, COT, and Nuc) consistently outperform both activation-based and weight-based methods. Among these, the COT method emerged as the most effective benchmark across all three experimental setups. This superiority is attributed to the

Table 2: Results on PACS and Wilds Benchmark. Values represent Absolute Error, MAE: Mean Absolute Error

	Wilds				PACS				
	Camelyon	IWildCam	FMoW	MAE	Art Painting	Cartoon	Photo	Sketch	MAE
Ground Truth Accuracy	72.91	67.69	52.90		81.45	76.79	93.29	76.48	
Entropy Score $_{\tau=0.1}$	54.52	0.94	25.34	26.94	5.32	0.21	2.81	0.31	2.16
Entropy Score $_{\tau=0.3}$	19.31	21.41	0.45	13.72	5.81	11.18	1.98	10.82	7.45
AC	16.13	2.11	5.69	7.98	7.13	11.8	0.48	7.29	6.68
DOC	14.44	1.41	3.26	6.37	6.68	11.07	0.25	6.61	6.15
COT	2.95	14.83	0.01	5.93	0.73	3.28	24.49	16.63	11.28
Nucl	14.45	7.13	4.05	8.54	9.28	13.06	0.44	3.63	6.60
FID	1.69	8.39	4.89	4.99	1.81	10.15	32.3	16.65	15.22
ProjNorm	16.47	10.95	7.83	11.75	3.02	3.82	28.8	36.48	18.03
Hess & Grad Norm	2.23	3.59	3.66	3.16	5.63	0.42	1.57	4.1	2.93

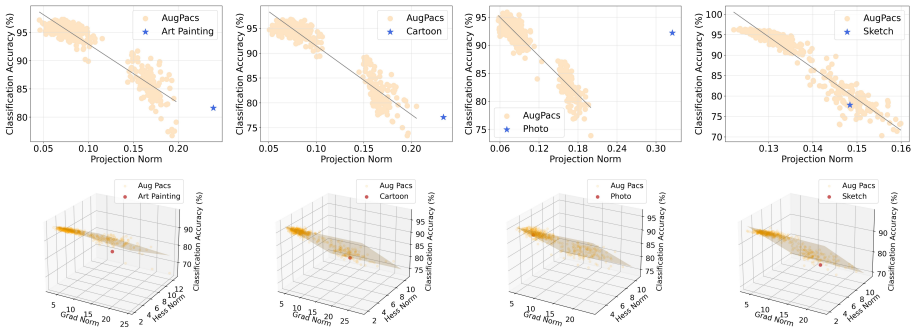

Fig. 4: Multi-Source results. Correlation between the classification accuracy of PACS datasets and the Projection Norm (Top), as well as our proposed gradient-based approach (Bottom).

model's tendency to overfit on the training data, making even the validation set from the same domain an effective estimator for the level of uncertainty.

The final baselines, Projection Norm and Fréchet Inception Distance (FID), exhibited a clear linear correlation with the accuracy of the meta-set; however, there was notable variability in the distribution of points across datasets. For instance, in the Digits setup, predictions of FID for the Synth dataset closely matched the ground truth, with an Absolute Error (AE) of only 1.76. However, for the USPS and SVHN datasets, the deviations from the main trend were substantial, leading to inaccurate predictions.

Finally, our gradient-based approach outperforms all other methods across all three experimental setups. By integrating both Hessian and gradient norms, we successfully developed a Linear Regressor capable of capturing the correlation with classification accuracy not only within the meta-dataset but also across the target datasets. The superiority of our method is demonstrated by the results presented in Table 1, which indicate that our method's predictions

Table 3: Results on DomainNet accross various domain shifts. Values represent Absolute Error, MAE: Mean Absolute Error.

	Clipart	Infographics	Paintings	QuickDraw	Real	Sketches	MAE
Ground Truth Accuracy	58.22	19.92	45.80	12.64	58.36	47.98	
AC	3.15	12.88	6.61	24.17	5.96	7.84	10.10
DOC	2.89	12.29	6.17	24.26	5.47	7.28	9.72
COT	4.2	4.85	1.05	51.83	31.99	0.3	15.70
Nucl	7.21	18.54	10.38	19.47	10.6	12.59	13.13
FID	19.19	19.58	13.36	12.64	33.65	8.67	17.84
ProjNorm	18.95	19.92	3.61	12.64	5.93	2.94	10.66
Hess&Grad	1.18	8.39	5.97	9.1	9.33	6.65	6.77

for the target datasets are more accurate than those produced by alternative approaches. Specifically, we observed an average absolute error of only 4.71% for Digits, 0.76% for ImageNet-V2 and 0.85% for CIFAR10.1.

Results on Multi-Source datasets. Next, we proceed to the multi-source setups, where instead of having a single domain available during training, the network is exposed to multiple domains. This generally leads to the development of more robust models than single-source setups. The results for datasets from Wilds benchmark and PACS dataset are detailed in Table 2, the results for DomainNet are shown in Table 3.

Our analysis reveals that the COT method is unstable across different datasets. For example, it achieves nearly perfect predictions on the FMoW dataset, with an AE of 0.01, as well as on the Sketch target set from DomainNet. It also performs well on the Art Painting subset of the PACS dataset, with an AE of less than 1%. However, its performance declines on the iWildCam dataset, where the AE reaches 14.83%. On the other hand, the FID metric represents a more robust baseline, particularly for datasets within the Wilds benchmark. Notably, it outperforms all other baselines in the evaluation of the Camelyon dataset.

The performance of the Projection Norm in multi-source setups is unsatisfactory. This is attributed to the fact that the reference model is not consistently capable of converging to a local minimum. As illustrated in Figure 4, Top, when the Photo test set is used in the PACS setup, the network fine-tuned on the test data continues to change even after the meta-set weights have stabilized. This leads to a larger Projection Norm for the target dataset compared to the meta-set, resulting in poor performance estimation.

Crucially, our proposed method outperforms existing approaches, achieving the lowest MAE across all evaluated multi-source datasets. The closest competitor to our method is the Projection Norm. However, in addition to inferior performance, the Projection Norm also demands greater computational complexity due to the requirement of fine-tuning the entire network (see Table 4).

Table 4: Comparison of Time Complexity. The values indicate the time (in seconds) required to compute a Projection Norm and our proposed Hess&Grad on a set of 1000 samples, evaluated using a single A100 GPU.

	ImageNet	PACS	IWildCam	FMoW	DomainNet
Proj Norm	100.1	47.5	235.7	114.2	48.64
Our Approach	2.6	2.1	4.8	3.9	2.9

5 Conclusion

In this work, we tackle the problem of predicting the performance of a network on unlabeled target data whose distribution differs from that of the source training data. To this end, we build on the findings of recent work [46] that estimates the performance of the network from the degree of weight changes incurred by fine-tuning the network on the target dataset with a self-supervised loss. In contrast, our method avoids the time-consuming process of updating network weights and relies on analyzing gradients and the Hessian matrix to capture weight differences efficiently. Our extensive experiments show that our approach effectively and efficiently predicts the accuracy across a variety of domain shifts and network architectures.

Acknowledgments. This work is partially supported by Australian Research Council Project FT230100426.

References

1. Bándi, P., Geessink, O.G.F., Manson, Q.F., van Dijk, M.C., Balkenhol, M.C.A., Hermsen, M., Bejnordi, B.E., Lee, B., Paeng, K., Zhong, A., Li, Q., Zanjani, F.G., Zinger, S., Fukuta, K., Komura, D., Ovtcharov, V., Cheng, S., Zeng, S., Thagaard, J., Dahl, A.B., Lin, H., Chen, H., Jacobsson, L., Hedlund, M., Çetin, M., Halici, E., Jackson, H., Chen, R., Both, F., Franke, J.K., Küsters-Vandevelde, H.V., Vreuls, W., Bult, P., van Ginneken, B., van der Laak, J.A., Litjens, G.J.S.: From detection of individual metastases to classification of lymph node status at the patient level: The camelyon17 challenge. IEEE Transactions on Medical Imaging **38**, 550–560 (2019)
2. Barsbey, M., Sefidgaran, M., Erdogdu, M.A., Richard, G., Simsekli, U.: Heavy tails in sgd and compressibility of overparametrized neural networks. In: NeurIPS (2021)
3. Beery, S., Agarwal, A., Cole, E., Birodkar, V.: The iwildcam 2021 competition dataset. arXiv preprint arXiv:2105.03494 (2021)
4. Birdal, T., Lou, A., Guibas, L.J., cSimcsekli, U.: Intrinsic dimension, persistent homology and generalization in neural networks. In: NeurIPS (2021)
5. Chen, J., Liu, F., Avci, B., Wu, X., Liang, Y., Jha, S.: Detecting errors and estimating accuracy on unlabeled data with self-training ensembles. In: NeurIPS (2021)

6. Chen, M., Goel, K., Sohoni, N., Poms, F., Fatahalian, K., Re, C.: Mandoline: Model evaluation under distribution shift. International Conference of Machine Learning (ICML) (2021)
7. Christie, G., Fendley, N., Wilson, J., Mukherjee, R.: Functional map of the world. In: CVPR. pp. 6172–6180 (2018)
8. Chuang, C.Y., Torralba, A., Jegelka, S.: Estimating generalization under distribution shifts via domain-invariant representations. ICML (2020)
9. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.: Randaugment: Practical automated data augmentation with a reduced search space. In: NeurIPS. vol. 33, pp. 18613–18624 (2020)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009)
11. Deng, W., Gould, S., Zheng, L.: What does rotation prediction tell us about classifier accuracy under varying testing environments? In: ICML (2021)
12. Deng, W., Suh, Y., Gould, S., Zheng, L.: Confidence and dispersity speak: Characterising prediction matrix for unsupervised accuracy estimation. In: ICML (2023)
13. Deng, W., Zheng, L.: Are labels always necessary for classifier accuracy evaluation? In: CVPR (2021)
14. Denker, J., Gardner, W., Graf, H., Henderson, D., Howard, R., Hubbard, W., Jackel, L.D., Baird, H., Guyon, I.: Neural network recognizer for hand-written zip code digits. In: NeurIPS. vol. 1 (1989)
15. Elsahar, H., Gallé, M.: To annotate or not? predicting performance drop under domain shift. In: EMNLP-IJCNLP. pp. 2163–2173 (2019)
16. Franchi, G., Yu, X., Bursuc, A., Aldea, E., Dubuisson, S., Filliat, D.: Latent discriminant deterministic uncertainty. ArXiv **abs/2207.10130** (2022)
17. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference of Machine Learning (2015)
18. Garg, S., Balakrishnan, S., Lipton, Z.C., Neyshabur, B., Sedghi, H.: Leveraging unlabeled data to predict out-of-distribution performance. In: ICLR (2022)
19. Guillory, D., Shankar, V., Ebrahimi, S., Darrell, T., Schmidt, L.: Predicting with confidence on unseen distributions. ICCV pp. 1114–1124 (2021)
20. Guillory, D., Shankar, V., Ebrahimi, S., Darrell, T., Schmidt, L.: Predicting with confidence on unseen distributions. In: ICCV. pp. 1114–1124 (2021)
21. Gulrajani, I., Lopez-Paz, D.: In search of lost domain generalization. In: ICLR (2021)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
23. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. ArXiv **abs/1610.02136** (2016)
24. Hu, W., Li, Z., Yu, D.: Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In: ICLR (2020)
25. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. pp. 2261–2269 (2017)
26. Jiang, Y., Nagarajan, V., Baek, C., Kolter, J.Z.: Assessing generalization of sgd via disagreement. ArXiv **abs/2106.13799** (2022)
27. Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P.: On large-batch training for deep learning: Generalization gap and sharp minima. In: ICLR (2017)
28. Koh, P.W., Sagawa, S., Marklund, H., Xie, S.M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R.L., Gao, I., Lee, T., David, E., Stavness, I., Guo,

W., Earnshaw, B., Haque, I., Beery, S.M., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., Liang, P.: Wilds: A benchmark of in-the-wild distribution shifts. In: ICML. vol. 139, pp. 5637–5664 (2021)
29. Kornblith, S., Chen, T., Lee, H., Norouzi, M.: Why do better loss functions lead to less transferable features? In: NeurIPS (2020)
30. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009)
31. Lacombe, T., Ike, Y., Umeda, Y.: Topological uncertainty: Monitoring trained neural networks through persistence of activation graphs. In: IJCAI (2021)
32. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
33. LeCun, Y., Cortes, C., Burges, C.: Mnist handwritten digit database. ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist **2** (2010)
34. Lu, Y., Wang, Z., Zhai, R., Kolouri, S., Campbell, J., Sycara, K.P.: Predicting out-of-distribution error with confidence optimal transport. In: ICLR Workshop (2023)
35. Nagarajan, V., Kolter, J.Z.: Generalization in deep networks: The role of distance from initialization. ArXiv **abs/1901.01672** (2019)
36. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning (2011)
37. Nichol, A., Achiam, J., Schulman, J.: On first-order meta-learning algorithms. ArXiv **abs/1803.02999** (2018)
38. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do cifar-10 classifiers generalize to cifar-10? (2018)
39. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do imagenet classifiers generalize to imagenet? In: ICML (2019)
40. Rieck, B., Togninalli, M., Bock, C., Moor, M., Horn, M., Gumbsch, T., Borgwardt, K.: Neural persistence: A complexity measure for deep neural networks using algebraic topology. In: ICLR (2019)
41. Schelter, S., Rukat, T., Biessmann, F.: Learning to validate the predictions of black box classifiers on unseen data. In: ACM SIGMOD International Conference on Management of Data. p. 1289–1299. SIGMOD (2020)
42. Shi, Y., Seely, J., Torr, P., N, S., Hannun, A., Usunier, N., Synnaeve, G.: Gradient matching for domain generalization. In: ICLR (2022)
43. Simsekli, U., Sener, O., Deligiannidis, G., Erdogdu, M.A.: Hausdorff dimension, heavy tails, and generalization in neural networks. Journal of Statistical Mechanics: Theory and Experiment (2020)
44. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(11), 1958–1970 (2008)
45. XIE, R., Wei, H., Feng, L., Cao, Y., An, B.: On the importance of feature separability in predicting out-of-distribution error. In: NeurIPS (2023)
46. Yu, Y., Yang, Z., Wei, A., Ma, Y., Steinhardt, J.: Predicting out-of-distribution error with the projection norm. In: International Conference on Machine Learning (ICML). vol. 162, pp. 25721–25746 (2022)
47. Zhang, X., Xu, R., Yu, H., Zou, H., Cui, P.: Gradient norm aware minimization seeks first-order flatness and improves generalization. In: CVPR. pp. 20247–20257 (2023)