Achilles-Bench: A Challenging Benchmark for Low-Resource Evaluation

Yudong Wang [*1], **Chang Ma** [*2], **Qingxiu Dong** [1], **Zhifang Sui** [1], **Lingpeng Kong** [2], **Jingjing Xu** [3]
[1] State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University.
[2] The University of Hong Kong [3] ByteDance
{yudongwang, dqx}@stu.pku.edu.cn, {cma, lpk}@cs.hku.hk, {szf, jingjingxu}@pku.edu.cn

Abstract

With promising yet saturated results in high-resource settings, low-resource datasets have gradually become crucial benchmarks (e.g., BigBench Hard, superGLUE) for evaluating the learning ability of advanced neural networks. In this work, we find that there exists a set of "hard examples" in low-resource settings that challenge neural networks but are not well evaluated, which causes over-estimated performance. We first give a theoretical analysis on which factors bring the difficulty of low-resource learning. It then motivates us to propose a challenging benchmark Achilles-Bench to better evaluate the learning ability, which covers 11 datasets, including 8 natural language process (NLP) datasets and 3 computer vision (CV) datasets. Experiments on a wide range of models show that neural networks, even pre-trained language models, have sharp performance drops on our benchmark, demonstrating the effectiveness of evaluating the weaknesses of neural networks. On NLP tasks, we surprisingly find that despite better results on traditional low-resource benchmarks, pre-trained networks, does not show performance improvements on our benchmarks. there is still a large robustness gap between existing models and human-level performance, highlighting the need for robust low-resource learning models. [1]

1 Introduction

Large-scale models have shown strong capabilities in learning from a handful of examples (Scao et al., 2022; Touvron et al., 2023a; OpenAI, 2023), resulting in an increased demand for low-resource benchmarks. Numerous research studies have highlighted the rapid adaptability of such models to new tasks, utilizing techniques like in-context learning (Dong et al., 2022). Consequently, the evaluation of large-scale pre-trained models has shifted towards assessing their ability to quickly learn new downstream tasks with limited available samples, including superGLUE (Wang et al., 2019) and BIG-Bench Hard (Suzgun et al., 2022b).

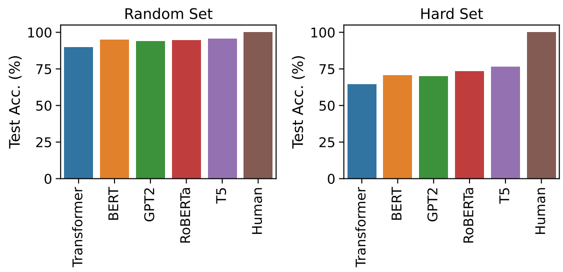


Figure 1: Results on sentiment classification (SST-2). The left figure shows average results on a randomly-sampled set as the test set. The right figure shows average results on a hard set as the test set. The hard test set is selected with smaller loss margins given a weak classifier. Although it is widely-accepted that neural networks can handle sentiment classification well with near-human accuracy (as shown in the left figure), the large drop on hard examples demonstrate that existing models still have generalization issues.

However, many low-resource datasets usually use random or manual selection methods to sample data from the cleaned and balanced training data. They struggle to capture the data biases and increased difficulty commonly encountered in real-world scenarios. Consequently, these benchmarks fall short in evaluating the true learning gap between existing models and human-level models. While some models can surpass human performance on these benchmarks (e.g., SST-2) (Yang et al., 2019; Nangia and Bowman, 2019; He et al., 2021), many studies have revealed that these robust models still face challenges such as spurious correlation (Sagawa et al., 2020; Hu et al., 2023a) or bias (Bolukbasi et al., 2016), which are relatively uncommon in human learning. As depicted in Figure 1, models on a randomly sampled low-resource set demonstrate performance comparable

[*] Equal Contribution
[1] Code and data are available on https://github.com/Qian2333/Achilles-Bench.

Findings of the Association for Computational Linguistics: ACL 2024, pages 2057–2080
August 11-16, 2024 ©2024 Association for Computational Linguistics

to human-level in sentiment analysis. However, their performance significantly deviates from human level when confronted with challenging examples. It motivates us to propose a challenging low-resource benchmark.

In this work, we aim to find challenging examples given any tasks. This approach differs significantly from existing challenging benchmarks, which are either focused on complex tasks such as Big-Bench-Hard (Suzgun et al., 2022a) or specific to extremely few-shot settings like fewGLUE. In contrast to these previous studies (Hu et al., 2024), our proposed benchmark aims to generate difficult examples for any given task[2]. In addition, real-world low-resource data samples often exhibit biases towards specific domains, such as blank backgrounds in image detection or short sentences in handwritten hate speech. Therefore, our evaluation also includes a bias assessment. Specifically, we consider two dimensions: misleading examples with smaller classification margins for performance evaluation, and biased examples for robust evaluation. We begin by conducting a comprehensive analysis of how these two dimensions impact low-resource learning. Based on the insights derived from our analysis, we present an empirical solution to construct a challenging low-resource benchmark. The final benchmark encompasses 3 computer vision datasets and 8 natural language processing datasets.

To prove the effectiveness of the constructed benchmark, we evaluate 13 models, including 8 pre-trained models, such as T5 (Raffel et al., 2020), Llama (Touvron et al., 2023a), etc. All these models struggle to handle our benchmarks, with a large performance gap compared with randomly-sampled low-resource benchmarks. On NLP tasks, we surprisingly find that despite better results on traditional low-resource benchmarks, pre-trained networks, do not show performance improvements on our benchmarks. The contribution of this paper is summarized as: **1)** We propose Achilles-Bench, a challenging benchmark designed to expose Achilles' heel (weaknesses) of neural networks. This benchmark provides a reflective view of the current progress in the field of low-resource learning. **2)** We conduct a comprehensive analysis to identify the factors that particularly exacerbate the difficulty of low-resource learning. **3)** Experimental results demonstrate that our proposed benchmark effectively challenges existing models, including robust pre-trained networks and large language models.

[2]To ensure the exclusion of mislabelled examples, we have implemented a human-check process in our work.

2 Related Work

Low-resource Evaluation Learning on low-resource datasets has recently come into the spotlight with the introduction of more powerful models (Radford et al., 2019; Brown et al., 2020). Recent low-resource benchmarks use a transfer learning setting (Dumoulin et al., 2021; Zheng et al., 2021) as well as in-context learning (Schick and Schütze, 2020; Bragg et al., 2021), and they have also added up on dataset difficulty (Wang et al., 2018). Among these, there are two major types of low-resource benchmark: natural low-resource datasets, and sampled low-resource datasets. The former requires additional dataset curation (Wang et al., 2018; Koh et al., 2021; Srivastava et al., 2022) and currently, most low-resource benchmarks are uniformly sampled from larger datasets (Kolesnikov et al., 2020; Schick and Schütze, 2020; Brown et al., 2020; Logan IV et al., 2021; Alayrac et al., 2022).

Challenging Benchmark Previous approaches in constructing challenging benchmark mainly curate from natural data (Schick and Schütze, 2020; Zheng et al., 2021; Xu et al., 2021; Koh et al., 2021). These methods require heavy annotation and faces misalignment between human-perceived difficulty and samples hard for models. Our methods, however, create an annotation-free framework for building challenging training sets, which has the potential to quickly apply to any available task. Other work involved benchmarking a more comprehensive and challenging list of tasks (Ye et al., 2021; Mukherjee et al., 2021; Hu et al., 2023b), which deviates from our focus in finding model weakness on common tasks.

Data Pruning Our approach is similar to data pruning literature in that we both hope to find a difficult subset in a large dataset. Previously, data pruning methods (Toneva et al., 2018; Hacohen and Weinshall, 2019; Paul et al., 2021; Sorscher et al., 2022; Zhang et al., 2024) use data difficulty metrics including GradNorm and Loss Score to rank and prune datasets. However, we approach dataset sampling from a drastically different goal as we hope to challenge low-resource learning models.

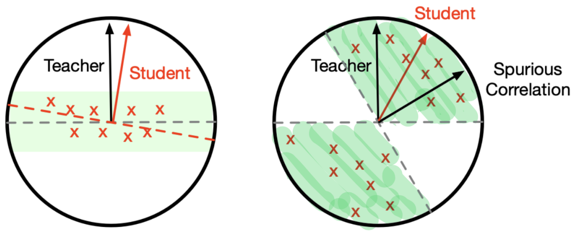


Figure 2: Plot of the perceptron model under hard low-resource learning (left) and biased low-resource learning setting (right). The green area shows the region where few-shot samples are sampled. (a) Under the hard low-resource learning setting, data samples are selected within a small margin to the decision boundary. (b) Under the biased low-resource learning setting, data samples are selected to satisfy the spurious classifier.

3 Understanding the Difficulty of Low-Resource Learning

To better understand the challenges of low resource learning, we first look at the teacher-student setting in learning perceptrons. Consider a large curated dataset of N examples $D = \{x_i, y_i\}_{i\in[N]}$ where $x_i \in \mathbb{R}^d$ are i.i.d. random Gaussian inputs $x_i \sim \mathcal{N}(0, I_d)$, with labels generated by a teacher perceptron $T \in \mathbb{R}^d$ as $y_i = \text{sign}(Tx_i)$. The number of samples $N \to \infty$ but sample per parameter $\alpha = \frac{N}{d} = O(1)$ to remain trainable. Now we consider the low resource scenario where the number of training samples available P is much less than N, where $\alpha_{\text{low}} = \frac{P}{d} \to 0$. For convenience, we sample the data for low resource learning from dataset D such that $D_{\text{low}} = \{x_\mu, y_\mu\}_{\mu\in[P]} \subset D$. Learning on D_{low}, we obtain a new student perceptron J that has generalization error ϵ_g.

Intuitively, three dimensions amount to the difficulty of learning perceptron J: (1) the number of training samples P (here we base the study of data scarity on the sample per parameter variable α_{low}); (2) the classification difficulty of the data samples, denoted by the margin $m = \min_\mu J(x_\mu y_\mu)$; (3) the bias of the training dataset: here we look at a specific type of bias, spurious correlation, which draws correlation based on peripheral attributes of data items with a target variable, denoted as a student perceptron J_{bias}. We explore the difficulty of low-resource learning by altering our selection procedure for D_{low} and explore how ϵ_g changes. Specifically, we look at three settings and use simulation experiments for analysis. 1) Low-resource learning, where D_{low} is uniformly sampled from D. 2) Hard low-resource learning, where the margin of each sample is calculated $m_\mu = T(x_\mu y_\mu)$ and the samples with the smallest margins are selected from D, as shown in Figure 2. 3) Biased low-resource learning, where a biased probe J_{bias} with θ angle to T is chosen as the spurious classifier. Then data that satisfies both $y_i = \text{sign}(J_{\text{bias}} x_i)$ and $y_i = \text{sign}(Tx_i)$ is uniformly sampled from D, as shown in Figure 2.

We elaborate on simulation settings in the Appendix.

Difficult data especially challenges low resource learning. We first compare the setting that increases data difficulty to the random-sampled version of Low-resource Learning. We vary our dataset size from 1% to 500% trainable parameters. As shown in Figure 3, the dark blue line corresponds to the setting where data is uniformly selected, and lighter lines range in data difficulty from margin 0.1 to 1. The functions of ϵ_g to α yield a crossover between the function for random-sampled training data and the one for increased difficulty training data, showing that increased data difficulty affects low resource settings more than sufficient data settings. Also, the increase in generalization error is more distinct for slightly larger training sets. As when the low-resource training set only has a few samples, it requires model to have strong generalization ability to beat the rule of generalization $\epsilon \propto \alpha^{-1}$ and the task is challenging enough.

Low resource learning is more sensitive to spurious correlations. In the biased learning scenario as shown in Figure 4, we compare students trained on biased datasets (red lines) to students trained on random-sampled datasets (blue lines). When the bias probe is more distinct from the teacher (larger θ), the drop in performance is more distinct. This is in line with the phenomenon that when a model overfits on spurious features that contain information distant from semantics, the model tends to suffer on generalization. Also, for smaller bias, low resource learning sees a larger drop in generalization while models with abundant data barely suffer. This show that low-resource learning is sensitive to even small biases.

Theoretical perspective Here we use theoretical analysis in addition to simulations to study the scenario that results in failed generalization in low resource learning. Again, we focus on the scenario where we have a large dataset D that represents the natural task distribution P. We sample a low resource dataset D_{low} from D that form the distribution P_{low}. We theoretically show that the generalization error for the model trained on the low-

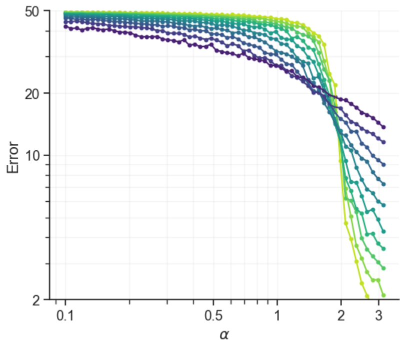
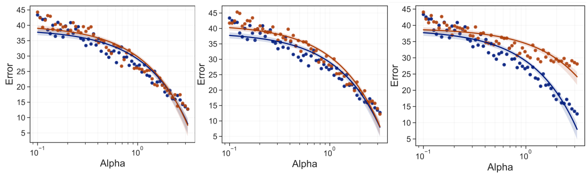


Figure 3: Plot of the generalization error with regard to data difficulty and the number of samples per parameter. Lighter lines represent more difficult data, and the dark blue line represents data uniformly selected.



(a) $\theta = 30^{\circ}$ (b) $\theta = 60^{\circ}$ (c) $\theta = 90^{\circ}$

Figure 4: Plot of generalization error with regard to the number of samples per parameter. Red lines represent biased training set. Blue lines represent unbiased set.

resource dataset is bounded by a function of data difficulty and the distribution bias of low-resource dataset.

Theorem 3.1. *(Low-resource Generalization Measured by Distribution Shift and Data difficulty) Let $\mathcal{H}$ be the hypothesis space $X \rightarrow \mathbb{R}^{\mathrm{d}}$. f_{low} is the empirical risk $\epsilon_{P_{low}}(f)$ minimizer, and f is the hypothesis that minimizes expected risk $\epsilon_Q(f)$, m is the smallest margin of D to decision boundary of f. $MMD(P_{low}, P)$ describes the Maximum Mean Discrepancy (Gretton et al., 2012) between the sampled distribution and the original distribution. Then with probability over 1-δ,*

$$\epsilon_Q(f_{low}) \leq \epsilon_Q(f) + c\sqrt{\frac{|\mathcal{H}|\ln m + \ln\left(\frac{2}{\delta}\right)}{m}} + MMD(P_{low}, P) + \epsilon_\alpha + \epsilon_{\mathcal{H}} \tag{1}$$

where ϵ_α, and ϵ_H are small constants describing the error that occurred in training and the hypothesis space complexity, while c is the constant describing the scale of the effect of margin on generalization. Details are shown in Appendix.

The value of the Equation 1 right-hand side increases when m decreases and the term $\mathrm{MMD}(P_{low}, P)$ increases, corresponding to the increase in data difficulty and the presence of data bias. This theorem applies not only to our simulated scenario of perceptron learning but also to deeper models. In our biased learning setting, the distribution gap between low resource data distribution is larger for biased training set than random-sampled training set, i.e., $\mathrm{MMD}(P^{\theta}_{low}, P) > \mathrm{MMD}(P^{\mathrm{random}}_{low}, P)$, since data samples forming $P^{\mathrm{random}}_{low}$ are sampled uniformly from P.

Based on our simulation experiments and theoretical results in the previous section, we find that low-resource learning is more likely to suffer from performance drop due to data difficulty and dataset bias. However, these scenarios are not covered in previous low-resource benchmarks. This motivates us to propose a challenging benchmark Achilles-Bench for better evaluation.

4 Achilles-Bench Challenge

We propose a new challenging benchmark that elevates low-resource learning difficulty on some well-known datasets. Unlike previous low-resource datasets that are randomly sampled from a training set, we curate the benchmark by selecting one of the most challenging low-resource training sets from GLUE, CIFAR10, CIFAR100, and ImageNet.

Following our theoretical analysis, we introduce the simple yet effective approach to build hard-Bench: First, we train a predictor for only one epoch on a large benchmark, obtaining a biased predictor; then, we score each sample on data difficulty for this stage of training. For each label, we pick the top k samples as our selected low-resource training set. We elaborate on the data difficulty metrics and the biased predictor respectively in section 4.1.

4.1 Metrics Measuring Data Difficulty

Previous literature in curriculum learning (Hacohen and Weinshall, 2019), data pruning (Paul et al., 2021), and continual learning (Toneva et al., 2018) propose metrics for data sample difficulty based on loss or gradient norms. Here we restate three metrics: *Loss score*, *GradNorm score* and explain how they can be applied in our problem scenario.

Loss Score Paul et al. (2021) and Sorscher et al. (2022) state this metric in the EL2N method, which intuitively measure data samples difficulty by looking at whether they can be learned correctly. Data samples with a higher loss score after training are more likely to be near the decision boundary. Therefore, we can select the hardest samples by ranking the loss score on the dataset. We call

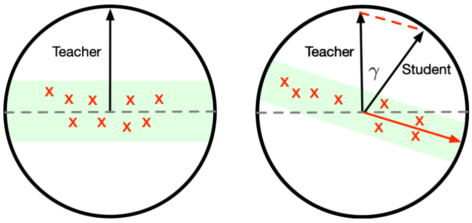


Figure 5: Plot of the perceptron model under both hard and biased low-resource learning setting. Compared to the no-bias setting on the left, , the resulting bias is $\frac{\gamma}{2}$ when the gap between J_{bias} and teacher is γ.

datasets constructed via loss scores as **Achilles-Bench (Loss)**. Examples with higher losses are selected as hard examples.

Gradient Norm Score Paul et al. (2021) discussed using gradient norm as an indicator of data importance. Samples with larger gradient norms shape the training geometry. However, there is little discussion on the connection between gradient norm and data difficulty. Here we give a brief and casual explanation. Based on previous analysis, we can find hard samples by checking their margin to the decision boundary of our model f, $f(x_0) = 0$. Therefore, we can define the L_p norm margin as,

$$m(x) = \min_{x_0} ||x - x_0||_p, s.t. f(x_0) = 0 \qquad (2)$$

We use Taylor's approximation for an approximate solution, following Elsayed et al. (2018).

$$m(x) \approx \frac{|f(x)|}{\|\nabla_x f(x)\|_q}, \qquad (3)$$

When the numerator is constrained (For a classification problem, we can constraint logits $f(x)$ within 1 using sigmoid function), we can maximize the gradient norm to minimize margin. We call datasets constructed via gradient norm scores as **Achilles-Bench (GradNorm)**. Examples with higher gradient norm scores are selected as hard examples.

4.2 Introducing Bias with Early Stopping

As shown in the above sections, we need to train a student predictor to estimate the decision boundary and thereby calculate the data difficulty score. However, we find that we can easily introduce bias into our selected benchmark dataset if we early stop training on the student predictor. We will give an explanation based on the Loss Score.

The Loss Score effectively estimates the difficulty of data examples to be classified correctly when the student predictor is exactly the same as the teacher model, i.e. $\theta = 0$. However, when the student model is undertrained, there would exist a gap γ between student $g(x) = sgn(Jx)$ and teacher $f(x) = sgn(Tx)$. For any x, the loss function would be $L(x) = g(x) - f(x) = (J-T)x$. Therefore, the resulting selected dataset $D_{low} = \{(x_i, y_i)|x_i = \max_x^{i=1,2,...P}(J - T)x, y_i = \text{sgn}(Tx_i)\}$ is isotropic in the nullspace of $J - T$, inducing a bias of $\frac{\gamma}{2}$.

This intuitively explains that we can use an early stopped predictor as well as data difficulty metrics to select a biased and difficult low-resource dataset that mimics the real-world setting. In the following sections, we use this approach to curate our Achilles-Bench.

5 Experiments

5.1 Benchmark Metric

Traditional low-resource benchmarks usually randomly choose a subset from the full-size training data as the training set. In this paper, we also follow this setting and extract hard examples from the full-size data as the training data in our benchmark. To be specific, we implement three benchmarks in this work, which are described as follows. **Random-Bench**. For each label, we randomly select k examples as the training set. We randomly select 3 subsets and report the average results. **Achilles-Bench (Loss)**. For each label, we choose top-k hard examples based on losses scores. **Achilles-Bench (GradNorm)**. For each label, we choose top-k hard examples based on gradient norm scores.

5.2 Benchmark Settings

Our framework is not limited to specific tasks, allowing for flexibility across various tasks. We benchmark on from-scratch models, pre-trained models, as well as large language models. In our implementation, we have chosen 11 tasks to generate a comprehensive and challenging benchmark.

NLP Tasks We choose 8 datasets from GLUE (Wang et al., 2018), a collection of understanding datasets. We select a subset of the full-size training set as a training set. Following previous studies, we use the validation set as the test set considering the hidden test set. For the convenience of the demonstration, we show all the results with accuracy scores. For all NLP datasets, we implement BERT trained with one epoch as a biased predictor to select hard examples. For all NLP datasets, we extract 500 examples for each label (except for WNLI with 100 examples) as

Models	SST-2	COLA	MNLI	QNLI	MRPC	QQP	RTE	WNLI	Average
Random-Bench									
Transformer (Vaswani et al., 2017)	68.16±1.46	69.15±0.04	36.42±0.58	55.45±0.94	68.58±0.39	67.18±0.67	53.65±1.04	56.34±0.00	59.37
BERT (Devlin et al., 2018)	88.68±0.73	79.00±0.59	57.60±1.30	76.02±1.26	77.65±1.38	75.53±0.48	60.58±2.01	**48.45**±3.63	70.44
GPT-2 (Radford et al., 2019)	88.08±0.72	70.35±1.76	58.35±1.65	74.11±2.56	75.93±0.47	76.22±0.86	65.49±2.62	**56.90**±3.40	70.68
RoBERTa (Liu et al., 2019)	**91.54**±0.61	**80.98**±0.56	**75.40**±0.52	**84.47**±0.53	**88.24**±0.27	**80.93**±0.56	**73.00**±1.98	54.93±2.82	**78.69**
T5 (Raffel et al., 2020)	88.73±0.97	78.62±0.58	64.53±2.48	82.56±0.83	74.56±1.71	80.13±0.44	56.46±1.95	52.39±7.74	72.25
Achilles-Bench (GradNorm)									
Transformer (Vaswani et al., 2017)	51.88±0.46	69.15±0.04	35.11±0.67	50.59±0.04	68.38±0.00	62.41±1.06	54.01±0.96	56.34±0.00	55.98
BERT (Devlin et al., 2018)	47.94±2.11	45.77±8.19	33.96±0.47	46.24±2.35	56.08±1.43	52.60±3.01	51.12±0.96	49.30±1.99	47.88
GPT-2 (Radford et al., 2019)	51.44±0.77	51.93±7.92	35.98±1.95	48.62±5.12	65.98±2.33	55.40±4.05	57.76±4.60	56.06±2.25	52.90
RoBERTa (Liu et al., 2019)	51.01±0.65	66.10±6.01	38.42±1.51	48.61±1.50	82.55±1.04	56.69±3.93	60.36±2.88	54.93±2.18	57.33
T5 (Raffel et al., 2020)	52.34±2.35	55.09±6.16	34.27±0.39	48.99±1.51	55.88±3.80	55.72±1.62	48.88±1.54	54.37±4.14	50.69
Achilles-Bench (Loss)									
Transformer (Vaswani et al., 2017)	51.38±0.40	69.11±0.04	34.98±0.69	50.57±0.04	65.64±5.49	48.17±7.69	53.43±0.40	56.34±0.00	53.70
BERT (Devlin et al., 2018)	**45.64**±5.32	**40.92**±4.29	**30.55**±0.88	**40.11**±3.69	38.24±2.52	35.55±2.57	**47.44**±1.22	53.52±3.67	**41.50**
GPT-2 (Radford et al., 2019)	49.79±2.06	56.18±9.92	31.41±1.19	51.01±3.89	50.54±8.02	40.33±5.57	54.73±3.67	55.49±1.44	48.69
RoBERTa (Liu et al., 2019)	50.55±0.62	48.32±11.78	31.66±2.49	41.79±5.62	**38.14**±2.54	**31.74**±2.44	55.09±1.97	55.77±1.91	44.13
T5 (Raffel et al., 2020)	49.86±2.85	55.32±6.06	32.76±0.23	47.15±1.76	53.19±5.12	48.84±5.38	48.45±1.20	53.52±4.45	48.64

Table 1: Results on NLP datasets. Achilles-Bench (Loss) brings higher performance drops than Achilles-Bench (GradNorm). Surprisingly, pre-trained networks does not show better generalization results than randomly-initialized models on our benchmark.

the training set for our main results. Regarding large language models, we adopted the in-context learning paradigm, details can be find in Appendix E. We also build more variants with less training data. More results can be found at Appendix F.

CV Tasks We also explore 3 widely-used image classification datasets, CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 (Krizhevsky et al., 2009), and ILSVRC-2012 ImageNet (Deng et al., 2009) to demonstrate the generality of our approach. For each dataset, we select a subset as the training set in our benchmark, with 500 examples in CIFAR-10, 50 examples in CIFAR-100, 100 examples in ImageNet-1K. results can be found at Appendix F.

5.3 Results

Achilles-Bench challenges neural networks As Table 1, Table 2 and Table 3 illustrate, Achilles-Bench can mislead neural networks with worse generalization errors. We re-implement strong understanding models, which have shown promising results in various low-resource tasks. For example, in Random-Bench, RoBERTa shows the near-human performance on SST-2 with 91% accuracy, which drops sharply on Achilles-Bench with only 51.01% accuracy on Achilles-Bench (GradNorm) and 50.55% accuracy on Achilles-Bench (Loss), nearly random-guessing results. Similar results are observed on CV datasets. For example, DenseNet-121 trained on a random sampling set achieves high test results with 71.33% accuracy on CIFAR-10. The accuracy drops to 59.87% on Achilles-Bench (GradNorm) and to 44.81% on Achilles-Bench (Loss). For LLMs, LLaMA-7B and LLaMA2-7B consistently demonstrate the lowest performance on Achilles-Bench. Regarding BLOOM-1.1B's performance on QQP, it is noteworthy that the model's results are subpar compared to the label distribution, where "not duplicate" constitutes 63.2% of the dataset. The large performance drop also indicates that there is still a large gap between existing models and human-level models. All these drops demonstrate that our benchmark poses a great challenge.

Pre-trained networks show strong generation results on CV benchmarks, but still suffer from handling NLP tasks Compared with randomly-initialized models, pre-trained networks show better generalization results in CV datasets, as shown in Table 3. For example, ViT-B/16 does not yield obvious performance drops on Achilles-Bench. As a comparison, pre-trained networks have much worse results on NLP tasks. On Random-Bench , pre-trained networks bring large performance improvements over random-initialized baseline (Transformer). However, on our benchmark, all pre-trained networks yield surprising performance drops. These results demonstrate that the results of pre-trained models on NLP tasks are more easily over-estimated.

Achilles-Bench (Loss) is more challenging than Achilles-Bench (GradNorm) We implement two metrics to select hard examples, including loss and gradient norm. Despite similar motivation, Achilles-Bench (loss) is more challenging than Achilles-Bench (GradNorm) according to our experimental results. On NLP tasks, Achilles-Bench (loss) also witnesses the worst results. Loss is the

Models	SST-2	COLA	MNLI	QNLI	MRPC	QQP	RTE	WNLI	Average
Random-Bench									
BLOOM-1.1B (Scao et al., 2022)	50.5	60.4	35.4	50.5	66.2	51.8	52.7	42.3	51.2
Llama-7B (Touvron et al., 2023a)	60.2	63.1	33.1	48.3	67.4	47.9	51.0	47.9	52.4
Llama2-7B (Touvron et al., 2023b)	95.4	68.9	53.7	58.0	68.1	73.7	79.4	63.4	68.5
Llama2-13B (Touvron et al., 2023b)	85.1	80.5	49.5	54.9	70.5	78.1	75.3	68.5	**70.3**
Llama2-70B (Touvron et al., 2023b)	90.3	78.8	61.7	49.8	68.4	42.4	79.2	85.5	69.5
Achilles-Bench (Loss)									
BLOOM-1.1B (Scao et al., 2022)	50.1	46.4	35.42	50.0	65.9	60.8	47.3	43.7	50.0
Llama-7B (Touvron et al., 2023a)	40.7	61.4	30.6	46.3	68.1	40.4	49.1	42.3	**47.4**
Llama2-7B (Touvron et al., 2023b)	64.6	53.2	46.4	59.7	68.1	79.5	76.5	64.8	63.0
Llama2-13B (Touvron et al., 2023b)	48.4	78.6	43.0	47.4	69.6	76.8	74.7	66.2	63.1
Llama2-70B (Touvron et al., 2023b)	48.4	71.0	43.8	47.0	68.4	37.2	76.1	90.1	60.3

Table 2: The in-context learning results of LLMs on NLP datasets. Achilles-Bench (Loss) consistently preserve its challenges for LLMs.

most direct signal to see how neural networks understand an example. These difficult examples confuse neural networks, which barely learn core features. This learning weakness is not covered by existing low-resource benchmarks. Achilles-Bench provides a new perspective for understanding the learning abilities of different models.

Data augmentation slightly improves results Table 4 shows the results on CIFAR-10 with data augmentation techniques, cutmix (Yun et al., 2019). We can see that data augmentation brings slight performance improvements, but also faces the challenges of generalization on our benchmarks.

Models	CIFAR10	CIFAR100	ImageNet
Random-Bench			
FFN	48.91±0.87	14.95±0.29	5.12±0.30
VGG-16	62.15±0.71	26.55±0.20	16.02±0.27
ResNet-18	65.47±0.84	25.49±0.60	29.34±0.31
DenseNet-121	71.33±0.56	33.66±1.48	**35.20**±0.41
ViT-B/16	97.20±0.22	**83.93**±0.43	-
EfficientNetV2-S	91.41±0.60	70.41±0.74	-
Achilles-Bench (GradNorm)			
FFN	29.64±0.88	8.75±0.28	3.13±0.18
VGG-16	55.11±0.89	17.22±0.44	9.51±0.20
ResNet-18	46.87±2.41	15.50±0.85	23.81±0.76
DenseNet-121	59.87±0.66	20.96±0.94	28.96±0.67
ViT-B/16	**97.39**±0.10	82.36±0.94	-
EfficientNetV2-S	92.51±0.24	69.56±0.49	-
Achilles-Bench (Loss)			
FFN	**17.26**±0.82	**3.18**±0.21	**2.66**±0.02
VGG-16	27.58±0.62	7.14±0.24	7.27±0.24
ResNet-18	33.20±1.00	6.96±0.32	13.34±0.19
DenseNet-121	44.81±2.30	11.59±0.98	22.00±0.46
ViT-B/16	96.85±0.11	80.87±0.58	-
EfficientNetV2-S	89.88±0.63	60.42±1.85	-

Table 3: Results on CV datasets. ViT and efficientNetV2-S are pre-trained on ImageNet. So we do not report their results on ImageNet to avoid data leak issues.

Models	Random-Bench	Achilles-Bench (GradNorm)	Achilles-Bench (Loss)
FFN	53.99±0.39	30.36±1.26	19.29±0.36
VGG-16	66.76±0.59	47.85±0.97	33.64±0.25
ResNet-18	68.94±0.66	52.73±1.54	37.96±1.12
DenseNet-121	75.44±0.34	63.23±0.42	47.70±1.38
ViT-B/16	97.71±0.17	97.79±0.08	97.09±0.12
EfficientNetV2-S	93.25±0.65	92.83±0.63	91.41±0.69

Table 4: Results with cutmix. Models with data augmentation still face the challenges of generalization on our benchmarks.

Models	Random-Bench	Achilles-Bench (Loss)	FewGLUE
RoBERTa	57.8± 3.62	**52.0**	62.8
GPT-2	58.8± 2.65	**47.3**	47.7

Table 5: Results compared with FewGLUE on 32-shot RTE.

Models	Achilles-Bench (Loss)		Forget Statistic	
	Accuracy	Gap	Accuracy	Gap
FFN	16.17	**30.33**	33.11	13.39
VGG-16	26.78	**33.03**	43.00	16.81
ResNet-18	32.10	**30.64**	45.77	16.97
DenseNet-121	41.45	**28.80**	59.63	10.62
ViT	96.70	**0.25**	97.48	-0.53
EfficientNet-V2	89.17	**0.54**	91.10	-1.39

Table 6: The comparison between Achilles-Bench (Loss) and Forget Statistic on CIFAR-10. "Gap" represents the test accuracy gap with Bench-Random.

Achilles-Bench (Loss) demonstrate greater challenges compared to FewGLUE (Schick and Schütze, 2020) Table 5 presents a performance comparison between RoBERTa and GPT-2 on the 32-shot RTE task. The performance of GPT-2 under both the Achilles-Bench (Loss) and FewGLUE approaches tends to resemble random selection. Regarding RoBERTa, FewGLUE does not seem significantly more challenging than Random-Bench , whereas Achilles-Bench (Loss) demonstrates a higher level of difficulty.

Results on different metrics Table 6 presents the outcomes obtained on the 500-shot datasets from CIFAR-10 using the forget statistic technique (Toneva et al., 2018). Achilles-Bench (Loss) surpasses the forget statistic approach in all models, including pre-trained models. The forget statistic technique does not appear to be more challenging than Random-Bench for pre-trained models.

5.4 Ablation Studies

Massive sampling fails to find a challenging benchmark In Random-Bench, we report the average results over 3 random samplings. In this part, we conduct 100 samplings and report the worst result in Figure 6 to figure out whether our methods can be replaced with massive sampling. As we can see, there is still a large gap between the worst results on Random-Bench and Achilles-Bench, indicating that the proposed method is an effective method to build challenging benchmarks.

Results on the selected set as the test set Figure 7 shows results on the selected set as the test set. As we can see, these "hard examples" capture the weakness of neural networks. If neural networks has not seen these examples, they fail on them.

Ablation studies on different models as predictors In our framework, we introduce a weak classifier as a biased predictor. For simplification, we choose FFN for CV datasets and BERT for NLP datasets. We conduct experiments on more networks to see whether the choice of predictors affects our conclusions. Table 7 and Table 8 show the attack results on SST-2 and CIFAR-10. For SST-2, we test two more models: randomly-initialized Transformer and GPT2, as predictors. For CV models, we test two more models: ResNet-18 and ViT-B/16, as predictors. All models show consistent

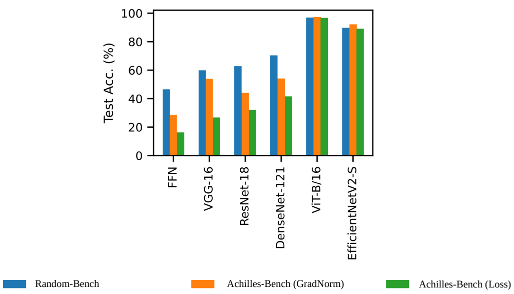


Figure 6: The worst performances among all the performances on CIFAR-10.

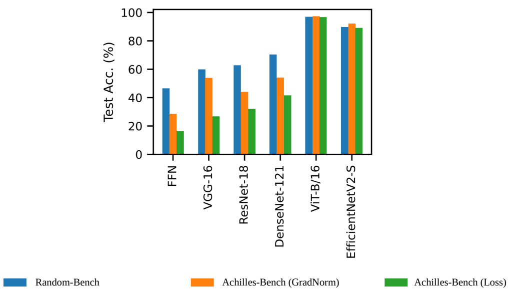


Figure 7: Results on the selected set as the test set.

performance drops, indicating that our method is a universal model to generate challenging datasets to attack various models.

Models	Transformer Predictor Accuracy	Transformer Predictor Gap	GPT-2 Predictor Accuracy	GPT-2 Predictor Gap
Transformer	51.17± 0.17	16.99	50.55± 0.73	**17.61**
BERT	51.06± 2.51	37.62	48.30± 2.00	**40.38**
GPT-2	50.46± 3.20	37.62	48.88± 4.00	**39.20**
RoBERTa	54.72± 3.04	36.82	48.33± 2.19	**43.21**
T5	60.48± 3.88	28.25	56.03± 2.62	**32.70**

Table 7: Results of Achilles-Bench (Loss) on SST-2 based on a random initialized Transformer and GPT-2. "Gap" represents the test accuracy gap with Bench-Random.

Models	ResNet Predictor Accuracy	ResNet Predictor Gap	ViT Predictor Accuracy	ViT Predictor Gap
FFN	40.69± 0.69	8.32	39.82± 0.61	**9.19**
VGG-16	51.83± 0.39	16.80	48.19± 0.64	**20.44**
ResNet-18	53.93± 0.72	11.58	50.59± 0.98	**14.92**
DenseNet-121	61.70± 0.23	9.72	58.05± 0.80	**13.37**
ViT-B/16	97.07± 0.19	0.00	96.92± 0.32	**0.15**
EfficientNet-V	89.70± 0.32	2.12	87.26± 1.01	**4.56**

Table 8: Results of Achilles-Bench (Loss) on CIFAR10 based on ResNet-18 and ViT-B/16. "Gap" represents the test accuracy gap with Bench-Random.

5.5 Explaining the Effectiveness of Achilles-Bench with Visualization

In this section, we compare samples selected by our Achilles-Bench with samples from Random-Bench to demonstrate our approach reaches the goal of building difficult low-resource training set with shifted distributions. To make our observation more straightforward, we show visualizations in the Appendix G. We make the following observations based on these visualization results:

Achilles-Bench induces bias in the low-resource training set From visualizations, we can see that both GradNorm and Loss variations of Achilles-Bench construct training sets that are drastically different from the data distribution. For SST2 task, specifically, Random-Bench exhibits ordinary statements containing words with clear emotional expressions. In contrast, both GradNorm and the Loss variations of Achilles-Bench opt for shorter sentences, incorporating statements with implicit emotional nuances. Similar biases are evident in other classes, showing that our approach successfully induces bias in the low-resource training set.

Achilles-Bench find challenging samples The method selects tough examples from datasets using difficulty metrics, notably in GradNorm and Loss. In the SST2 task, it favors terse, uninformative samples or input sentences that use sophisticated vocabularies.

6 Conclusion

This paper proposes a challenging benchmark for low-resource learning. We first analyze which factors affect the difficulty of low-resource learning. We prove that low-resource generalization results in worse performance with more difficult and biased datasets. Hence we choose two metrics for measuring data difficulty, which result in two variants, Achilles-Bench (Loss) and Achilles-Bench (GradNorm) . Experiments show that both can better tell the learning gap between existing models than randomly-sampled low-resource datasets.

7 Acknowledge

This research is supported by the National Key Research and Development Program of China 2020AAA0106700, the joint research scheme of the National Natural Science Foundation of China (NSFC) and the Research Grants Council (RGC) under grant number N_HKU714/21.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*.

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4349–4357.

Jonathan Bragg, Arman Cohan, Kyle Lo, and Iz Beltagy. 2021. Flex: Unifying evaluation for few-shot nlp. *Advances in Neural Information Processing Systems*, 34:15787–15800.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proc. of CVPR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. of ICLR*.

Vincent Dumoulin, Neil Houlsby, Utku Evci, Xiaohua Zhai, Ross Goroshin, Sylvain Gelly, and Hugo Larochelle. 2021. A unified few-shot classification benchmark to compare transfer and meta learning approaches. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Gamaleldin Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. 2018. Large margin deep networks for classification. *Advances in neural information processing systems*, 31.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*.

Guy Hacohen and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*, pages 2535–2544. PMLR.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *Proc. of ICLR*.

Lijie Hu, Chenyang Ren, Zhengyu Hu, Cheng-Long Wang, and Di Wang. 2024. Editable concept bottleneck models. *arXiv preprint arXiv:2405.15476*.

Zhengyu Hu, Jieyu Zhang, Haonan Wang, Siwei Liu, and Shangsong Liang. 2023a. Leveraging relational graph neural network for transductive model ensemble. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 775–787.

Zhengyu Hu, Jieyu Zhang, Yue Yu, Yuchen Zhuang, and Hui Xiong. 2023b. How many validation labels do you need? exploring the design space of label-efficient model ranking. *arXiv preprint arXiv:2312.01619*.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proc. of CVPR*.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. 2020. Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 491–507. Springer.

Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Robert L Logan IV, Ivana Balažević, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2021. Cutting down on prompts and parameters: Simple few-shot learning with language models. *arXiv preprint arXiv:2106.13353*.

Subhabrata Mukherjee, Xiaodong Liu, Guoqing Zheng, Saghar Hosseini, Hao Cheng, Greg Yang, Christopher Meek, Ahmed Hassan Awadallah, and Jianfeng Gao. 2021. Clues: few-shot learning evaluation in natural language understanding. *arXiv preprint arXiv:2111.02570*.

Nikita Nangia and Samuel R. Bowman. 2019. Human vs. muppet: A conservative estimate of human performance on the GLUE benchmark. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4566–4575. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. 2021. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. 2020. An investigation of why overparameterization exacerbates spurious correlations. In *Proc. of ICML*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. *Computing Research Repository*, arXiv:2009.07118.

Hyunjune Sebastian Seung, Haim Sompolinsky, and Naftali Tishby. 1992. Statistical mechanics of learning from examples. *Physical review A*, 45(8):6056.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Leslie N Smith and Nicholay Topin. 2019. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S Morcos. 2022. Beyond neural scaling laws: beating power law scaling via data pruning. *arXiv preprint arXiv:2206.14486*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew K. Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakas, and et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *CoRR*, abs/2206.04615.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022a. Challenging big-bench tasks and whether chain-of-thought can solve them. *CoRR*, abs/2210.09261.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022b. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Mingxing Tan and Quoc V. Le. 2021. Efficientnetv2: Smaller models and faster training. In *Proc. of ICML*.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. 2018. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Liang Xu, Xiaojing Lu, Chenyang Yuan, Xuanwei Zhang, Huilin Xu, Hu Yuan, Guoao Wei, Xiang Pan, Xin Tian, Libo Qin, et al. 2021. Fewclue: A chinese few-shot learning evaluation benchmark. *arXiv preprint arXiv:2107.07498*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. Crossfit: A few-shot learning challenge for cross-task generalization in nlp. *arXiv preprint arXiv:2104.08835*.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032.

Jieyu Zhang, Bohan Wang, Zhengyu Hu, Pang Wei W Koh, and Alexander J Ratner. 2024. On the trade-off of intra-/inter-class diversity for supervised pre-training. *Advances in Neural Information Processing Systems*, 36.

Yanan Zheng, Jing Zhou, Yujie Qian, Ming Ding, Chonghua Liao, Jian Li, Ruslan Salakhutdinov, Jie Tang, Sebastian Ruder, and Zhilin Yang. 2021. Fewnlu: Benchmarking state-of-the-art methods for few-shot natural language understanding. *arXiv preprint arXiv:2109.12742*.

A Limitation and Future Work

The method proposed in this paper can be extended to a wider range of tasks and datasets. In future studies, we aim to expand the tasks to more challenging datasets, such as superGLUE.

Furthermore, we did not test the results on the latest models, such as GPT-4. We intend to extend the tasks to include the latest large-scale models in the future.

B Perceptron Model of Low Resource Learning

In this section, notations are defined as follows. We look at the teacher-student setting in learning perceptrons. Consider a large curated dataset of N examples $D = \{x_i, y_i\}_{i \in [N]}$ where $x_i \in \mathbb{R}^d$ are i.i.d. random Gaussian inputs $x_i \sim \mathcal{N}(0, I_d)$, with labels generated by a teacher perceptron $T \in \mathbb{R}^d$ as $y_i = \text{sign}(Tx_i)$. The number of samples $N \to \infty$ but sample per parameter $\alpha = \frac{N}{d} = O(1)$ to remain trainable. Now we consider the low resource scenario where the number of training samples available P is much less than N, where $\alpha_{\text{low}} = \frac{P}{d} \to 0$. For convenience, we sample the data for low resource learning from dataset D such that $D_{\text{low}} = \{x_\mu, y_\mu\}_{i \in [P]} \subset D$. Learning on D_{low}, we obtain a new student perceptron J that has generalization error ϵ_g.

In the basic low-resource learning scenario, we use a uniform sampling strategy to obtain D_{low} from D. We model ϵ_g as a function of α_{low}. The results are as follows.

Lemma B.1. *(Low-resource Learning, Seung et al. (1992)) For student perceptron J learned on high dimension dataset D_{low}, the generalization error satisfies,*

$$\epsilon_g \propto \alpha_{low}^{-1} \tag{4}$$

For other settings, the generalization error is only related to the angle between teacher model T and learned student model J. $\epsilon_g = \arccos R/\pi$, $R = \frac{JT}{|J||T|}$. Based on different low-resource dataset sampling strategies, we calculate the teacher-student overlap R with the geometry of each dataset distribution. Sorscher et al. (2022) has proved similar results in data pruning with our hard and biased learning setting. Here we only cite their results and don't elaborate on proofs. Note that despite the proofs being similar, we use a different setting in perceptron learning. Their main objective is to understand how data-pruning can improve data efficiency, while we take an inverse stand, trying to understand challenging settings of low-resource learning.

Lemma B.2. *(Hard Low-resource Learning, Sorscher et al. (2022)) D_{low} is sampled from D such that $\forall x_\mu \in D_{low}, \forall x_\gamma \in D/D_{low}$, their margins satisfy $|Tx_\mu| \geq |Tx_\mu|$. Let J be the student perceptron learned on high dimension dataset D_{low}, and κ be the minimum margin $\min_\mu J(x^\mu y^\mu)$. If the perceptron is trained to maximum margin, the generalization error of J satisfies,*

$$\epsilon_g = \arccos R/\pi \tag{5}$$

where R satisfies the saddle point equation,

$$\begin{aligned} R = \frac{2\alpha}{f\sqrt{2\pi}\sqrt{1-R^2}} \int_{-\infty}^{\kappa} Dt \, \exp\left(-\frac{R^2t^2}{2(1-R^2)}\right) \\ \cdot \left[1 - \exp\left(-\frac{\gamma(\gamma - 2Rt)}{2(1-R^2)}\right)\right](\kappa - t) \end{aligned} \tag{6}$$

in which $\gamma = H^{-1}(\frac{N-P}{2N})$, $p(z) = \frac{e^{-\frac{z^2}{2}} N}{\sqrt{2\pi} P} \Theta(\gamma - |z|)$

The proof for this lemma can be found in Sorscher et al. (2022) A.5.1 and is omitted here for brevity.

C Low-Resource Generalization

C.1 Proof for Theorem 3.1

Lemma C.1. *Define $\epsilon_S(h, f) := E_{x \sim S}|\delta(h(x)) - \delta(f(x))|$. For any hypothesis $h, h' \in \mathcal{H}$, there exists $\epsilon_H > 0$ which satisfies,*

$$|\epsilon_{P_{low}}(h,h') - \epsilon_P(h,h')| \leq \textit{MMD}(\mathcal{H}, P_{low}, P) + \frac{\epsilon_{\mathcal{H}}}{2} \tag{7}$$

ϵ_H is a constant for the complexity of hypothesis space.

Lemma C.2. *Let f_{low} be the trained classifier on the low resource distribution P_{low}, and f be the trained classifier on distribution P. Since P_{low} is formed by a subset of the training examples, when training error $\epsilon_P(f) \to 0$ and $\epsilon_{P_{low}}(f_{low}) \to 0$, $\epsilon_{P_{low}}(f_{low}, f) \leq \epsilon_\alpha$, where ϵ_α is a constant approaching zero.*

Proof.

$$\begin{aligned}
\left|\epsilon_{P_{I_k}}\left(h,h'\right) - \epsilon_P\left(h,h'\right)\right| &\leq \sup_{h,h'\in\mathcal{H}} \left|\epsilon_{P_{I_k}}\left(h,h'\right) - \epsilon_P\left(h,h'\right)\right| \\
&= \sup_{h,h'\in\mathcal{H}} \left|\mathbf{P}_{\boldsymbol{x}\sim P_{I_k}}\left[\delta(h(\boldsymbol{x})) \neq \delta(h'(\boldsymbol{x}))\right] - \mathbf{P}_{\boldsymbol{x}\sim P}\left[\delta(h(\boldsymbol{x})) \neq \delta(h'(\boldsymbol{x}))\right]\right| \\
&= \sup_{h,h'\in\mathcal{H}} \left|\mathbf{P}_{\boldsymbol{x}\sim P_{I_k}}\left[h(\boldsymbol{x}) \neq h'(\boldsymbol{x})\right] - \mathbf{P}_{\boldsymbol{x}\sim P}\left[h(\boldsymbol{x}) \neq h'(\boldsymbol{x})\right]\right| \\
&= \sup_{h,h'\in\mathcal{H}} \left|\int_{\mathcal{X}} \mathbf{1}_{h(\boldsymbol{x})\neq h'(\boldsymbol{x})} d\mu_{P_{I_k}} - \int_{\mathcal{X}} \mathbf{1}_{h(\boldsymbol{x})\neq h'(\boldsymbol{x})} d\mu_P\right|
\end{aligned} \tag{8}$$

Lemma C.3. *Let f be the trained classifier on dataset D that is drawn i.i.d. from distribution P. f is tested on a test dataset S that is also drawn i.i.d. from the distribution P. Let m be the maximum margin of classifier f. Then with probability at least $1-\delta$,*

$$\epsilon_S(f) \leq \epsilon_D(f) + c\sqrt{\frac{|\mathcal{H}|\ln m + \ln\left(\frac{1}{\delta}\right)}{m}} \tag{9}$$

where $\epsilon_D(f)$ is the error on training set, and $\epsilon_S(f)$ be the error on test set.

Following Ben-David et al. (2010), we use Lemma C.1 and C.2 to prove Theorem 3.1.

Proof

$$\begin{aligned}
\epsilon_Q(f_{low}) &\leq \epsilon_Q(f) + \epsilon_Q(f_{low}, f) \\
&= \epsilon_Q(f) + \epsilon_{P_{low}}(f_{low}, f) + (\epsilon_Q(f_{low}, f) - \epsilon_P(f_{low}, f)) + (\epsilon_P(f_{low}, f) - \epsilon_{P_{low}}(f_{low}, f)) \\
&\leq \epsilon_Q(f) + \epsilon_{P_{low}}(f_{low}, f) + |\epsilon_P(f_{low}, f) - \epsilon_Q(f_{low}, f)| + |\epsilon_{P_{low}}(f_{low}, f) - \epsilon_P(f_{low}, f)| \\
&\leq \epsilon_Q(f) + \epsilon_\alpha + |\epsilon_P(f_{low}, f) - \epsilon_Q(f_{low}, f)| + \text{MMD}(P_{low}, P) + \epsilon_{\mathcal{H}}
\end{aligned} \tag{10}$$

In which,

$$\begin{aligned}
|\epsilon_P(f_{low}, f) - \epsilon_Q(f_{low}, f)| &= \left|\int_{\mathcal{X}} \mathbf{1}_{f_{low}(\boldsymbol{x})\neq f(\boldsymbol{x})} d\mu_P - \int_{\mathcal{X}} \mathbf{1}_{f_{low}(\boldsymbol{x})\neq f(\boldsymbol{x})} d\mu_Q\right| \\
&= |\sum_{i=1}^{n} \mathbf{1}_{f_{low}(\boldsymbol{x_i})\neq f(\boldsymbol{x_i})} - E_Q \mathbf{1}_{f_{low}(\boldsymbol{x})\neq f(\boldsymbol{x})}|
\end{aligned} \tag{11}$$

Here suppose test set Q matches the distribution of data for this classification task, and P is constructed by sampling n i.i.d. samples from the distribution Q. Using Lemma C.3 we have,

$$P(|\epsilon_P(f_{low}, f) - \epsilon_Q(f_{low}, f)| > c\sqrt{\frac{|\mathcal{H}|\ln m + \ln\left(\frac{2}{\delta}\right)}{m}}) \leq \delta \tag{12}$$

Therefore, with a probability over $1-\delta$, we have

$$\epsilon_Q(f_{low}) \leq \epsilon_Q(f) + \text{MMD}(P_{low}, P) + \epsilon_\alpha + \epsilon_{\mathcal{H}} + c\sqrt{\frac{|\mathcal{H}|\ln m + \ln\left(\frac{2}{\delta}\right)}{m}} \tag{13}$$

□

Table 9: Hyper-parameter settings. The Linear refers LinearLR scheduler in Pytorch. OneCycle refers 1-cycle learning rate policy (Smith and Topin, 2019).

Models	Datasets	Batch Size	Epochs	Optimizer	Learning Rate
FFN	CIFAR-10	128	50	Adam	[1e-3, 5e-4, 2.5e-4]
	CIFAR-100	128	50	Adam	[1e-3, 5e-4, 2.5e-4]
	ImageNet-1K	32	30	SGD	[0.01, 0.001, 0.0001]
VGG	CIFAR-10	128	50	Adam	[1e-4, 5e-5, 2.5e-5]
	CIFAR-100	128	50	Adam	[1e-4, 5e-5, 2.5e-5]
	ImageNet-1K	32	30	SGD	[0.01, 0.001, 0.0001]
ResNet	CIFAR-10	128	50	Adam	[1e-3, 5e-4, 2.5e-4]
	CIFAR-100	128	50	Adam	[1e-3, 5e-4, 2.5e-4]
	ImageNet-1K	32	30	SGD	[0.1, 0.01, 0.001]
DenseNet	CIFAR-10	128	50	Adam	[1e-3, 5e-4, 2.5e-4]
	CIFAR-100	128	50	Adam	[1e-3, 5e-4, 2.5e-4]
	ImageNet-1K	32	30	SGD	[0.1, 0.01, 0.001]
ViT-B/16	CIFAR-10	32	10	Adam	5e-5 (Linear)
	CIFAR-100	32	10	Adam	5e-5 (Linear)
EfficientNetV2-S	CIFAR-10	32	10	AdamW	1e-3 (OneCycle)
	CIFAR-100	32	10	AdamW	1e-3 (OneCycle)

D Models and Hyperparameters

We implement the following models for experiments in this paper.

1) **FFN**, a feed-forward neural network with two convolution and pooling layers and three feed-forward layers. 2) **VGG** (Simonyan and Zisserman, 2014), a classical convolutional neural network. We use the VGG-16 with 13 convolution layers and three fully connected layers as implementation. 3) **ResNet** (He et al., 2016), a residual neural network. We use the ResNet-18 with 16 residual blocks, one convolution layer, and one fully connected layer as implementation. 4) **DenseNet** (Huang et al., 2017). We use DenseNet-121 with 121 layers, one convolution layer, and one fully connected layer as re-implementation.[3] Besides, to verify the attack ability Gradon the pre-trained models, we also re-implement two pre-trained models: 1) Transformer-based **ViT** (Dosovitskiy et al., 2021)[4] and 2) Convolutional-based **EfficientNetV2** (Tan and Le, 2021)[5]. For FFN, VGG, ResNet, ResNeXt, and DenseNet on ImageNet, we resize all the images into 256×256 and then center-crop them into 224×224. For ViT on CIFAR, we resize all the images into 224×224, while 384×384 for EfficientNetV2.

We list hyper-paramters in Table 9. All the SGD optimizers are with a momentum of 0.9. For Adam/AdamW, we set $\beta = (0.9, 0.999)$. We employed the torch.optim.lr_scheduler.MultiStepLR module to dynamically adjust the learning rate during training. Specifically, we set the milestones at epochs 20 and 40 (for ImageNet-1K, epochs 10 and 20 respectively) to adaptively update the learning rate based on the progress of training. The corresponding learning rate values used during three periods in our experiments are provided in Table 9. We conduct all the experiments on a single A100 GPU. We use Adam as the optimizer for all the NLP tasks with a learning rate of $2e-5$ and a linear scheduler.

[3] For VGG, ResNet, ResNeXt, and DenseNet on CIFAR and MNIST, we use the implementation from `https://github.com/kuangliu/pytorch-cifar`. As for ImageNet, we use the implementation from torch.models.

[4] We use the implementation from `https://huggingface.co/google/vit-base-patch16-224`

[5] We use the implementation from torch.models.

E Details of ICL and LLMs

Considering the sentence length of different tasks and limitations of the GPU, we tested SST2 and COLA with 16-shots, MNLI, QNLI, MRPC, RTE, WNLI with 8-shots each, and QQP with 4-shots.

We generate the prompt refer to lm-eval, the results of the prompt with zero-shots are shown below.

Models	SST-2	COLA	MNLI	QNLI	MRPC	QQP	RTE	WNLI	Average
Llama2-7B	86.70	38.70	50.10	62.00	58.09	63.30	72.56	61.97	61.68

Table 10: The zero-shot results of LLaMA2 on NLP datasets.

F Results with Different Shots

To better explore the effectiveness of Achilles-Bench (GradNorm) and Achilles-Bench (Loss) , we demonstrate the results with different shots. The results are shown in the following tables.

Based on the results presented in Table 11 and Table 12, as compared to the findings discussed in Section 5.4, it is evident that all the model demonstrates a notable decline in performance when trained on a more limited dataset (20, 50-shot) in Achilles-Bench (Loss) and Achilles-Bench (GradNorm) , as compared to Random-Bench . This observation suggests that Achilles-Bench (Loss) and Achilles-Bench (GradNorm) pose greater challenges with fewer shots. While the pretrained model exhibits some level of robustness in Section 5.4, its performance still suffers when faced with more limited data. This highlights the significance of employing challenging benchmarks that incorporate scenarios with limited training data.

The results in NLP, as shown in Table 14, Table 15, Table 16, are generally consistent with the previous findings presented in Section 5.4. Nonetheless, a few specific models, characterized by inadequate few-shot learning capabilities, exhibited poor performance across all three benchmarks.

Table 11: Results on CIFAR-10.

shots	Models	Random-Bench Accuracy	Achilles-Bench (GradNorm) Accuracy	Achilles-Bench (GradNorm) Gap	Achilles-Bench (Loss) Accuracy	Achilles-Bench (Loss) Gap
20-shot	FFN	27.28± 1.51	13.68± 0.57	13.60	10.74± 1.38	16.54
	VGG-16	31.73± 1.26	14.76± 0.86	16.97	10.27± 0.32	21.46
	ResNet-18	30.54± 1.82	14.80± 0.56	15.74	10.79± 0.35	19.75
	DenseNet-121	34.69± 1.51	15.25± 0.29	19.44	10.15± 0.61	24.54
	ViT-B/16	79.84± 1.70	62.92± 1.97	16.92	57.62± 2.46	22.22
	EfficientNetV2-S	61.59± 4.36	40.67± 3.38	20.92	31.44± 3.86	30.15
50-shot	FFN	33.31± 1.01	14.15± 0.71	19.16	9.94± 0.98	23.37
	VGG-16	38.95± 0.61	17.47± 0.96	21.48	10.36± 0.45	28.59
	ResNet-18	39.18± 1.19	17.78± 0.62	21.40	10.64± 0.64	28.54
	DenseNet-121	43.64± 0.68	18.56± 0.43	25.08	10.15± 0.79	33.49
	ViT-B/16	87.92± 0.45	82.77± 1.76	5.15	81.05± 2.08	6.87
	EfficientNetV2-S	74.75± 1.05	59.92± 3.84	14.83	56.17± 3.56	18.58
200-shot	FFN	41.98± 0.79	21.05± 0.28	20.93	13.11± 0.73	28.87
	VGG-16	52.87± 0.78	25.29± 0.46	27.58	15.35± 0.91	37.52
	ResNet-18	53.67± 0.95	25.58± 0.42	28.09	15.87± 0.51	37.80
	DenseNet-121	61.69± 0.36	33.06± 1.57	28.63	19.48± 0.86	42.21
	ViT-B/16	95.30± 0.14	95.77± 0.19	-0.47	95.22± 0.26	0.08
	EfficientNetV2-S	88.25± 0.23	83.61± 1.24	4.64	82.28± 1.95	5.97
2000-shot	FFN	58.83± 1.44	46.19± 0.66	12.64	44.56± 1.86	14.27
	VGG-16	78.50± 0.59	77.58± 0.40	0.92	76.34± 0.55	2.16
	ResNet-18	79.40± 0.35	79.00± 0.37	0.40	78.14± 0.17	1.26
	DenseNet-121	84.65± 0.34	84.70± 0.17	-0.05	83.58± 0.30	1.07
	ViT-B/16	97.86± 0.09	98.06± 0.12	-0.20	98.01± 0.08	-0.15
	EfficientNetV2-S	95.30± 0.18	95.79± 0.09	-0.49	95.07± 0.16	0.23

Table 12: Resuls on CIFAR100.

shots	Models	Random-Bench Accuracy	Achilles-Bench (GradNorm) Accuracy	Gap	Achilles-Bench (Loss) Accuracy	Gap
20-shot	FFN	10.48± 0.32	5.69± 0.11	4.79	1.90± 0.20	8.58
	VGG-16	18.87± 0.46	10.42± 0.17	8.45	2.77± 0.16	16.10
	ResNet-18	17.20± 0.57	9.01± 0.34	8.19	2.61± 0.11	14.59
	DenseNet-121	21.48± 0.80	10.84± 0.88	10.64	3.24± 0.23	18.24
	ViT-B/16	68.23± 1.68	61.45± 4.99	6.78	54.99± 2.04	13.24
	EfficientNetV2-S	55.10± 0.18	51.87± 1.43	3.23	40.68± 1.30	14.42
200-shot	FFN	23.67± 1.00	16.83± 0.46	6.84	12.91± 1.45	10.76
	VGG-16	45.52± 0.45	41.41± 0.77	4.11	36.22± 0.34	9.30
	ResNet-18	44.37± 0.70	41.03± 1.03	3.34	36.95± 1.02	7.42
	DenseNet-121	53.75± 0.39	51.80± 0.61	1.95	48.04± 0.37	5.71
	ViT-B/16	88.89± 0.24	89.00± 0.21	-0.11	88.86± 0.40	0.03
	EfficientNetV2-S	79.63± 0.64	80.36± 0.45	-0.73	78.32± 0.43	1.31

Table 13: Results on ImageNet.

shots	Models	Random-Bench Accuracy	Achilles-Bench (GradNorm) Accuracy	Gap	Achilles-Bench (Loss) Accuracy	Gap
50-shot	FFN	3.93± 0.51	1.59± 0.09	2.34	1.72± 0.04	2.21
	VGG-16	6.94± 0.43	2.11± 0.23	4.83	2.97± 0.23	3.97
	ResNet-18	18.84± 0.46	11.67± 0.27	7.17	9.46± 0.24	9.38
	DenseNet-121	22.96± 0.44	13.89± 0.45	9.07	10.29± 0.20	12.67

Table 14: The results on GLUE with 16-shots.

Datasets	Models	Random-Bench Accuracy	Achilles-Bench (GradNorm) Accuracy	Achilles-Bench (GradNorm) Gap	Achilles-Bench (Loss) Accuracy	Achilles-Bench (Loss) Gap
SST2	Transformer	52.64± 2.16	52.89± 0.56	-0.25	52.34± 0.26	0.30
	BERT	68.39± 7.14	56.86± 4.88	11.53	50.28± 0.88	18.11
	GPT-2	55.62± 4.12	52.52± 2.00	3.10	51.54± 1.22	4.08
	RoBERTa	76.67± 3.44	58.12± 1.47	18.55	50.25± 0.96	26.42
	T5	55.94± 3.74	51.95± 1.90	3.99	51.19± 2.26	4.75
COLA	Transformer	68.95± 0.44	68.74± 0.86	0.21	68.88± 0.50	0.07
	BERT	66.94± 3.55	64.99± 6.39	1.95	58.16± 13.44	8.78
	GPT-2	66.40± 5.50	66.19± 5.92	0.21	66.56± 5.19	-0.16
	RoBERTa	69.66± 1.02	65.23± 5.14	4.43	49.38± 11.73	20.28
	T5	55.82± 8.92	59.54± 4.84	-3.72	56.80± 6.79	-0.98
MNLI	Transformer	35.40± 0.09	35.45± 0.00	-0.05	35.28± 0.21	0.12
	BERT	36.21± 0.96	34.21± 0.54	2.00	34.03± 0.96	2.18
	GPT-2	37.63± 1.29	34.40± 1.39	3.23	33.88± 1.31	3.75
	RoBERTa	43.13± 2.07	35.48± 0.82	7.65	33.38± 1.10	9.75
	T5	33.98± 0.50	33.44± 0.21	0.54	33.39± 0.18	0.59
QNLI	Transformer	53.48± 2.46	50.95± 0.55	2.53	51.22± 0.38	2.26
	BERT	53.75± 0.69	50.79± 0.31	2.96	50.10± 0.66	3.65
	GPT-2	55.16± 3.26	53.65± 2.94	1.51	52.49± 2.08	2.67
	RoBERTa	63.52± 3.92	50.80± 0.37	12.72	49.78± 0.40	13.74
	T5	54.03± 2.36	50.85± 1.02	3.18	49.69± 0.94	4.34
MRPC	Transformer	68.63± 0.31	68.38± 0.00	0.25	68.33± 0.10	0.30
	BERT	66.47± 3.22	63.19± 6.52	3.28	54.61± 16.68	11.86
	GPT-2	67.75± 1.53	66.23± 4.44	1.52	63.87± 8.90	3.88
	RoBERTa	69.26± 1.48	57.60± 7.79	11.66	33.33± 0.83	35.93
	T5	58.58± 5.94	59.90± 4.06	-1.32	56.32± 8.15	2.26
QQP	Transformer	63.75± 0.55	63.23± 0.08	0.52	63.19± 0.02	0.56
	BERT	64.81± 2.15	59.19± 2.86	5.62	57.27± 4.69	7.54
	GPT-2	62.57± 1.34	54.64± 4.89	7.93	56.84± 3.40	5.73
	RoBERTa	65.55± 1.36	63.18± 0.00	2.37	63.10± 0.10	2.45
	T5	55.49± 3.35	56.61± 3.44	-1.12	56.14± 2.58	-0.65
RTE	Transformer	53.72± 0.90	54.95± 1.01	-1.23	54.80± 0.42	-1.08
	BERT	55.02± 1.56	53.43± 2.41	1.59	50.40± 2.59	4.62
	GPT-2	58.77± 3.98	58.84± 2.84	-0.07	52.71± 1.69	6.06
	RoBERTa	55.16± 1.73	53.14± 0.42	2.02	52.56± 0.37	2.60
	T5	51.05± 2.44	49.03± 1.76	2.02	49.10± 0.97	1.95
WNLI	Transformer	58.03± 2.07	56.62± 0.56	1.41	57.46± 1.05	0.57
	BERT	56.34± 6.96	54.37± 4.51	1.97	56.62± 2.87	-0.28
	GPT-2	56.34± 0.00	56.62± 1.38	-0.28	56.90± 2.61	-0.56
	RoBERTa	57.75± 3.09	56.90± 1.13	0.85	56.34± 1.54	1.41
	T5	58.31± 0.69	53.52± 5.12	4.79	52.11± 5.42	6.20

Table 15: Results on GLUE with 32-shots.

Datasets	Models	Random-Bench Accuracy	Achilles-Bench (GradNorm) Accuracy	Achilles-Bench (GradNorm) Gap	Achilles-Bench (Loss) Accuracy	Achilles-Bench (Loss) Gap
SST2	Transformer	55.96± 1.33	52.50± 0.58	3.46	52.27± 0.48	3.69
	BERT	77.20± 4.97	54.70± 4.07	22.50	50.28± 0.85	26.92
	GPT-2	68.46± 5.20	57.73± 1.01	10.73	51.72± 0.72	16.74
	RoBERTa	83.81± 2.25	57.36± 2.45	26.45	50.09± 0.97	33.72
	T5	63.10± 3.97	54.06± 2.86	9.04	51.06± 2.38	12.04
COLA	Transformer	69.36± 0.37	69.13± 0.00	0.23	69.15± 0.04	0.21
	BERT	67.56± 3.19	62.05± 9.21	5.51	60.44± 10.81	7.12
	GPT-2	66.94± 3.91	66.06± 5.77	0.88	66.04± 6.22	0.90
	RoBERTa	72.23± 1.32	66.27± 3.90	5.96	59.85± 12.56	12.38
	T5	59.50± 4.25	58.39± 5.32	1.11	57.81± 6.12	1.69
MNLI	Transformer	35.46± 0.03	35.45± 0.00	0.01	35.42± 0.04	0.04
	BERT	39.21± 2.50	34.45± 0.76	4.76	33.41± 0.63	5.80
	GPT-2	39.81± 0.88	34.75± 1.47	5.06	34.00± 1.28	5.81
	RoBERTa	45.44± 2.02	36.55± 0.97	8.89	33.81± 1.18	11.63
	T5	34.45± 0.50	33.87± 0.24	0.58	33.13± 0.24	1.32
QNLI	Transformer	53.96± 0.92	51.36± 0.40	2.60	50.78± 0.09	3.18
	BERT	57.07± 2.02	50.85± 0.40	6.22	50.08± 0.35	6.99
	GPT-2	57.97± 2.83	53.70± 3.03	4.27	52.86± 2.50	5.11
	RoBERTa	71.64± 1.99	50.67± 0.62	20.97	49.61± 0.31	22.03
	T5	60.41± 4.20	50.74± 1.29	9.67	49.50± 1.04	10.91
MRPC	Transformer	68.63± 0.27	68.43± 0.10	0.20	68.48± 0.20	0.15
	BERT	66.96± 2.36	61.72± 7.05	5.24	54.80± 16.04	12.16
	GPT-2	69.07± 1.82	67.11± 2.33	1.96	63.97± 8.95	5.10
	RoBERTa	73.73± 2.79	55.34± 6.31	18.39	35.39± 4.57	38.34
	T5	62.21± 3.44	58.04± 5.53	4.17	55.78± 8.91	6.43
QQP	Transformer	64.06± 0.30	63.26± 0.09	0.80	63.18± 0.00	0.88
	BERT	65.49± 1.73	61.12± 1.72	4.37	54.83± 5.47	10.66
	GPT-2	63.37± 3.21	56.03± 4.17	7.34	55.00± 5.30	8.37
	RoBERTa	70.10± 0.98	61.82± 2.73	8.28	62.81± 0.45	7.29
	T5	61.44± 4.99	56.21± 2.96	5.23	54.53± 4.05	6.91
RTE	Transformer	53.72± 0.98	55.38± 0.49	-1.66	55.02± 0.74	-1.30
	BERT	55.02± 3.83	52.85± 1.96	2.17	49.46± 2.45	5.56
	GPT-2	58.77± 2.65	60.36± 2.96	-1.59	52.85± 3.15	5.92
	RoBERTa	57.76± 3.62	54.95± 2.37	2.81	52.78± 0.58	4.98
	T5	51.48± 1.13	52.06± 1.77	-0.58	49.17± 1.60	2.31
WNLI	Transformer	58.59± 2.76	56.34± 0.00	2.25	58.31± 1.44	0.28
	BERT	54.08± 3.94	54.93± 4.27	-0.85	55.21± 2.87	-1.13
	GPT-2	58.03± 2.25	57.46± 1.87	0.57	56.34± 1.99	1.69
	RoBERTa	56.62± 0.56	56.90± 1.13	-0.28	57.18± 1.44	-0.56
	T5	53.80± 3.92	57.18± 3.03	-3.38	53.24± 5.52	0.56

Table 16: Results on GLUE with 100-shots.

Datasets	Models	Random-Bench Accuracy	Achilles-Bench (GradNorm) Accuracy	Achilles-Bench (GradNorm) Gap	Achilles-Bench (Loss) Accuracy	Achilles-Bench (Loss) Gap
SST2	Transformer	59.50± 1.52	52.41± 0.64	7.09	51.74± 0.38	7.76
	BERT	86.22± 0.39	51.38± 1.86	34.84	49.33± 2.12	36.89
	GPT-2	83.00± 1.43	53.46± 1.76	29.54	51.22± 1.85	31.78
	RoBERTa	88.37± 1.08	51.93± 0.73	36.44	50.57± 0.75	37.80
	T5	83.35± 4.21	52.25± 1.54	31.10	51.01± 2.49	32.34
COLA	Transformer	69.19± 0.08	69.17± 0.05	0.02	68.78± 0.69	0.41
	BERT	74.84± 1.36	61.25± 7.11	13.59	57.09± 12.93	17.75
	GPT-2	66.62± 3.15	65.77± 5.94	0.85	64.60± 6.87	2.02
	RoBERTa	77.28± 1.09	62.84± 6.71	14.44	59.64± 5.95	17.64
	T5	75.24± 1.07	57.49± 5.44	17.75	56.72± 7.18	18.52
MNLI	Transformer	35.61± 0.22	35.25± 0.32	0.36	35.11± 0.41	0.50
	BERT	43.98± 2.71	34.74± 0.62	9.24	33.13± 0.60	10.85
	GPT-2	48.69± 1.80	34.77± 1.12	13.92	33.86± 1.24	14.83
	RoBERTa	61.44± 2.69	36.87± 1.00	24.57	33.64± 0.98	27.80
	T5	40.63± 4.32	34.27± 0.36	6.36	32.98± 0.29	7.65
QNLI	Transformer	56.23± 0.86	50.68± 0.15	5.55	50.54± 0.00	5.69
	BERT	63.82± 4.63	49.98± 1.01	13.84	47.26± 2.17	16.56
	GPT-2	62.52± 4.58	53.34± 2.93	9.18	51.80± 2.02	10.72
	RoBERTa	78.44± 2.05	50.08± 0.51	28.36	50.23± 0.46	28.21
	T5	73.75± 2.43	49.99± 1.01	23.76	48.91± 1.36	24.84
MRPC	Transformer	69.02± 0.69	68.43± 0.10	0.59	66.67± 3.43	2.35
	BERT	69.41± 0.95	58.77± 5.61	10.64	48.24± 11.18	21.17
	GPT-2	71.76± 1.91	65.34± 2.14	6.42	64.07± 5.23	7.69
	RoBERTa	77.16± 2.77	60.34± 4.15	16.82	41.47± 6.42	35.69
	T5	65.64± 1.33	57.79± 5.58	7.85	56.52± 5.77	9.12
QQP	Transformer	65.27± 0.61	63.45± 0.32	1.82	60.54± 1.67	4.73
	BERT	69.62± 1.85	60.11± 1.68	9.51	47.14± 5.08	22.48
	GPT-2	70.13± 2.28	55.04± 4.57	15.09	51.27± 6.85	18.86
	RoBERTa	75.33± 1.24	63.02± 1.13	12.31	48.10± 12.34	27.23
	T5	73.02± 1.75	57.54± 4.38	15.48	53.35± 4.40	19.67
RTE	Transformer	53.72± 0.90	56.10± 0.71	-2.38	53.07± 0.23	0.65
	BERT	54.66± 2.65	52.56± 1.06	2.10	47.44± 1.49	7.22
	GPT-2	59.35± 3.23	57.76± 2.33	1.59	51.26± 2.40	8.09
	RoBERTa	63.39± 2.49	53.94± 0.67	9.45	51.05± 2.19	12.34
	T5	53.72± 3.97	51.19± 1.59	2.53	48.59± 1.32	5.13

G Visualization

G.1 Selected Sentences by Achilles-Bench for NLP tasks

Table 17: Sentences of the set random selected on SST2. We sample randomly 10 examples for each label.

sentence	label
inconsistent , meandering , and sometimes dry plot	negative
made a great saturday night live sketch , but a great movie it is not	
an mtv , sugar hysteria ,	
was only	
it 's been 13 months and 295 preview screenings since i last walked out on a movie , but resident evil really earned my indignant , preemptive departure	
act weird	
humbuggery ...	
90 punitive minutes of eardrum-dicing gunplay , screeching-metal smashups , and flaccid odd-couple sniping .	
of screenwriting cliches that sink it faster than a leaky freighter	
sit still for two hours and change watching such a character , especially when rendered in as flat and impassive a manner as phoenix 's	
a smart , solid , kinetically-charged spy flick worthy of a couple hours of summertime and a bucket of popcorn	positive
great acting	
have ever seen , constantly pulling the rug from underneath us , seeing things from new sides , plunging deeper , getting more intense	
is a film in which the talent is undeniable	
come away with a greater knowledge of the facts of cuban music	
shows how deeply felt emotions can draw people together across the walls that might otherwise separate them .	
the crazy things that keep people going in this crazy life	
appeal to asian cult cinema fans and asiaphiles interested to see what all the fuss is about .	
potentially interesting	
thrusts the audience	

Table 18: Sentences of the set searched by Achilles-Bench (GradNorm) for SST2. We choose top 10 examples for each label.

sentence	label
is well below expectations .	negative
make it sting	
is well below expectations	
huge sacrifice	
best spent elsewhere	
few ‘ cool ’ actors	
laughably	
below is well below expectations .	
spare dialogue	
temperamental	
to winger fans who have missed her since 1995 's forget paris	positive
rocky and	
becomes compulsively watchable	
particularly balk , who 's finally been given a part worthy of her considerable talents	
balk , who 's finally been given a part worthy of her considerable talents	
clearly a manipulative film	
entertainingly nasty	
busts out of its comfy little cell	
fascinate me	
rediscovers his passion in life	

Table 19: Sentences of the set searched by Achilles-Bench (Loss) for SST2. We choose top 10 examples for each label.

sentence	label
a damn fine and a truly distinctive and a deeply pertinent film provides an invaluable service is an undeniably worthy and devastating experience gain the unconditional love she seeks unfolds as one of the most politically audacious films of recent decades from any country , but especially from france self-deprecating , biting and witty feature reasonably creative eighth-grader chilling tale noble end from sharing the awe in which it holds itself	negative
fails to have a heart , mind or humor of its own terminally bland , ’s not a brilliant piece of filmmaking after next spreads them pretty thin an admittedly middling film the movie is silly beyond comprehension , just a bunch of good actors flailing around in a caper that ’s neither original nor terribly funny , incoherence and sub-sophomoric he script is n’t up to the level of the direction an overcooked souffl	positive

G.2 Visualization of the "Average Examples (Random)"

Figure 8: Visualization of the set random selected on CIFAR-10. We sample randomly 50 examples for each label.

G.3 Visualization of the Searched "Hard Examples (GradNorm)"

Figure 9: Visualization of the set searched by Achilles-Bench (GradNorm) on CIFAR-10. We choose top 50 examples for each label.

G.4 Visualization of the Searched "Hard Examples (Loss)"

Figure 10: Visualization of the set searched by Achilles-Bench (Loss) on CIFAR-10. We choose top 50 examples for each label.