Proceedings of Machine Learning Research 133:170–190, 2021 NeurIPS 2020 Competition and Demonstration Track

Methods and Analysis of The First Competition in Predicting Generalization of Deep Learning

Yiding Jiang[*] YIDINGJI@ANDREW.CMU.EDU
Machine Learning Department, Carnegie Mellon University
Parth Natekar[†] PATNAT26@GMAIL.COM
Manik Sharma[†] SHMAKN99@GMAIL.COM
Department of Engineering Design, Indian Institute of Technology, Madras
Sumukh K Aithal [†] SUMUKHAITHAL6@GMAIL.COM
Dhruva Kashyap[†] DHRUVA12KASHYAP@GMAIL.COM
Natarajan Subramanyam[†] NATARAJAN@PES.EDU
Department of Computer Science, PES University
Carlos Lassance[†‡] CARLOS.LASSANCE@NAVERLABS.COM
Naver Labs Europe
Daniel M. Roy DROY@UTSTAT.TORONTO.EDU
University of Toronto
Gintare Karolina Dziugaite KAROLINA.DZIUGAITE@ELEMENTAI.COM
Element AI
Suriya Gunasekar SURIYAG@MICROSOFT.COM
Microsoft Research
Isabelle Guyon GUYON@CHALEARN.ORG
LISN, CNRS/INRIA, University Paris-Saclay & ChaLearn
Pierre Foret PIERRE.PFORET@GMAIL.COM
Scott Yak SCOTTYAK@GOOGLE.COM
Hossein Mobahi HMOBAHI@GOOGLE.COM
Behnam Neyshabur BNEYSHABUR@GOOGLE.COM
Samy Bengio[§] BENGIO@GMAIL.COM
Google Research

Editors: Hugo Jair Escalante and Katja Hofmann

Abstract

Deep learning has been recently successfully applied to an ever larger number of problems, ranging from pattern recognition to complex decision making. However, several concerns have been raised, including guarantees of good **generalization**, which is of foremost importance. Despite numerous attempts, conventional statistical learning approaches fall short of providing a satisfactory explanation on why deep learning works. In a competition hosted at the Thirty-Fourth Conference on Neural Information Processing Systems (NeurIPS 2020), we invited the community to design robust and general complexity measures that can accurately predict the generalization of models. In this paper, we describe

[†] Members of the top three teams
[*] Partly done while at Google
[‡] Partly done while at IMT Atlantique
[§] Now at Apple

the competition design, the protocols, and the solutions of the top-three teams at the competition in details. In addition, we discuss the outcomes, common failure modes, and potential future directions for the competition.
Keywords: Generalization, Deep Learning, Learning Theory

1. Introduction

Over the past decade, deep learning has indubitably transformed machine learning. For the first time, we have both the computing hardware and the algorithms to extract information from enormous amount of high-dimensional data such as images and text (Krizhevsky et al., 2012). Extensive research brought to fruition powerful neural network architectures that can learn from hundreds of millions images (Sun et al., 2017) and optimization algorithms that can optimize these models (Kingma and Ba, 2014). However, notwithstanding the large body of research, a clear understanding of underlying root causes that control generalization of neural networks is still elusive (Neyshabur et al., 2019; Zhang et al., 2016; Recht et al., 2019). Instead of deriving a generalization bound, a recent empirical study (Jiang* et al., 2020) looked into many popular complexity measures. By a carefully controlled analysis on hyperparameter choices on a large number of models, the study came to surprising findings about which complexity measures worked well and which did not. The observations suggest that conventional methods of evaluating generalization theories may not be sufficient. Since many measures do not formally measure the size of the hypothesis space, we refer to them as **generalization measures**. Studying what generalization measures work well in practice can grant us theoretical insights about why certain models generalize better (which cross-validation cannot), and ultimately help us design better models. Beyond the theoretical interests and usage in model selections, these complexity measures can also be instrumental in Neural Architecture Search (NAS) by alleviating the need of having a validation dataset.

Rigorously evaluating these complexity measures required training a large number of neural networks, computing the complexity measures on them, and analyzing statistics that condition over all variations in hyperparameters. The cumbersome process is painstaking, error-prone, and computationally expensive. Consequently, this procedure is not accessible to members of the wider machine learning community who do not have access to larger compute power. Based on the framework of Jiang* et al. (2020) which compares different generalization by ranking, we introduced the *Predicting Generalization in Deep Learning* (PGDL) competition (Jiang et al., 2020) at NeurIPS 2020, where competition participants can test their generalization measures on a large and unified benchmark without the burden of setting up the complex pipeline and acquiring necessary computational resources. This competition accomplishes several goals:

1. **Providing Standardized Benchmarks for Generalization**. Most generalization bounds are tested on a small number of selected models. Recent works (Jiang* et al., 2020; Dziugaite et al., 2020) find that many generalization bounds are not predictive of the generalization observed in practice. Large-scale competitions (Bennett et al., 2007; Russakovsky et al., 2015) have been instrumental in advancing machine learning research; in a similar spirit, we believe that constructing a large, diverse, and unified benchmark for generalization prediction can help us discover generalization measures that can explain generalization of deep learning in practice.

2. **Making Large-scale Compute Accessible**. Unlike solutions to traditional machine learning competitions, the generalization measures are functions that are evaluated on a trained neural network and its training data. In addition to preparing the data, evaluating every complexity measure on all the models also incurs a non-trivial computational cost, as many generalization measures require inference over the training dataset. As the results, computing one complexity measure over all models in the competition could take as much as ten hours. In the competition, the participants can submit several generalization measures to be evaluated in parallel. This reduces the turnaround time for testing hypothesis and accelerates the research cycles.

3. **Preventing Potential Exploits**. The format helps keep the models in private – if the models and their training data were public, it would be easy to reverse engineer the identity of the actual dataset without the test data. Centralized solution processing prevents such potential exploits.

2. Background

In this section, we will describe the tasks and evaluation metric of the competition. To facilitate exposition, the description will be simplified to convey the high-level goals and the philosophy behind the design choices. The full details of the tasks and metric used can be found in the competition proposal Jiang et al. (2020).

2.1. Tasks

We seek a generalization measure that is *robust* to changes in different model architectures and training data distributions, and can be computed in a short amount of time (compared to training the models). To avoid the confusion between the datasets consisted of trained neural networks and the more conventional datasets consisted of labeled data points, we will refer to a set of neural networks trained on the same dataset as a **task**. Every task contains models from a fixed archetype (e.g. different connectivity patterns) with different tunable hyperparameters (e.g. depth and width). The definition of different tasks allows us to cover a wide range of different models and datasets, while keeping the number of models tractable.

All the 8 tasks we considered in the competition are image recognition tasks with different variations of convolutional neural networks. The tasks combined contain 454 models (largest task contains 96 models and the smallest task contains 32 models). Each task is trained on one of the six following datasets: CIFAR-10 (Krizhevsky et al., 2009), SVHN (Netzer et al., 2011), Fashion MNIST (Xiao et al., 2017), CINIC-10 (Darlow et al., 2018), Oxford Flowers (Nilsback and Zisserman, 2006), and Oxford Pets (Parkhi et al., 2012). We did not provide the same model trained from different initialization since this would increase computational cost, and Jiang* et al. (2020) observed that randomness does not affect the results significantly. To our knowledge, the variation in architecture and dataset is the largest compared to the variations any generalization measures in the literature have been tested on. The competition is split into two stages: *Phase 1* which features 2 tasks, and *Phase 2* which features 4 tasks. The remaining tasks were made available to the participants for developing and debugging solutions. Over the course of the competition,

we observed that many existing generalization measures are not robust over all the tasks and are thus not as reliable, which corroborates our motivation for using diverse tasks for evaluation. Details of the tasks and the time limit for the solutions are described in Section 1.3 of Jiang et al. (2020). Furthermore, we provide several baseline solutions. These solutions range from classical approaches such as the VC-Dimension (Vapnik and Chervonenkis, 2015) to more modern approaches geared towards deep learning such as Neyshabur et al. (2017). The purposes of these baselines are twofold – they show what can be expected from the existing methods, and also demonstrate how the competitors can interact with the competition's API.

2.2. Metric

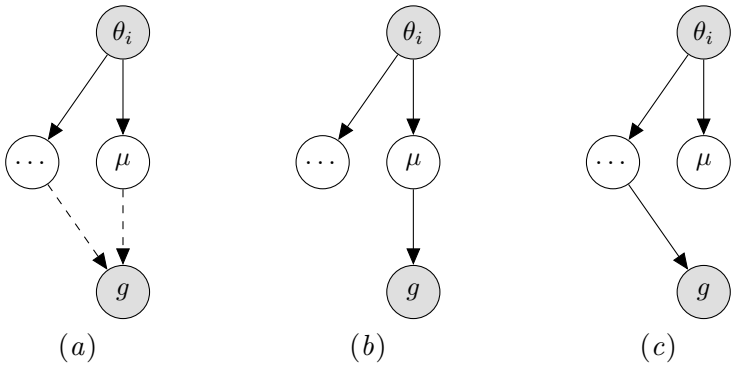


Figure 1: Graphical models of different scenarios. **(a)** Dashed lines represent two possible causal graph structure. **(b)** μ is causal of g so an arrow exist. **(c)** μ is not causal of g and θ_i is their confounder.

We use the same mutual information metric used in Jiang* et al. (2020) to measure a complexity measure's performance on every task, and use the average mutual information metric across all tasks as the final score for a generalization measure. We describe the high-level idea of the evaluation metric here. The full description of the metric can be found in Section 1.5.2 of Jiang et al. (2020).

Given a set of conditioning hyperparameters θ_i (e.g., learning rate and depth of the model), for any ordered pair of models that share the values for parameters in the conditioning set, we can define g as a random variable that indicates whether the first model generalizes better than the second model; likewise, for any generalization measure, we can define μ as a random variable that indicates whether the generalization measures predicts the first model generalizes better than the second model. The mutual information between μ and g conditioned on the hyperparameter sets θ_i represents the amount of inherent dependence between the two random variables. In other words, the mutual information metric represents how well a generalization measure can predict the actual generalization difference between any pair of models.

An effective metric should indicate whether μ contains all the information about g such that knowing the hyperparameters does not provide additional information – this is precisely the notion that conditional mutual information is designed to capture. We compute conditional mutual information for all possible conditioning sets and use the minimum as the final score. This procedure can also be seen as determining whether an edge exists between the generalization measure and the true generalization, in a particular form of causal Bayesian network (Figure 1).

3. Solutions

Throughout the competition, we received a multitude of creative solutions to the challenge. Overall, the solutions can be clustered into three large categories:

1. **Principled complexity measures**: In statistical learning theory, the generalization behavior of a model is often described by PAC bounds of the following form:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\ell(f(\boldsymbol{x}),y)\right] \leq \frac{1}{n}\sum_{i=1}^{n}\ell(f(\boldsymbol{x}_i),y_i) + \sqrt{\frac{\boldsymbol{\mu}}{n}} + \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{n}}\right) \tag{1}$$

   where $(\boldsymbol{x}, y)$ is a pair of labeled data, $\mathcal{D}$ is the true data distribution, f is the model, ℓ is the loss, and μ is a complexity measure which can depend on the model and data. Generalization measures derived from generalization bounds have favorable theoretical properties since they provide an upper-bound on the generalization error (Dziugaite and Roy, 2017; Neyshabur et al., 2017; Bartlett et al., 2017; Neyshabur et al., 2014; Nagarajan and Kolter, 2019).

2. **Data augmentation**: Data augmentation is an important technique for improving generalization performance of machine learning models (Cubuk et al., 2018). Although the exact mechanics by which data augmentation improves generalization are still under active research (Dao et al., 2019), one popular hypothesis is that data augmentation can synthesize more data from the data distribution. Subscribing to this hypothesis, one could estimate the generalization error of a trained model by computing its accuracy on data synthesized by appropriate data augmentation procedures.

3. **Intermediate Representation Analysis**: Deep learning relies on composition of non-linear operations to achieve complex mapping from the input space to the label space. Composition of operations gives rise to intermediate representations between different layers. There is a long line of research on visualizing the intermediate representation of deep neural networks to understand how they perform classification (Olah et al., 2017; Zeiler and Fergus, 2014); recently, several works use the geometry and statistics of the intermediate representation to study generalization (Morcos et al., 2018; Jiang et al., 2018), and demonstrate that these methods have many unusual properties compared to more traditional methods.

Next, we present the detailed solutions of the top three teams. Many participants have written reports that detail their solutions. The reports can be found at: https://sites.

google.com/view/pgdl2020. Each team presents a very unique approach to the problem, which suggests that there may be much more generalization measures waiting to be discovered!

3.1. First Place: *interpex*

The first place solution from Team Interpex borrows on ideas from Neuroscience and uses metrics that capture quality of hidden representation of a model to predict generalization. The final winning solution uses the Davies-Bouldin Index to quantify consistency of representations and Mixup to quantify robustness of representations. This solution can be categorized as an Intermediate Representation Analysis.

3.1.1. Motivation

The ability of the human visual system to create invariant and consistent feature representations irrespective of controlled naturalistic variations of objects is thought to be a major reason for humans' ability to generalize (Karimi-Rouzbahani et al., 2017; Nielsen et al., 2008). One key aspect of invariant object detection is the ability of the visual hierarchy to create easily separable representation manifolds (DiCarlo and Cox, 2007). Robustness is another well-known quality of human vision that helps in invariant object detection. Human vision has been shown to be more robust than deep neural networks on image manipulations like contrast reduction, additive noise or eidolon-distortions (Geirhos et al., 2017).

From a representation learning perspective, the success of deep learning has been attributed, much like humans, to the ability of deep neural networks to create intermediate representations that are consistent, invariant to nuisances in the input, and are well separated (Achille and Soatto, 2018; Cohen et al., 2020). Bengio et al. (2013) discuss, broadly, the properties of good feature representations and discuss how general priors about the world around us can be reflected through these. On a high level, it makes sense that deep models which build "better" feature representations are more likely to generalize better. However, there have been limited studies which explicitly try to understand the relationship between quality of intermediate representations and generalization ability.

Drawing on the parallels between neuroscientific ideas detailed earlier and perspectives from representation learning, three metrics based on the quality of intermediate representations are proposed that have the potential to predict generalization in a post-hoc diagnostic setting like that which the competition provides. These are: (i) Consistency, (ii) Robustness (to controlled variations), and (iii) Separability. The next sections detail the methods used to quantify these properties of intermediate representations and the corresponding results on the competition tasks.

3.1.2. Methods

Notation: We denote a feedforward neural network by $f_w : X \to \mathbb{R}^\kappa$ and its weight parameters by w. The weight tensor of the i^{th} layer of the network is denoted by $\mathcal{W}_i$, so that $w = \text{vec}(\mathcal{W}_1, ..., \mathcal{W}_d)$ where d is the network depth, and vec represents the vectorization operator. $\mathcal{A} = (\mathcal{A}_1, ..., \mathcal{A}_d)$ are the corresponding feature maps of the network at each layer, and $f_{w,k} : \mathcal{A}_k \to \mathbb{R}^\kappa$ is the intermediate model that maps representations at layer k to the network output. Let $\mathcal{D}$ be the data distribution over inputs and outputs. Let $S = \{X, y\}$

be the training set, where $X = \{X_i\}_{i=1}^N$ are the data points and $y = \{y_i\}_{i=1}^N \in \{1, ..., \kappa\}$ are the corresponding training labels sampled randomly from $\mathcal{D}$. We denote by $f_w(X)[j]$ be the j^{th} output of the network. The empirical 0-1 classification loss L is defined as $\hat{L} = \frac{1}{m}\sum_{i=1}^m I\big(f_w(\tilde{X}_i)[y_i] \leq \max_{j \neq \tilde{y}_i} f_w(\tilde{X}_i)[j]\big)$.

Consistency of Representations: To measure consistency of intermediate representations, we use the Davies-Bouldin Index (DB Index), a well known metric to determine clustering quality, which measures the ratio of within-cluster scatter to between cluster separation (Davies and Bouldin, 1979). We compute the DB index of intermediate representations with class labels as cluster indices. This tells us how consistent representations are within a class and how different they are from other classes. Models where representations at an intermediate layer for each class lie in their own cluster and are distinct from other class clusters are likely to have similar representations at subsequent layers, and hence are likely to generalize better. Mathematically, for a given layer L_k and its activations $\mathcal{A}_k$,

$$\tilde{\mathcal{A}}_k = \Phi(\mathcal{A}_k) \qquad \mathcal{S}_i = \Big(\frac{1}{n_i}\sum_{1}^{n_i} |\tilde{\mathcal{A}}_k^i - \mu_{\tilde{\mathcal{A}}_k^i}|^p\Big)^{1/p} \qquad \mathcal{M}_{i,j} = ||\mu_{\tilde{\mathcal{A}}_k^i} - \mu_{\tilde{\mathcal{A}}_k^j}||_p \tag{2}$$

Where, Φ indicates a dimensionality reduction operation, i and j are two different classes, $\mathcal{A}_k^i$ are the feature representations of samples belonging to class i, $\mu_{\tilde{\mathcal{A}}_k^i}$ is the cluster centroid of the representations of class i, S_i is a measure of scatter within representations of class i, and $\mathcal{M}_{i,j}$ is a measure of separation between representations of classes i and j. Then, the complexity measure based on the DB Index for representations at layer k is defined as:

$$\mathcal{C}_k = \frac{1}{\kappa}\sum_{i=1}^{\kappa} \max_{i \neq j} \frac{\mathcal{S}_i + \mathcal{S}_i}{\mathcal{M}_{i,j}} \tag{3}$$

Robustness of Representations: Robustness of intermediate representations can be determined by evaluating model performance on perturbed samples. However, in a post-hoc diagnostic setting that the competition provides, perturbations in the input or representation space need to be data-agnostic, non-stochastic, and controlled (i.e. they should not take the model into a region of the representation manifold that it has never seen before). Mixup (Zhang et al., 2018) and manifold mixup (Verma et al., 2019) provide a potential solution here. Mixup is data agnostic, and perturbs representations in directions that our model has already seen, i.e. towards another training sample. We use a label-wise version of mixup where we check performance on linear combinations of training samples restricted to a class. Mathematically, we create mixed-up representations $\tilde{A}_k$ at layer k for each class i such that,

$$\tilde{\mathcal{A}}_k^i = \lambda \mathcal{A}_{k,1}^i + (1-\lambda)\mathcal{A}_{k,2}^i \tag{4}$$

$$\tilde{y}^i = y^i \tag{5}$$

Where $\mathcal{A}_{k,j}^i$ indicates the representation at layer k for sample j of class i, and inputs X can be denoted by $\mathcal{A}_0$. Computing model performance on mixed-up representations can then tell us about its generalization ability. The mixup complexity measure based on representations at layer k is then defined as:

$$\mathcal{C}_k = \sum_{i=1}^{\kappa} \frac{1}{N_i} \sum_{n=1}^{N_i} I\big(f_{w,k}(\tilde{\mathcal{A}}^i_{k,n})[y_i] \leq \max_{j \neq \tilde{y}_n} f_{w,k}(\tilde{\mathcal{A}}^i_{k,n})[j]\big) = \sum_{i=1}^{\kappa} \hat{L}\big(f_{w,k}(\tilde{\mathcal{A}}^i_k)\big) \tag{6}$$

where $f_{w,k}(\tilde{\mathcal{A}})$ is the model mapping intermediate representations to the output.

Separability of Representations: We use the margin distribution to quantify separability of representations. The margin distribution on perturbed samples is found to be more predictive of generalizing than the vanilla margin. We find simply summarizing the margin distribution with a measure of central tendancy serves our purpose well. Let $D_{(i,j)}$ be the decision boundary between two classes i and j, and X' be a perturbed input sample. Let $\mathcal{A}'_k$ be the corresponding intermediate layer representation. Then, the margin is defined as:

$$D_{(i,j),k} = \{\mathcal{A}'_k \mid f_{w,k}(\mathcal{A}'_k)[i] = f_{w,k}(A'_k)[j]\} \tag{7}$$

And the margin distance is approximated as:

$$d_{f_{w,k},(i,j)}(\mathcal{A}'_k) = \frac{f_{w,k}(\mathcal{A}'_k)[i] - f_{w,k}(\mathcal{A}'_k)[j]}{||\nabla_{\mathcal{A}'_k} f_{w,k}(\mathcal{A}'_k)[i] - \nabla_{\mathcal{A}'_k} f_{w,k}(\mathcal{A}'_k)[j]||_2} \tag{8}$$

The margin complexity measure at layer k is then:

$$\mathcal{C}_k = -\theta(d_{f_{w,k},(i,j)}(\mathcal{A}'_k)) \tag{9}$$

Where θ is a distribution summary measure.

3.1.3. Results and Discussion

As opposed to theoretical measures which give bounds on capacity, simple and intuitive complexity measures are introduced based on quality of intermediate representations that can be predictive of generalization in a post-hoc setting. These are inspired by neuroscientific theories on the ability of the human visual system to create invariant and untangled object representations. Our winning solution involves quantifying both consistency and robustness of intermediate representations by combining the DB Index and Mixup. This gives us an average score of 22.92 on the final set of tasks. Our other two solutions, based on the perturbed margin distribution, achieve scores of 13.93 and 9.29, which rank 1st and 3rd on the private leaderboard respectively. Detailed scores on the final tasks are in Table 1.

Our results indicate that complexity measures based on quality of internal representations may be uniquely qualified to probe generalization ability of models across a wide range of hyperparameters and model architectures. Independent experiments on the public tasks as well as competition results on the private tasks indicate that the proposed measures are more predictive of generalization ability of deep neural networks than complexity measures based on statistical capacity, weight norms, smoothness of the manifold, data augmentation, and other empirical measures. Detailed results and discussions are provided in Natekar and Sharma (2020). An implementation of our solution is available at https://github.com/parthnatekar/pgdl.

Complexity Measure	Task 6	Task 7	Task 8	Task 9	Task Average
DBI * Label-Wise Mixup	43.99	12.59	9.24	25.86	22.92
Augment Margin Summary	8.67	11.97	1.28	15.25	9.29
Mixup Margin Summary	11.46	21.98	1.48	20.78	13.93

Table 1: Mutual Information scores of our final complexity measures on final tasks of PGDL

3.2. Second Place: *Always Generalize*

The second place solution by Team Always Generalize features a generalization measure that aims to quantify the resilience of a classifier to different type of data augmentation. A sophisticated augmentation procedure makes the solution much more robust than other solutions based on data augmentation.

3.2.1. Motivation

It has been shown in Azulay and Weiss (2019) that convolutional neural networks are sensitive to small geometric transformations that are imperceptible to humans. Geirhos et al. (2018) showed that training on augmented input helps improve the performance of the model on the particular augmentations but still the model fails to generalize to unseen augmentations.

The proposed method is based on a simple hypothesis that a model capable of generalizing must be robust to augmentations. The output of the model should not change much when certain augmentations are applied on the input in a way that key features of the input is still retained. In this work, the generalization metric is viewed from the point of corruption robustness of the model.

Recent works Geirhos et al. (2019) have shown that ImageNet trained models are texture biased unlike humans who predominantly classify based on shape. Augmentations that perform changes to the texture would be a suitable test of a model's generalizability.

3.2.2. Proposed Metric

Algorithm 1 describes the proposed metric. For every sample in a randomly sampled subset of the training set, the input is augmented with a collection of augmentations and the class prediction of each output is compared to that of the original image.

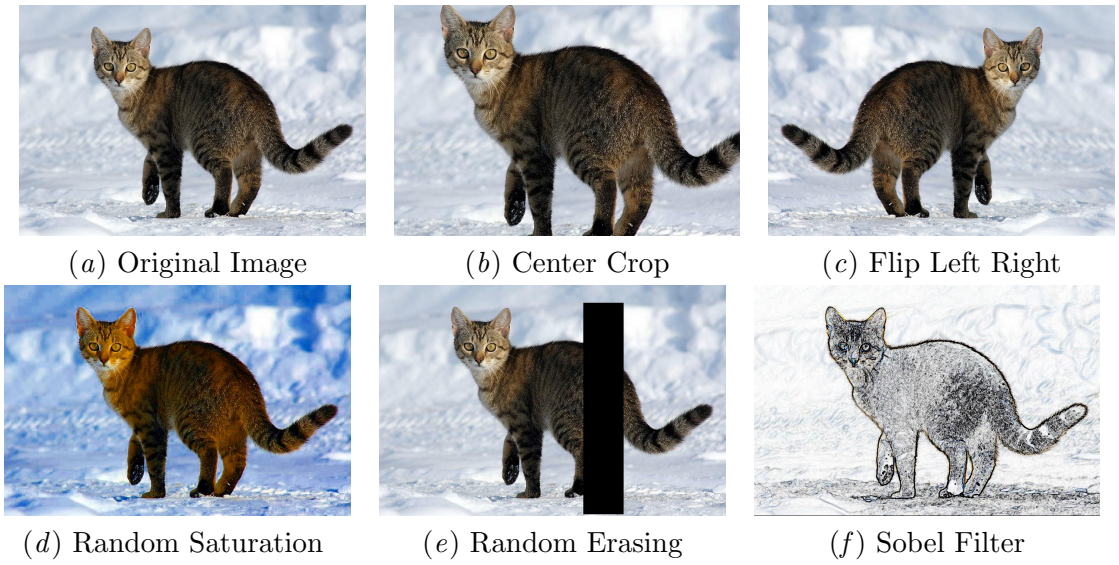
On a successful match between the original and augmented input's predicted label, a penalty equal to the absolute value of the difference between the confidence of the label is added. If there is a mismatch between the prediction of the original and the augmented image's predicted class label, there is a penalty applied to the score based on the strength of the augmentation on which the model misclassified.

The strength is determined by the ability of the augmentation to change the texture of the input. Augmentations that do not alter texture, but alter shape, are considered weak. A model that misclassifies samples on weak augmentations are considered to generalize

Algorithm 1 Measure calculation

Input: Consider a model θ; x is the input; λ is the penalty for an augmentation.
Result: Generalization metric ϕ
$\phi = 0$;
forall $(x, y) \in \mathcal{D}$ **do**
  $x' = \text{Augment}(x)$;
  if $\arg\max_{\hat{y}} P_\theta(\hat{y}|x) = \arg\max_{\hat{y}} P_\theta(\hat{y}|x')$ **then**
    $\phi = \phi - |\max_{\hat{y}} P_\theta(\hat{y}|x) - P_\theta(\hat{y}|x')|$
  else
    $\phi = \phi - \lambda$
  end
end

poorly and are punished for it. The generalization metric reflects the penalty that has been incurred over an augmented subset of the training data. Thus, models that accumulate a large penalty have a higher negative score and tend to generalize poorly. Figure 2 is a visual depiction of some of the augmentations we have used.



(*a*) Original Image (*b*) Center Crop (*c*) Flip Left Right

(*d*) Random Saturation (*e*) Random Erasing (*f*) Sobel Filter

Figure 2: Illustration of experimented augmentations. (Original image cc-by: Von.grzanka)

The list of augmentations used were Flip, Random Saturation, Crop and Resize, Brightness, Random Erasing (Zhong et al.), Sobel filter (Kanopoulos et al., 1988) and Virtual Adversarial Perturbation (Miyato et al., 2018).

Table 2 contains the detailed results of all our submissions. The details of each submission along with their associated penalty terms are described in Appendix B below. We had experimented with individual augmentations and compound augmentation with relative penalties. The penalties for each augmentation were empirically calculated. More

	Task 6	Task 7	Task 8	Task 9	Model Average
Submission 1	**8.38**	6.93	11.37	9.98	9.16
Submission 2	7.97	6.47	11.27	**11.30**	9.25
Submission 3	7.63	**10.16**	**13.79**	9.07	**10.16**
Task Average	7.99	7.85	12.14	10.12	-

Table 2: Phase 2 scores

details on the solution can be found in K et al. (2021) with the code for implementation at : https://github.com/sumukhaithal6/pgdl.

3.3. Third Place: *BrAIn*

The third place solution from Team BrAIn constructs a graph in the hidden representation of the model with different data points as the nodes and the representation similarity between different data points as the entries in the adjacency matrices. A notion of consistency between the labels and the induced graph is used as the final generalization measure. This approach may also be categorized as an intermediate representation analysis.

3.3.1. Motivation

The solution builds upon recent advances of using graphs to represent the latent space of DNNs. Such graphs are called Latent Geometry Graphs (LGGs) (Lassance et al., 2021) and are obtained by connecting samples of a batch depending on how similar their corresponding representations are. In this way, one can represent the latent space by the relations between training set samples and focus on their geometry, instead of the absolute position of each sample.

Label variation is used to derive the complexity measure, which evaluates the separation of classes in the latent space. Indeed, the obtained score indicates how well the latent space is aligned with the classification task to solve and a score of zero can be seen as perfectly separating all (training) samples of distinct classes. Label variation (also called label smoothness) has been demonstrated to correlate well with DNN generalization in (Gripon et al., 2018). Note that such score could be tricked by a DNN that is extremely overfitted (i.e.,the neural network perfectly separates training examples, but performs randomly on test ones). In order to mitigate the overfitting scenario and allow for exploring the interpolation between different samples, we add mixup augmentation (Zhang et al., 2018) to the training examples.

3.3.2. Background

In order to measure label variation, one needs to first construct LGGs. In such a graph, Vertices ($v \in \mathbb{V}$) are data samples and the edge weight between two vertices is a function of the similarity between the intermediate representations of samples at a given layer. In the following paragraphs we detail in three steps how we construct these LGGs:

1. Generate a symmetric matrix $\mathcal{A} \in \mathbb{R}^{|\mathbb{V}|\times|\mathbb{V}|}$ using a similarity measure between intermediate representations, at a given depth ℓ, of a batch of data samples $\mathbf{X}$. For the competition, we used an RBF kernel.

2. Threshold $\mathcal{A}$ so that each vertex is only connected to its k-nearest neighbors and binarize the output. We used only the closest neighbor for the PGDL competition.

3. Normalize $\mathcal{A}$ using its degree diagonal matrix $\mathbf{D}$: $\hat{\mathcal{A}} = \mathbf{D}^{-\frac{1}{2}}\mathcal{A}\mathbf{D}^{-\frac{1}{2}}$.

Now that the LGG from a given layer ℓ is constructed, we measure the aligment between the representations and the classification task. To do this, we opt to use label variation, a measure derived from the framework of Graph Signal Processing (GSP) (Shuman et al., 2013). This measure corresponds to the sum of all edge weights connecting samples of distinct classes. In the case of mixup augmented data a scaling term is added based on the difference in the label space. Let us define the needed concepts of label signal, mixup augmentation and label variation:

Definition 1 (Label signal) *The label signal is defined as the class indicator matrix (one-hot encoding) of each sample* $\mathbf{Y} \in \{0,1\}^{|\mathbb{V}|\times|\mathbb{C}|}$*, where* $\mathbb{C}$ *is the set of class labels.*

Definition 2 (Mixup augmentation)
The mixup *augmentation strategy simply consists in interpolating pairs of examples in the input space* $(\mathbf{X}_i, \mathbf{X}_j)$ *and in the label space* $(\mathbf{Y}_i, \mathbf{Y}_j)$ *using an interpolation factor* $\lambda \sim Beta(\alpha, \alpha)$.

Definition 3 (Label variation) *Consider a LGG with adjacency matrix* $\mathcal{A}$ *and the label signal* $\mathbf{Y}$*. Label variation is defined as:*

$$\sigma_\ell = tr(\mathbf{Y}^\top \mathbf{L}^\ell \mathbf{Y}) = \sum_{i\in\mathbb{V}}\sum_{j\in\mathbb{V}} \mathcal{A}^\ell_{i,j} \sum_{c\in\mathbb{C}} (\mathbf{Y}_{i,c} - \mathbf{Y}_{j,c})^2 \ , \tag{10}$$

where $\mathbf{L}^\ell = \mathbf{D}^\ell - \mathcal{A}^\ell$ *is the combinatorial Laplacian of the LGG from layer* ℓ *and* $\mathbf{D}^\ell$ *its degree matrix.*

Note that a small value of σ indicates that the graph structure is well aligned with the classification task. However, there is a caveat that highly overfitted scenarios may also lead to small values of σ.

3.3.3. Methodology

In order to produce the final scores, $|\mathcal{G}|$ graphs are generated and the median label variation over these graphs is used as the final score. To construct each graph, $\max(1, \frac{500}{C})$ training samples per class are used so that each graph has $|\mathbb{V}| = \max(500, C)$ vertices. A large number of combinations of LGGs, label variation, *mixup* and hyperparameters were tested for the PGDL competition, but for brevity we present only the parameters used in phase 2. More details and code to reproduce the results can be found at: https://github.com/cadurosar/pgdl.

In summary, the label **V**ariation of the **P**enultimate layer (σ_2) with **M**ixup (VPM) was used as the complexity metric for phase 2. We note that the graph formalism could also allow us to represent other commonly used metrics. For example, if we used fully connected graphs without *mixup*, the metric (VP) would be equivalent to the sum of similarities of samples in distinct classes. In the case of our submission, where we use non-symmetric normalized graphs, with $k = 1$, the metric is equivalent to the sum of euclidean label distances between each sample and its closest neighbor in latent space.

3.3.4. Phase 2 results

In the final evaluation (phase 2), competitors could only send their solutions and be informed if they had finished in time or not. Due to these constraints, only one graph was used, which increases the variance of the results reflected by the high variance across different tasks. Nonetheless, the results were pretty consistent with the ones we had on the public tasks and the proposed solution was one of the few solutions whose scores improve from phase 1 to phase 2. In Table 3 the per task results are presented on the final set. As previously noted, the fact that there is a high variability on the results shows that a more in-depth trade-off analysis between computational complexity and performance could have led to improved results. More analysis of the results can be found in Appendix A.

Measure	Mean on final set	Task 6	Task 7	Task 8	Task 9
Third Place - VPM	9.99	13.90	7.56	16.23	2.28

Table 3: VPM results on phase two

4. Results and Observations of the Competition

In this section, we describe the overall outcome of the competition and discuss some important observations we made. In total, more than 200 participants signed for the competition and more than 100 participants made at least one submission. Each team is allowed to submit at most 200 times and the actually maximum number a team made was 197. The participants formed 47 teams[1], of which 35 teams made at least one submission. In total, we received 1105 submissions over the course of the entire competition, which took more than 20000 GPU hours to compute. In phase 2 of the competition, 24 teams submitted 72 solutions and the highest scoring solution is used for determining the final winners. The participation exceeds our original estimation of 50 participants, which suggests that there is potentially a large interest in competitions of this format. One of the most notable observations is how much the performance changed between phase 1 and phase 2 of the competition. In a machine learning or data science competition, besides providing training data, it is common to separate the competition into two phases, each featuring a different datasets. In phase 1, the participants are given a set of unlabeled data and are asked to submit the predicted label. This phase usually lasts for an extended period of time, and the participants are allowed to submit many times and receive feedback (e.g. accuracy) from the submissions. Phase 2 features a different set of unlabeled data and participants are

1. Not every participant is in a team.

Rank	Team	Phase 2 Score	Phase 1 Score	Change	Percent Change
1	interpex	**22.92**	38.73	-15.81	-40.82%
2	Always-Generalize	10.16	**41.79**	-31.63	-75.69%
3	BrAIn	9.99	9.27	+0.72	+7.77%
4	spn	7.99	40.33	-32.34	-80.19%
5	Vashisht	6.51	9.38	-2.87	-30.60%
6	Tuebingen	6.39	6.55	+0.16	+2.44%
7	samiul	5.98	4.21	+1.77	+42.04%
8	smeznar	5.94	1.51	+4.53	+300.0%
9	FZL	5.60	3.75	+1.85	+49.33%
10	IBM-NTUST	4.92	15.21	-10.29	-67.65%

Table 4: Phase 1 score, Phase 2 score and score change of highest ranking 10 teams. The theoretical maximum score is 100.

only allowed to submit a small number of solutions for the final evaluation[2]. The purpose of the setup is to ensure that winning solution does not overfit to the data of phase 1; it is common for performance to drop slightly from phase 1 to phase 2 due to overfitting. However, we observed a significant performance drop in PGDL competition (Table 4). In particular, performance of many solutions with high scores in phase 1 dropped significantly in phase 2; in extreme cases, the performance dropped by more than 70%, suggesting that the solutions have overfit to the phase 1 data severely.

Although overfitting occurs in many different solutions, we found that *data augmentation based approaches* were particularly vulnerable to severe overfitting. This is perhaps not surprising since data augmentation that can simulate new data for datasets in phase 1 (e.g. CINIC 10) is not guaranteed to simulate data accurately for the datasets found in phase 2 (e.g. Fashion-MNIST). This observation confirms the hypothesis that robustness to changes in architecture and datasets is a desideratum for a reliable generalization measure with effective explanatory power of generalization. In addition, it is noteworthy that representation analysis (Natekar and Sharma, 2020; Lassance et al., 2021) may be particularly well-suited for studying generalization in deep learning as many solutions based on representation of the intermediate activation are robust to changes in architectures and data; however, these approaches that leverage representation analysis are currently under-studied and their theoretical properties of these methods are not well understood.

5. Conclusion

Overall, the PGDL competition demonstrated the effectiveness of our evaluation protocol on large scale experiments and that it is hard to measure progress in understanding generalization without a standardized benchmark. We also saw many interesting generalization measures (e.g., intermediate representation analysis) that work well in practice but are not

2. On Kaggle, these two phases are usually called *Public Leaderboard* and *Private Leaderboard*

well understood theoretically; we hope future works would shed light on the theoretical properties of these generalization measures, and ultimately establish theoretical foundation of generalization that accurately reflects the reality.

Moreover, while the competition covered a wide range of architectures and datasets, all of the architectures were sequential models without any skip connections (Srivastava et al., 2015; He et al., 2016). Future versions of the competition should support models of arbitrary computation graphs, which would open up possibility for graph theoretical analysis (You et al., 2020). Another shortcoming of this competition is that all tasks are image recognition tasks. Future iterations should include tasks from other modality such as sentiment analysis with language models or graph classification. Overall, we believe that the PGDL competition has fulfilled our original vision of the competition. We hope that the competition will instigate research in these newly discovered generalization measure, and promote the importance of incorporating rigorous experimental procedures as a standard tool in studying deep learning theory.

Acknowledgments

We would like to thank Google Research for generously providing all the computational resources used in the competition and CodaLab for providing technical support for the competition infrastructure. Codalab is an open-source platform of which a public instance is available at https://competitions.codalab.org/. Finally, we would like to thank all the participants, without whom the competition would not have been possible.

References

Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018.

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations?, 2019.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. Citeseer, 2007.

Uri Cohen, SueYeon Chung, Daniel D Lee, and Haim Sompolinsky. Separability and geometry of object manifolds in deep neural networks. *Nature communications*, 11(1):1–13, 2020.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Tri Dao, Albert Gu, Alexander Ratner, Virginia Smith, Chris De Sa, and Christopher Ré. A kernel theory of modern data augmentation. In *International Conference on Machine Learning*, pages 1528–1537. PMLR, 2019.

Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.

David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

James J DiCarlo and David D Cox. Untangling invariant object recognition. *Trends in cognitive sciences*, 11(8):333–341, 2007.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Gintare Karolina Dziugaite, Alexandre Drouin, Brady Neal, Nitarshan Rajkumar, Ethan Caballero, Linbo Wang, Ioannis Mitliagkas, and Daniel M Roy. In search of robust measures of generalization. *arXiv preprint arXiv:2010.11924*, 2020.

Robert Geirhos, David HJ Janssen, Heiko H Schütt, Jonas Rauber, Matthias Bethge, and Felix A Wichmann. Comparing deep neural networks against humans: object recognition when the signal gets weaker. *arXiv preprint arXiv:1706.06969*, 2017.

Robert Geirhos, Carlos R. Medina Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. Generalisation in humans and deep neural networks. In *NeurIPS*, pages 7549–7561, 2018. URL http://papers.nips.cc/paper/7982-generalisation-in-humans-and-deep-neural-networks.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bygh9j09KX.

Vincent Gripon, Antonio Ortega, and Benjamin Girault. An inside look at deep neural networks using graph signal processing. In *Proceedings of ITA*, February 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. Predicting the generalization gap in deep networks with margin distributions. *arXiv preprint arXiv:1810.00113*, 2018.

Yiding Jiang, Pierre Foret, Scott Yak, Daniel M Roy, Hossein Mobahi, Gintare Karolina Dziugaite, Samy Bengio, Suriya Gunasekar, Isabelle Guyon, and Behnam Neyshabur. Neurips 2020 competition: Predicting generalization in deep learning. *arXiv preprint arXiv:2012.07976*, 2020.

Yiding Jiang*, Behnam Neyshabur*, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgIPJBFvH.

Sumukh Aithal K, Dhruva Kashyap, and Natarajan Subramanyam. Robustness to augmentations as a generalization metric, 2021.

Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2): 358–367, 1988.

Hamid Karimi-Rouzbahani, Nasour Bagheri, and Reza Ebrahimpour. Invariant object recognition is a personalized selection of invariant features in humans, not simply explained by hierarchical feed-forward vision models. *Scientific reports*, 7(1):1–24, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105, 2012.

Carlos Lassance, Vincent Gripon, and Antonio Ortega. Representing deep neural networks latent space geometries with graphs. *Algorithms*, 14(2), 2021. ISSN 1999-4893. doi: 10.3390/a14020039. URL https://www.mdpi.com/1999-4893/14/2/39.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

Ari S Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *arXiv preprint arXiv:1806.05759*, 2018.

Vaishnavh Nagarajan and J Zico Kolter. Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience. *arXiv preprint arXiv:1905.13344*, 2019.

Parth Natekar and Manik Sharma. Representation based complexity measures for predicting generalization in deep learning. *arXiv preprint arXiv:2012.02775*, 2020.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BygfghAcYX.

Kristina J Nielsen, Nikos K Logothetis, and Gregor Rainer. Object features used by humans and monkeys to identify rotated shapes. *Journal of Vision*, 8(2):9–9, 2008.

M-E Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1447–1454. IEEE, 2006.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. https://distill.pub/2017/feature-visualization.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019.

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. *International journal of computer vision*, 115(3):211–252, 2015.

David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pages 11–30. Springer, 2015.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Jiaxuan You, Jure Leskovec, Kaiming He, and Saining Xie. Graph structure of neural networks. In *International Conference on Machine Learning*, pages 10881–10891. PMLR, 2020.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

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

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

Z Zhong, L Zheng, G Kang, S Li, and Y Yang. Random erasing data augmentation. arxiv 2017. *arXiv preprint arXiv:1708.04896*.

Appendix A. Analysis on graph based methods (Third place solution)

In the third place solution, graph-based metrics are used to compute the complexity of the network. While VPM (c.f. Section 3.3.3) was used for phase 2, other graph constructions and metrics were investigated during phase 1. The two most promising methods were:

We detail the other two metrics below:

- **Variation Rate (VR)**: The average rate of change in label variation between the last three layers: $\frac{|\sigma_3-\sigma_2|+|\sigma_2-\sigma_1|}{2}$, where σ_i refers to the i-th layer in the architectures starting from the end. This score comes from the experiments described in Gripon et al. (2018). A handful of neighbors are kept for each sample ($k = 20$) and the graph is symmetrized, but not normalized, so that each sample has the same amount of outgoing and incoming connections, but not the same degree.

- **Worst Case Variation (WCV)**: The maximum value of label variation (σ) over the last 3 layers. Only the closest sample is kept ($k = 1$). The graphs are normalized and symmetrized so that every node has the same degree and all nodes have the same amount of incoming and outgoing connections.

Note that both of these solutions differ vastly from the one we finally used VPM, and that none of these solutions used augmented data. Before the start of the competition, we

were invested in the first solution (VR), as it is the one that most closely mimics the work that motivated our interest in this analysis.

The results for phase 1 are depicted in Table 5. We noticed that reducing the amount of edges gave better results in the public datasets, but we were unable to obtain a reasonable score on the development sets (less than 1). It took our team a while to understand that data augmentation, such as mixup, would be paramount to harder tasks such as the one in the development and final sets. We can see that WCV and VR give better results than VPM on the easier public datasets, but are unable to perform reasonably on the development ones. Another interesting result is that the metric we used for phase 2, VPM ($|G| = 1$), actually performed better in the "final" set (phase 2) than on both datasets of phase 1. Finally, understanding the drawbacks of VR and WCV is an interesting future work that could lead to improvements in the technique.

Measure	Public	Development	Task1 Public	Task 2 Public	Task 4 Dev	Task 5 Dev
VR($\|G\| = 11$)	14.45	0.72	9.31	19.58	0.44	1.00
WCV($\|G\| = 1$)	**32.6**	0.37	**27.74**	**37.44**	0.21	0.55
VPM ($\|G\| = 80$)	11.22	**13.04**	5.61	16.82	**15.42**	**10.66**
VPM ($\|G\| = 1$)	6.26	8.35	6.07	6.44	6.32	10.39

Table 5: Results on the public and development sets (phase 1) for graph-based methods. Note that the solution used for phase two is the one indicated in the last row. $|G|$ refers to the amount of graphs used.

Finally, as we have previously mentioned in Section 3.3, we used only one graph (VPM $|G| = 1$ as we were not sure that the one with more graphs would run in the given time constraints. For completeness we display the results obtained with this more computationally complex solution in Table 6. As expected, increasing the amount of graphs leads to a better mean result and a better score in almost all tasks, but has the drawback of taking more time (as we need to compute more forward passes to generate our graphs).

Measure	Mean on final set	Task 6	Task 7	Task 8	Task 9
VPM ($\|G\| = 80$)	12.27	20.99	8.39	14.77	4.93
Third Place - VPM ($\|G\| = 1$)	9.99	13.90	7.56	16.23	2.28

Table 6: VPM comparison on phase two for 1 and 80 graphs. $|G|$ refers to the amount of graphs used.

Appendix B. Analysis of Augmentation Robustness(Second place Solution)

Table 7 shows the penalties for each of the augmentations and the respective scores on both the public and private datasets. λ indicates the penalty value for the respective augmentation and λ_v indicates the penalty for the virtual adversarial perturbation.

Submission	λ_{flip}	$\lambda_{saturation}$	λ_{crop_resize}	λ_{sobel}	$\lambda_{brightness}$	$\lambda_{flip+saturation}$	λ_{cutout}	λ_{v}	Public Score	Private Score
Submission 1	6	1	3	2	1	12	0	3	33.67	9.16
Submission 2	6	1	2	3	1	9	0	0	40.9	9.25
Submission 3	6	1	2	3	1	12	2	0	**41.8**	**10.16**

Table 7: Penalties for misclassification on each augmentation and scores obtained on 3 submissions

	Task 4	Task 5	Model Average
Submission 1	43.99	23.35	33.67
Submission 2	57.07	**24.72**	40.90
Submission 3	**59.37**	24.19	**41.78**
Task Average	53.48	24.09	-

Table 8: Phase 1 scores

Table 8 contains the results for the first phase of the competition.

The proposed metric performs very well on VGG like models (Task 8) and fully convolutional models (Task 4 and Task 5) compared to other proposed solutions. There is a huge drop in the score from phase 1 to phase 2. It is interesting to note that for Task 6 and task 7, though the models are trained with augmentations from the AutoAugment (Cubuk et al. (2018)) policy, the proposed metric gives a reasonably good score.

In all of the submissions, we only sample 12,800 samples from each dataset. This was done to handle the time constraint of 5 minutes per model defined by the competition organizers.

We had experimented with Neural Style transfer as an augmentation technique. Neural Style transfer can be used to generate images with texture-shape conflict as in Geirhos et al. (2019). This was based on the hypothesis that more generalizable models are more shape biased, similar to humans. Surprisingly, this did not give us a good score. We suspect this may be because most models are texture biased, and most models fail to classify the augmented image correctly.

We also noticed that a simple penalty misclassification term works better than comparing KL divergence of the logits before and after applying the augmentation. We also experimented using the difference of cross entropy of the sample before and after applying the augmentations as the penalty.

Some of the future directions of the proposed method is to develop a standard framework of applying multiple data augmentations. It would also be useful to understand how much the model will improve in terms of generalization capacity when trained with such augmentations.