Lifelong Benchmarks: Efficient Model Evaluation in an Era of Rapid Progress

Ameya Prabhu[*] AMEYA@PRABHU.BE
University of Oxford

Vishaal Udandarao[*] VU214@CAM.AC.UK
University of Tuebingen

Philip H.S. Torr PHILIP.TORR@ENG.OX.AC.UK
University of Oxford

Matthias Bethge[†] MATTHIAS.BETHGE@UNI-TUEBINGEN.DE
University of Tuebingen

Adel Bibi[†] ADEL.BIBI@ENG.OX.AC.UK
University of Oxford

Samuel Albanie[†] SAMUEL.ALBANIE@GMAIL.COM
Google Deepmind

Abstract

Standardized benchmarks drive progress in machine learning. However, with repeated testing, the risk of overfitting grows as algorithms over-exploit benchmark idiosyncrasies. In our work, we seek to mitigate this challenge by compiling *ever-expanding* large-scale benchmarks called *Lifelong Benchmarks*. As exemplars of our approach, we create *Lifelong-CIFAR10* and *Lifelong-ImageNet*, containing (for now) 1.69M and 1.98M test samples, respectively. While reducing overfitting, lifelong benchmarks introduce a key challenge: the high cost of evaluating a growing number of models across an ever-expanding sample set. To address this challenge, we also introduce an efficient evaluation framework: *Sort & Search (S&S)*, which reuses previously evaluated models by leveraging dynamic programming algorithms to selectively rank and sub-select test samples, enabling cost-effective lifelong benchmarking. Extensive empirical evaluations across ∼31,000 models demonstrate that *S&S* achieves highly-efficient approximate accuracy measurement, reducing compute cost from 180 GPU days to 5 GPU hours (∼1000x reduction) on a single A100 GPU, with low approximation error. As such, lifelong benchmarks offer a robust, practical solution to the "benchmark exhaustion" problem.

Keywords: benchmarking, efficient model evaluation, dynamic benchmarks

1 Introduction

We are in the midst of a benchmark revolution. Datasets like ImageNet (Deng et al., 2009), MS-COCO (Lin et al., 2014), GLUE (Wang et al., 2018) and BigBench (Srivastava et al., 2022) have been instrumental in advancing machine learning research by providing standardised scenarios for comparing models.

However, over time, these static benchmarks have been exposed to many evaluations, each leaking cues about their test data and weakening their statistical power as tools of generalisation measurement (Ott et al., 2022; Mazumder et al., 2023; Kiela et al., 2021).

∗. equal contribution, ordering decided by coin flip.
†. equal advising

Fresh approaches must compete with a body of methods that have been highly tuned to such benchmarks, incentivising further overfitting if they are to compete (Bender et al., 2021; Beyer et al., 2021). This raises a critical question: *What function should such benchmarks serve?*

Towards Lifelong Benchmarks. The primary goal of the vision benchmarks considered in this work is to assess model performance on some task using data that is *representative of the visual world* (Torralba and Efros, 2011). For instance, the CIFAR10 (Krizhevsky et al., 2009) benchmark tested whether classifiers can distinguish between 10 categories, such as dogs and cats. Subsequent versions like CIFAR10.1 (Lu et al., 2020), CIFAR10.2 (Lu et al., 2020), CINIC10 (Darlow et al., 2018), and CIFAR10-W (Sun et al., 2023) introduced more challenging and diverse samples to evaluate the same objective of classifying 10 categories. Over time, however, thanks to repeated evaluation exposure from competing approaches, each individual benchmark diminishes in representativeness as overfitting occurs at both the individual method and research community level (Fang et al., 2023; Vishniakov et al., 2023). In this work, we aim to tackle this challenge by introducing two *Lifelong Benchmarks*: *Lifelong-CIFAR10* and *Lifelong-ImageNet*. These are ever-expanding pools of test samples that aim to restore the representativeness of benchmarks to the visual world (see Fig. 3) by preventing models from overfitting specifically to the biases of any subset benchmark.

Evaluation Cost. Our *Lifelong-CIFAR10* and *Lifelong-ImageNet* benchmarks contain 1.69 million and 1.98 million test samples, respectively. A challenge we face with this expanding dataset is the increasing cost of evaluation—it takes roughly 140 and 40 days to evaluate our current model set on *Lifelong-CIFAR10* and *Lifelong-ImageNet* respectively. Similar issues occur across various domains, especially in large-scale foundation model (Bommasani et al., 2021) evaluation. For instance, evaluating a single large language model (LLM) on the MMLU benchmark (Hendrycks et al., 2021b) (standard benchmark for evaluating LLMs) takes 24 hours on a consumer-grade GPU (Ilyas Moutawwakil, 2023). As models grow in complexity, lifelong testing will inevitably lead to a surge in evaluation costs when benchmarking a large set of increasingly expensive models against an ever-growing collection of test samples (Sardana and Frankle, 2023; Dehghani et al., 2021). *Can we reduce this evaluation cost while minimising the prediction error?*

Efficient Model Evaluation. We develop algorithms for efficient evaluation in lifelong benchmarks by drawing inspiration from computerized adaptive testing (CAT) (Van der Linden and Glas, 2000), which can generate exams like the GRE and SAT from an ever-expanding pool of questions. Unlike traditional tests where all questions must be answered, CAT adaptively sub-samples questions based on examinee responses. This approach efficiently gauges proficiency with far fewer questions, while maintaining assessment accuracy.

Similarly, in our lifelong benchmarking framework, we aim to evaluate the classification ability of new models without testing them on all samples, instead selecting a subset of samples to evaluate models. We propose a method named *Sort & Search (S&S)*, which reuses past evaluated models on a sample set through dynamic programming to enable efficient evaluation of new, incoming models. *S&S* operates by first ranking test samples by their difficulty, done efficiently by leveraging data from previous tests. It then uses these updated rankings to evaluate new models, streamlining the benchmarking process. This strategy enables efficient lifelong benchmarking, reducing the cost dramatically from a collective of 180 GPU days to 5 GPU hours on a single A100 GPU. This signifies a dramatic 1000x

reduction in inference costs compared to static evaluation on all samples, having the potential for large downstream impact. To summarize, our key contributions are: (1) we introduce and formalise *lifelong benchmarking* as a novel framework for robust, efficient model evaluation, (2) we curate two lifelong benchmarks: *Lifelong-CIFAR10* and *Lifelong-ImageNet*, consisting of 1.69M and 1.98M samples respectively, and (3) we propose a novel framework, *Sort & Search* for efficient model evaluation, reducing over 99.9% of computation costs on our lifelong benchmarks while accurately predicting sample-wise performance.

2 Lifelong Benchmarks: Curation

Considerations. We aim to establish lifelong benchmarking as a standard evaluation protocol in computer vision. To demonstrate this, we considered two popular datasets as our basis: CIFAR10 (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009). We chose them due to (1) their widespread adoption in prior art, (2) the diverse set of models trained on them, and (3) the presence of numerous dataset variants with the same set of labels, encompassing distribution shifts (Barbu et al., 2019), temporal variations (Shirali and Hardt, 2023), and adversarial samples (Hendrycks et al., 2021c). We describe the precise construction of our datasets below. See Table 1 for key statistics and a detailed breakdown.
Lifelong-CIFAR10. We combine 22 domains of different CIFAR10-like datasets comprising samples applied with synthetic distribution shifts, synthetic samples generated by diffusion models, and samples queried from different search engines using different colors and domains. We deduplicate our dataset and downsample all images to the standard CIFAR10 resolution of 32×32. Our final dataset consists of 1.69 million samples.
Lifelong-ImageNet. We source our test samples from ImageNet and its corresponding variants. Similar to *Lifelong-CIFAR10*, our benchmark is designed for increased sample diversity (43 unique domains) We include samples sourced from different web-engines and generated using diffusion models. Our final Lifelong-ImageNet contains 1.98 million samples.

3 Lifelong Benchmarks: Formulation, Challenges and Approach

We start by formalizing the objective of lifelong benchmarking. Assume we have an evaluation benchmark $\mathcal{D}_n$ containing n samples, $\mathcal{D}_n=\{(x_1, y_1), \ldots, (x_n, y_n)\}$[1]. Also assume we have evaluated a set of m models, $\mathcal{M}=\{f_1, f_2, \ldots, f_m\}$ on $\mathcal{D}_n$. Using these evaluations[2], we note two key problems in the lifelong benchmarking paradigm (illustrated in Fig. 4):

- for Δm new models, how can we *efficiently evaluate* them on the n current samples in the benchmark?
- for Δn new samples, how can we *efficiently insert* them into our current lifelong benchmark in a way which facilitates efficient evaluation?

Efficient Evaluation of Δm Models. Our first challenge is to efficiently estimate the accuracy of the Δm new models on the n samples. Since n can be prohibitively large, we aim to estimate accuracy by querying only a subset $\mathcal{D}_{n'}$ containing $n' \ll n$ samples. The goal is

1. x_is are the data samples and y_is are the ground-truth labels.
2. one is allowed to use features, logits, predictions or other aspects from the already evaluated models for sample-efficient accuracy prediction.

to evaluate new Δm models only on n' samples and estimate the sample-wise accuracy on all the other remaining samples with minimal cost. Formally, given Δm, we want to predict whether each sample is classified correctly, *i.e.*, generate predictions $\mathbf{A}_{pred} \in \{0,1\}^{\Delta m \times n}$.

Insertion of Δn Samples. Our second challenge arises when we get new samples—here our goal is to insert the new samples into our lifelong benchmark in a way that enables efficient future evaluations. For this, we have to estimate the difficulty of the Δn samples relative to current samples in the benchmark[3]. A simple way would be by estimating performance of all m models on the Δn samples. As before, our goal is to minimise prediction error while also minimising evaluation cost—we can do this by only querying a subset $\mathcal{M}_{m'}$ containing $m' \ll m$ models to estimate the sample-wise accuracy.

Given this formalism of the lifelong benchmarking problem, a natural question arises: *How can we reduce the evaluation cost while minimising the error in predictions?*

4 Efficient Benchmarking with *Sort & Search*

Taking inspiration from computerized adaptive testing (Van der Linden and Glas, 2000), we propose an efficient evaluation framework, *Sort & Search (S&S)*, consisting of two components: (1) Ranking test samples from the entire dataset pool according to their difficulty, *i.e.*, *Sort* and (2) Sampling a subset from the pool to predict performance on, *i.e.*, *Search*. We aim to solve the two key challenges that we noted in Section 3 with our framework. We now describe the objective and algorithms used in *S&S*.

4.1 Ranking by Sort

Setup. We recall that our lifelong benchmark pool consists of evaluations of m models on n samples. For our method, given each model f_i, $i \in \{1, .., m\}$, we use the binary accuracy prediction per sample, across all n samples obtaining $\mathbf{a}_i = [p_{i1}, p_{i2} \ldots, p_{in}]$. Here, $p_{ij} \in \{0, 1\}$ represents whether the model f_i classified the sample x_j correctly. Thus for m models and n evaluation samples, we construct a binary matrix $\mathbf{A} \in \{0,1\}^{m \times n}$ by row-wise stacking all the accuracy predictions $\mathbf{a}_i$ (see Fig. 4 left).

Goal. The goal of sort is to find the best global permutation matrix $\mathbf{P} \in \{0,1\}^{n \times n}$ such that $\mathbf{AP}$ permutes the columns of $\mathbf{A}$ so that we can rank samples from *easy* (all 1s) to *hard* (all 0s). We say this has a minimum distance from the optimal ranked accuracy prediction matrix $\mathbf{Y} \in \{0,1\}^{m \times n}$, formally defined as:

$$
\begin{aligned}
&\mathbf{P}^*, \mathbf{Y}^* = \text{argmin}_{\mathbf{P},\mathbf{Y}} \|\mathbf{AP} - \mathbf{Y}\|, \\
&s.t. \quad \mathbf{P} \in \{0,1\}^{n \times n}, \mathbf{P}\mathbf{1}_n = \mathbf{1}_n, \mathbf{1}_n^\top \mathbf{P} = \mathbf{1}_m, \\
&\text{if} \quad \mathbf{Y}_{ij} = 1, \text{ then } \mathbf{Y}_{ij'} = 1 \ \ \forall j' \leq j, \\
&\text{if} \quad \mathbf{Y}_{ij} = 0, \text{ then } \mathbf{Y}_{ij'} = 0 \ \ \forall j' \geq j.
\end{aligned}
\tag{1}
$$

The ranked accuracy prediction matrix $\mathbf{Y}$ applies a thresholding operator for every row in $\mathbf{Y}$. If the threshold for the i$^{\text{th}}$ row is k, then the i$^{\text{th}}$ row is of the form $[\mathbf{1}_k^\top, \mathbf{0}_{n-k}^\top]$ where $\mathbf{1}_k$ is a vector of all ones of size k and $\mathbf{0}_{n-k}$ is a zero vector of size $n - k$.

3. Here, there exists a notion of "difficult" which satisfies the property that if a sample x_i is easier than a sample x_j then at least equal number of models predict x_i correctly as the number of models predicting x_j correctly (Baldock et al., 2021).

Sorting by Sum. Considering elements column-wise, the difficulty of each sample (a column) is proportional to the number of 1s in that column, which indicates most models classify this sample correctly. Sorting by sum is detailed in Listing 1— intuitively, this algorithm sorts samples from easy (more 1s) to hard (less 1s) by sorting the sum array across rows per column. We call this method *Sorting by Sum*, which optimizes $\mathbf{P}$.

Optimizing P given Y. An intuitive question is: *How does one order samples which have equal difficulty, defined by having equal number of 1s?* We recursively order samples for each bucket of points $\tilde{n}$ where the sum is the same by considering thresholds obtained from $\mathbf{Y}$ which lie within this region. Let $\tilde{m}$ having their thresholds in this region. We can optimize the ranking by applying Sorting by sum algorithm only on the matrix $\tilde{\mathbf{A}} \in \{0,1\}^{\tilde{m}\times\tilde{n}}$. This does not change the $\mathbf{Y}$ for other samples whose thresholds do not lie in this area, thereby strictly improving the solution at each iteration. We provide the algorithm for two iterations for an illustration in Listing 1.

Optimizing Y given a P. We use the DP search algorithm (see Listing 1) for optimizing $\mathbf{Y}$ given a $\mathbf{P}$ here. We see in Equation 1 that $\mathbf{P}$ is binary. This makes finding the optimal $\mathbf{P}^*$ a NP-Hard problem (Yuan and Ghanem, 2016). We optimize Equation 1 by alternating between minimizing $\mathbf{P}$ and $\mathbf{Y}$, with the goal of finding the best solution $\mathbf{P}^*$. However, we show stationarity, that is with each alternating step, the Recursive sum algorithm further improves the solution, until it reaches a stationary point.

Theorem 1. ***Stationarity.*** *Recursive Sum eventually converges to a stable solution.*

Sorting by Confidence Sum. One can additional relax the constraint on $\mathbf{a}_i = [p_{i1}, p_{i2} \ldots, p_{in}]$ from $p_{ij} \in \{0,1\}$ to $p_{ij} \in [0,1]$, and use confidence of the ground truth class. This modification allows Sorting by Sum to be the best solution without needing re-ranking and could enable more sample efficient ranking.

4.2 Efficient Selection by Search

Given that we have found the best $\mathbf{P}^*$ in the sorting phase, we assume this ordering of difficulty of samples generalizes to new incoming models Δm. Hence, when we get Δm new models, we want to predict samplewise accuracies for each new model on each datapoint. Formally,

Goal: Search. Given the permutation matrix $\mathbf{P}^*$ and Δm new models, we want to generate a ranked accuracy prediction matrix $\mathbf{Y}_{pred} \in \{0,1\}^{\Delta m\times n}$ with a query budget $n' \ll n$.

We first restate that the constraints on $\mathbf{Y}$ in equation 1 imply a thresholding operator of index from $\{1,\ldots,n\}$ for every row in $\mathbf{Y}$, i.e. every model in Δm independently. Hence, we consider the problem of optimizing the rows $\mathbf{y}_{pred} \in \{0,1\}^{1\times n}$ separately here.

We now detail: (i) How do we quantify how good a ranked accuracy prediction vector $\mathbf{y}_{pred}$ is? and (ii) How to find the best ranked accuracy prediction vector $\mathbf{y}_{pred}$?

(i) How good are my predictions? Given a prediction vector $\mathbf{y}_{pred}$, we can compute the mean-absolute error $E(\mathbf{a}_{gt}, \mathbf{y}_{pred})$ given by the Hamming distance to the ground truth vector $\mathbf{a}_{gt} \in \{0,1\}^{1\times n}$, defined as:

$$E(\mathbf{a}_{gt}, \mathbf{y}_{pred}) = \|\mathbf{a}_{gt}\mathbf{P}^* - \mathbf{y}_{pred}\|_1 \tag{2}$$

However, we want to further take into account the fact that predictions on models with, for example, high accuracies, will necessarily agree with the ground truth often by chance alone (Geirhos et al., 2020). The agreement by chance is given by:

$$E_{rand}(\mathbf{a}_{\text{gt}},\mathbf{y}_{\text{pred}}) = \frac{\|\mathbf{a}_{\text{gt}}\|_1}{n}\frac{\|\ \mathbf{y}_{\text{pred}}\|_1}{n} + \left(1 - \frac{\|\mathbf{a}_{gt}\|_1}{n}\right)\left(1 - \frac{\|\ \mathbf{y}_{pred}\|_1}{n}\right) \tag{3}$$

The normalized agreement between any two vectors $\mathbf{a}$ and $\mathbf{y}$ is defined by the Cohen's Kappa (Cohen, 1960) given as:

$$\kappa(\mathbf{a},\mathbf{y}) = \frac{1 - E(\mathbf{a},\mathbf{y}) - E_{\text{rand}}(\mathbf{a},\mathbf{y})}{1 - E_{\text{rand}}(\mathbf{a},\mathbf{y})} \tag{4}$$

We measure both E and κ as sample-wise metrics in this work. Note that smaller E is better but higher κ is better.

(ii) How to get the optimal $\mathbf{y}_{pred}$? Our goal here is to generate the sample-wise prediction array $\mathbf{y} \in \{0,1\}^{1\times n}$. The selection task is to select the best n' observations. The optimization task is, given $\mathbf{a}' \in \{0,1\}^{1\times n'}$, to first generate $\mathbf{y}' \in \{0,1\}^{1\times n'}$. Subsequently, we project the threshold found in $\mathbf{y}'$ by index to obtain the full vector $\mathbf{y}$.

4.2.1 Selection Subtask

The selection task involves finding the best n' observations such that when we project the threshold found in $\mathbf{y}'$ by index to obtain the full vector $\mathbf{y}$, we minimize the error. We use the best one shot solution, which is to uniformly sample n' points across n, providing the algorithm in Listing 2.

While we note that there could be better iterative algorithms which could quickly find intervals (such as binary search), uniform sampling achieves surprisingly good results, with limited need for further improvement.

4.2.2 Optimization Subtask

We propose an algorithm based on dynamic programming, called DP Search, detailed in Listing 1. It computes the difference between number of 1s and number of 0s for each index based on previous index across the row. The optimal threshold is the maximum value in this array. For an input of size n', the dynamic programming approach reduces the time complexity from $\mathcal{O}(n'^2)$ to $\mathcal{O}(n')$. Furthermore, it is guaranteed to return the globally optimal solution, defined as:

Theorem 2. ***Optimality***. *For any given* $\mathbf{a}'$ *and* $\mathbf{P}^*$*, the DP Search algorithm returns a* $\mathbf{y}' \in \{0,1\}^{1\times n'}$ *which is a global minimum of* $E(\mathbf{a}',\mathbf{y}')$.

Until now, we discussed efficiently evaluating new models Δm. How do we approach the problem when we want to efficiently extend the benchmark, adding Δn new samples?

4.3 Efficient Insertion of New Samples

To add new samples into our lifelong benchmark efficiently, we have to estimate their "difficulty" with respect to the other samples in the benchmark. To efficiently determinine difficulty by only evaluating $m' \ll m$ models, a ranking over models is required to enable optimally sub-sampling a subset of m' models. Hence, the reader can notice that we can perform efficient insertion by following the same procedure, recast with the the same optimisation objectives as described in Eq. (1) by replacing A with A^T, and correspondingly changing dimensions of other vectors/matrices from $1 \times n$ to $1 \times m$.

5 Experiments

We first describe our experimental setup, then demonstrate empirical results on our two tasks: (1) *efficient evaluation of new models* and (2) *efficient estimation of new sample difficulties*. We next provide a comprehensive analysis over various design choices.

5.1 Experimental Details

Model Space. For *Lifelong-CIFAR10*, we use a total of 31,250 CIFAR-10 pretrained models sampled from the NATS-Bench-Topology-search space (Dong et al., 2021). For *Lifelong-ImageNet*, we use a set of 167 ImageNet-1K and ImageNet-21K pre-trained models, sourced primarily from timm (Wightman, 2019) and Imagenet-Testbed (Taori et al., 2020).

Model Evaluation Split. To study efficient evaluation of new models, we use all the samples but split the model set for the *Lifelong-CIFAR10* benchmark into a randomly selected subset of $6,000$ models for ordering the samples (*i.e., Sort*) and evaluate metrics on the rest $25,250$ models (*i.e., Search*). Similarly, for the *Lifelong-Imagenet* benchmark, we use 50 models for ordering the samples (*i.e., Sort*) and evaluate on 117 models (*i.e., Search*).

Sample Addition Split. To study efficient estimation of new sample difficulties on *Lifelong-CIFAR10*, we use all the models but hold-out the CIFAR-10W dataset for evaluation ($\sim$500,000 samples) and use the rest of the samples for ranking ($\sim$1.2 million samples).

Metrics. We measure errors between estimated predictions $\mathbf{Y}_{pred}$ and ground-truth predictions $\mathbf{A}_{gt}$ in a sample-wise fashion and over aggregate samples. For sample-wise predictions, we measure the mean-average error $E(\mathbf{a}_{gt}, \mathbf{y}_{pred})$ using Equation 2 along with normalized agreement κ using Equation 3. Additionally, we measure aggregate performance by the difference between estimated and ground truth accuracies by $E_{agg} = |(|\mathbf{y}_{pred}| - |\mathbf{a}_{gt}|)|/n$.

Design choices. We first provide an overview of alternatives possible for the *sorting* and *searching* stages in Table 2. For *sorting,* we benchmark three algorithms: (i) Sorting by Sum, (ii) Sorting by Recursive Sum, and (iii) Sorting by Confidence Sum. For *sampling* in the search process, we benchmark (i) uniform, and (ii) random sampling.

Unless otherwise specified, our main results in Sections 5.2 and 5.3 use the simple Sorting by Sum algorithm for obtaining P^*, and uniform sampling for the sample budget n'. We analyse the other design choices in Section 11.

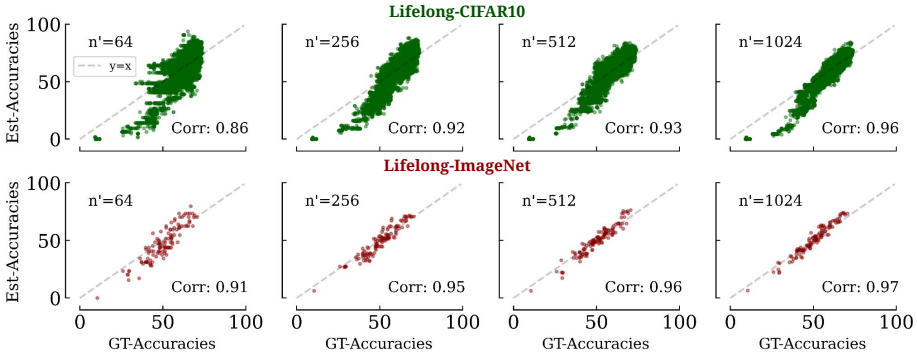
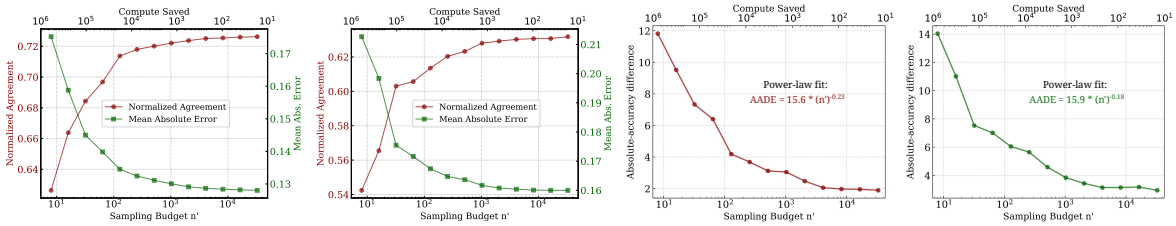


Figure 1: **Estimated v/s Ground-Truth accuracies.**



(a) *Lifelong-ImageNet* (b) *Lifelong-CIFAR10* (c) *Lifelong-ImageNet* (d) *Lifelong-CIFAR10*

Figure 2: **Main Results.**

5.2 Model Performance Estimation

In this set of experiments, we evaluate the predictive power of *S&S* when subjected to different sampling budgets n' *i.e.,* we run *S&S* over 13 different sampling budgets: {8, 16, 32, 64, 128, 256, ..., 32768} on both *Lifelong-ImageNet* and *Lifelong-CIFAR10*.

Key Result 1: Extreme cost-efficiency. From Figs. 2(a) and 2(b), we observe that our approach converges to a very low mean-absolute error and high normalized agreement for upto 1000x smaller total number of evaluation samples, leading to extreme cost savings at inference time. This consistently holds true across both datasets on all three metrics: Normalized Agreement, Mean Absolute Error and Absolute-accuracy difference.

Key Result 2: Prediction error scales as a power-law. We further analyse the observed E_{agg} v/s sampling budget relationship by fitting power-laws in Figs. 2(c) and 2(d): We discover that the power-laws have large exponential coefficients, further demonstrating the surprisingly high sample-efficiency obtained by *Sort & Search*.

Key Result 3: Highly accurate performance estimation. We note from Fig. 1 that our *S & S* method is able to very accurately predict the ground-truth accuracies of models. Note that this performance prediction ability is especially surprising given these results are aggregated over 25,250 models for *Lifelong-CIFAR10* and 117 models for *Lifelong-ImageNet*, spanning a wide range of architectures, model sizes and accuracies.

5.3 Sample Difficulty Estimation

Here, we showcase results with the transpose task where for new samples, the goal is to sub-sample the number of models to evaluate on the new samples, for accurately determining

sample difficulty. Section 5.3 showcases our results on this task on the *Lifelong-CIFAR10* benchmark with two different methods for ranking models[4], *Sum* and *Confidence Sum*. We evaluate over different model budgets (the number of models we use to evaluate our samples over): $\{8, 16, 32, 64, 128, 256, 512, 1024, 2048\}$. Both methods converge very quickly–the Sum method reaches an MAE of less than 0.15 by only evaluating on 64 models (10^4 times compute savings). This demonstrates our method's ability to efficiently determine sample difficulty, enabling efficient insertion back into the benchmark pool.

6 Conclusion

In this work, we introduced *Lifelong-Benchmarks*: a dynamically expanding pool of test samples designed to enhance the robustness of current benchmarks by mitigating the issue of overfitting to specific dataset biases. As two instances of this paradigm, we curated *Lifelong-CIFAR-10* and *Lifelong-ImageNet* containing over a million evaluation samples each. To counter the challenge of increasing evaluation costs on such large-scale benchmarks, we proposed an efficient framework called *Sort & Search* that leverages previous model predictions to rank and selectively evaluate test samples. Our extensive experiments, involving over 30,000 models, demonstrate that our method reduces over 99% of evaluation costs. We hope our *Lifelong Benchmarking* strategy spurs more robust and efficient evaluations.

References

Chirag Agarwal, Daniel D'souza, and Sara Hooker. Estimating example difficulty using variance of gradients. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Eslam Mohamed Bakr, Pengzhan Sun, Xiaogian Shen, Faizan Farooq Khan, Li Erran Li, and Mohamed Elhoseiny. Hrs-bench: Holistic, reliable and scalable benchmark for text-to-image models. In *International Conference on Computer Vision (ICCV)*, 2023.

Robert Baldock, Hartmut Maennel, and Behnam Neyshabur. Deep learning through the lens of example difficulty. *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Hritik Bansal and Aditya Grover. Leaving reality to imagination: Robust classification via generated datasets. *International Conference on Learning Representations Workshop (ICLR-W)*, 2023.

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Conference on Fairness, Accountability, and Transparency (FAccT)*, 2021.

Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Haoyang Bi, Haiping Ma, Zhenya Huang, Yu Yin, Qi Liu, Enhong Chen, Yu Su, and Shijin Wang. Quality meets diversity: A model-agnostic framework for computerized adaptive testing. In *International Conference on Data Mining (ICDM)*, 2020.

4. Recursive sum is not applicable here as all sum values are unique

Yonatan Bitton, Hritik Bansal, Jack Hessel, Rulin Shao, Wanrong Zhu, Anas Awadalla, Josh Gardner, Rohan Taori, and Ludwig Schimdt. Visit-bench: A benchmark for vision-language instruction following inspired by real-world use. *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Nitzan Bitton-Guetta, Yonatan Bitton, Jack Hessel, Ludwig Schmidt, Yuval Elovici, Gabriel Stanovsky, and Roy Schwartz. Breaking common sense: Whoops! a vision-and-language benchmark of synthetic and compositional images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2616–2627, 2023.

Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In *International Conference on Machine Learning (ICML)*, 2015.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Florian Bordes, Shashank Shekhar, Mark Ibrahim, Diane Bouchacourt, Pascal Vincent, and Ari S Morcos. Pug: Photorealistic and semantically controllable synthetic data for representation learning. *arXiv preprint arXiv:2308.03977*, 2023.

Samuel R Bowman and George E Dahl. What will it take to fix benchmarking in natural language understanding? In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Muxi Chen, Yu Li, and Qiang Xu. Hibug: On human-interpretable model debug. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 1960.

Ciprian A Corneanu, Sergio Escalera, and Aleix M Martinez. Computing the testing error without a testing set. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

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.

Mostafa Dehghani, Anurag Arnab, Lucas Beyer, Ashish Vaswani, and Yi Tay. The efficiency misnomer. *arXiv preprint arXiv:2110.12894*, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Greg d'Eon, Jason d'Eon, James R Wright, and Kevin Leyton-Brown. The spotlight: A general method for discovering systematic errors in deep learning models. In *Conference on Fairness, Accountability, and Transparency (FAccT)*, 2022.

Xuanyi Dong, Lu Liu, Katarzyna Musial, and Bogdan Gabrys. Nats-bench: Benchmarking nas algorithms for architecture topology and size. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding dataset difficulty with v-usable information. In *International Conference on Machine Learning (ICML)*, 2022.

Sabri Eyuboglu, Maya Varma, Khaled Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. Domino: Discovering systematic errors with cross-modal embeddings. *International Conference on Learning Representations (ICLR)*, 2022.

Alex Fang, Simon Kornblith, and Ludwig Schmidt. Does progress on imagenet transfer to real-world datasets? In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Wanyong Feng, Aritra Ghosh, Stephen Sireci, and Andrew S Lan. Balancing test accuracy and security in computerized adaptive testing. *International Conference on Artificial Intelligence in Education (AIED)*, 2023.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

Irena Gao, Gabriel Ilharco, Scott Lundberg, and Marco Tulio Ribeiro. Adaptive testing of computer vision models. In *International Conference on Computer Vision (ICCV)*, 2023.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. Evaluating models' local decision boundaries via contrast sets. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International Conference on Machine Learning (ICML)*, 2023.

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 (ICLR)*, 2018.

Robert Geirhos, Kristof Meding, and Felix A Wichmann. Beyond accuracy: quantifying trial-by-trial behaviour of cnns and humans by measuring error consistency. *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Aritra Ghosh and Andrew Lan. Bobcat: Bilevel optimization-based computerized adaptive testing. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *International Conference on Learning Representations (ICLR)*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *International Conference on Computer Vision (ICCV)*, 2021a.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *International Conference on Learning Representations (ICLR)*, 2021b.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021c.

Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *arXiv preprint arXiv:2306.14610*, 2023.

Zhenya Huang, Qi Liu, Chengxiang Zhai, Yu Yin, Enhong Chen, Weibo Gao, and Guoping Hu. Exploring multi-objective exercise recommendations in online education systems. In *International Conference on Information and Knowledge Management (CIKM)*, 2019.

Ben Hutchinson, Negar Rostamzadeh, Christina Greer, Katherine Heller, and Vinodkumar Prabhakaran. Evaluation gaps in machine learning practice. In *Conference on Fairness, Accountability, and Transparency (FAccT)*, 2022.

Régis Pierrard Ilyas Moutawwakil. Llm-perf leaderboard. https://huggingface.co/spaces/optimum/llm-perf-leaderboard, 2023.

Neel Jain, Khalid Saifullah, Yuxin Wen, John Kirchenbauer, Manli Shu, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Bring your own data! self-supervised evaluation for large language models. *arXiv preprint arXiv:2306.13651*, 2023.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. Text encoders are performance bottlenecks in contrastive vision-language models. *arXiv preprint arXiv:2305.14897*, 2023.

Gal Kaplun, Nikhil Ghosh, Saurabh Garg, Boaz Barak, and Preetum Nakkiran. Deconstructing distributions: A pointwise framework of learning. *International Conference on Learning Representations (ICLR)*, 2023.

Faisal Khan, Bilge Mutlu, and Jerry Zhu. How do humans teach: On curriculum learning and teaching dimension. *Advances in neural information processing systems*, 24, 2011.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. Dynabench: Rethinking benchmarking in nlp. *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.

Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. Active testing: Sample-efficient model evaluation. In *International Conference on Machine Learning (ICML)*, 2021.

Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Thomas Rainforth. Active surrogate estimators: An active learning approach to label-efficient model evaluation. *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

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

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision (IJCV)*, 128(7):1956–1981, 2020.

Tony Lee, Michihiro Yasunaga, Chenlin Meng, Yifan Mai, Joon Sung Park, Agrim Gupta, Yunzhi Zhang, Deepak Narayanan, Hannah Benita Teufel, Marco Bellagente, et al. Holistic evaluation of text-to-image models. *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Thomas Liao, Rohan Taori, Inioluwa Deborah Raji, and Ludwig Schmidt. Are we learning yet? a meta review of evaluation failures across machine learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

Shangyun Lu, Bradley Nott, Aaron Olson, Alberto Todeschini, Hossein Vahabi, Yair Carmon, and Ludwig Schmidt. Harder or different? a closer look at distribution shift in dataset reproduction. In *International Conference on Machine Learning Workshops (ICML-W)*, 2020.

Inbal Magar and Roy Schwartz. Data contamination: From memorization to exploitation. *arXiv preprint arXiv:2203.08242*, 2022.

Horia Mania, John Miller, Ludwig Schmidt, Moritz Hardt, and Benjamin Recht. Model similarity mitigates test set overuse. *Conference on Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Mark Mazumder, Colby Banbury, Xiaozhe Yao, Bojan Karlaš, William Gaviria Rojas, Sudnya Diamos, Greg Diamos, Lynn He, Alicia Parrish, Hannah Rose Kirk, et al. Dataperf: Benchmarks for data-centric ai development. *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Dena F Mujtaba and Nihar R Mahapatra. Multi-objective optimization of item selection in computerized adaptive testing. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1018–1026, 2021.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Simon Ott, Adriano Barbosa-Silva, Kathrin Blagec, Jan Brauner, and Matthias Samwald. Mapping global dynamics of benchmark creation and saturation in artificial intelligence. *Nature Communications*, 13(1):6793, 2022.

Letitia Parcalabescu, Michele Cafagna, Lilitta Muradjan, Anette Frank, Iacer Calixto, and Albert Gatt. Valse: A task-independent benchmark for vision and language models centered on linguistic phenomena. *arXiv preprint arXiv:2112.07566*, 2021.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

Yotam Perlitz, Elron Bandel, Ariel Gera, Ofir Arviv, Liat Ein-Dor, Eyal Shnarch, Noam Slonim, Michal Shmueli-Scheuer, and Leshem Choshen. Efficient benchmarking (of language models). *arXiv preprint arXiv:2308.11696*, 2023.

Momchil Peychev, Mark Niklas Müller, Marc Fischer, and Martin Vechev. Automated classification of model errors on imagenet. *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. Dynasent: A dynamic benchmark for sentiment analysis. *Dynasent: A dynamic benchmark for sentiment analysis*, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Inioluwa Deborah Raji, Emily M Bender, Amandalynne Paullada, Emily Denton, and Alex Hanna. Ai and the everything in the whole wide world benchmark. *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, 2019.

Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P Lalor, Robin Jia, and Jordan Boyd-Graber. Evaluation examples are not equally informative: How should that change nlp leaderboards? In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

Rebecca Roelofs, Vaishaal Shankar, Benjamin Recht, Sara Fridovich-Keil, Moritz Hardt, John Miller, and Ludwig Schmidt. A meta-analysis of overfitting in machine learning. *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Mark Rofin, Vladislav Mikhailov, Mikhail Florinskiy, Andrey Kravchenko, Elena Tutubalina, Tatiana Shavrina, Daniel Karabekyan, and Ekaterina Artemova. Vote'n'rank: Revision of benchmarking with social choice theory. *Annual Meeting of the Association for Computational Linguistics (EACL)*, 2022.

Nikhil Sardana and Jonathan Frankle. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws. *arXiv preprint arXiv:2401.00448*, 2023.

Zhelun Shi, Zhipin Wang, Hongxing Fan, Zhenfei Yin, Lu Sheng, Yu Qiao, and Jing Shao. Chef: A comprehensive evaluation framework for standardized assessment of multimodal large language models. *arXiv preprint arXiv:2311.02692*, 2023.

Ali Shirali and Moritz Hardt. What makes imagenet look unlike laion. *arXiv preprint arXiv:2306.15769*, 2023.

Ali Shirali, Rediet Abebe, and Moritz Hardt. A theory of dynamic benchmarks. *arXiv preprint arXiv:2210.03165*, 2022.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Xiaoxiao Sun, Xingjian Leng, Zijian Wang, Yang Yang, Zi Huang, and Liang Zheng. Cifar-10-warehouse: Broad and more realistic testbeds in model generalization analysis. *arXiv preprint arXiv:2310.04414*, 2023.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, 2022.

Yonglong Tian, Lijie Fan, Kaifeng Chen, Dina Katabi, Dilip Krishnan, and Phillip Isola. Learning vision from models rivals learning vision from data. *arXiv preprint arXiv:2312.17742*, 2023.

Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

Vishaal Udandarao, Max F Burg, Samuel Albanie, and Matthias Bethge. Visual data-type understanding does not emerge from scaling vision-language models. *arXiv preprint arXiv:2310.08577*, 2023.

Wim J Van der Linden and Cees AW Glas. *Computerized adaptive testing: Theory and practice*. Springer, 2000.

Kirill Vishniakov, Zhiqiang Shen, and Zhuang Liu. Convnet vs transformer, supervised vs clip: Beyond imagenet accuracy. 2023.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. Anchor points: Benchmarking models with much fewer examples. *arXiv preprint arXiv:2309.08638*, 2023.

Eric Wallace, Adina Williams, Robin Jia, and Douwe Kiela. Analyzing dynamic adversarial training data in the limit. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 202–217, 2022.

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

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Conference on Neural Information Processing Systems (NeurIPS)*, 2019a.

Hangyu Wang, Ting Long, Liang Yin, Weinan Zhang, Wei Xia, Qichen Hong, Dingyin Xia, Ruiming Tang, and Yong Yu. Gmocat: A graph-enhanced multi-objective method for computerized adaptive testing. In *Conference on Knowledge Discovery and Data Mining (KDD)*, 2023.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Conference on Neural Information Processing Systems (NeurIPS)*, 2019b.

Zan Wang, Hanmo You, Junjie Chen, Yingyi Zhang, Xuyuan Dong, and Wenbin Zhang. Prioritizing test inputs for deep neural networks via mutation analysis. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 397–409. IEEE, 2021.

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Olivia Wiles, Isabela Albuquerque, and Sven Gowal. Discovering bugs in vision models using off-the-shelf image generation and captioning. *arXiv preprint arXiv:2208.08831*, 2022.

Jingwei Yu, Mu Zhenyu, Jiayi Lei, Li'Ang Yin, Wei Xia, Yong Yu, and Ting Long. Sacat: Student-adaptive computerized adaptive testing. In *The Fifth International Conference on Distributed Artificial Intelligence*, 2023.

Ganzhao Yuan and Bernard Ghanem. Binary optimization via mathematical programming with equilibrium constraints. *arXiv preprint arXiv:1608.04425*, 2016.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*, 2023.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2022.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. The visual task adaptation benchmark. 2019.

Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, and Han-Jia Ye. Model spider: Learning to rank pre-trained models efficiently. *arXiv preprint arXiv:2306.03900*, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

Wangchunshu Zhou, Yan Zeng, Shizhe Diao, and Xinsong Zhang. Vlue: A multi-task multi-dimension benchmark for evaluating vision-language pre-training. In *International Conference on Machine Learning (ICML)*, 2022.

Yan Zhuang, Qi Liu, Zhenya Huang, Zhi Li, Shuanghong Shen, and Haiping Ma. Fully adaptive framework: Neural computerized adaptive testing for online education. In *Conference on Artificial Intelligence (AAAI)*, 2022.

Orr Zohar, Shih-Cheng Huang, Kuan-Chieh Wang, and Serena Yeung. Lovm: Language-only vision model selection. *arXiv preprint arXiv:2306.08893*, 2023.

7 Depiction of Lifelong Benchmarking

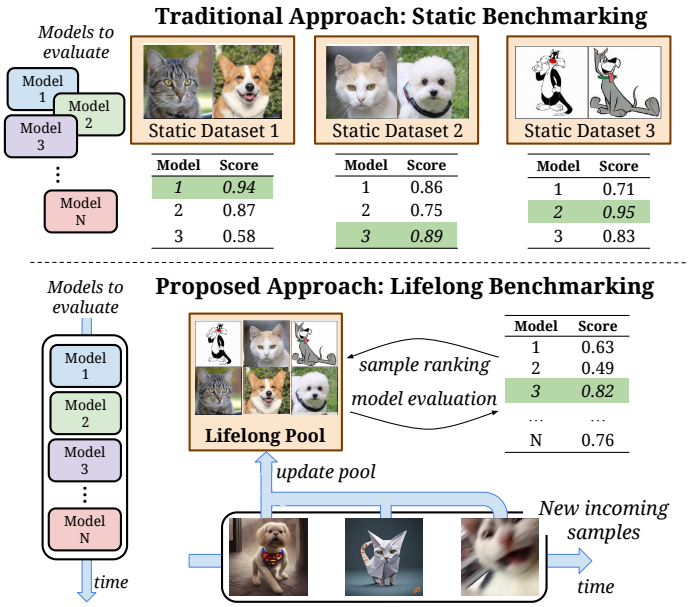


Figure 3: **Static vs Lifelong Benchmarking.** *(Top)* Static benchmarks incentivise machine learning practitioners to overfit models to specific datasets, weakening their ability to assess generalisation. *(Bottom)* We introduce *Lifelong Benchmarks* as an alternative paradigm—ever-expanding pools of test samples that resist overfitting while retaining computational tractability.

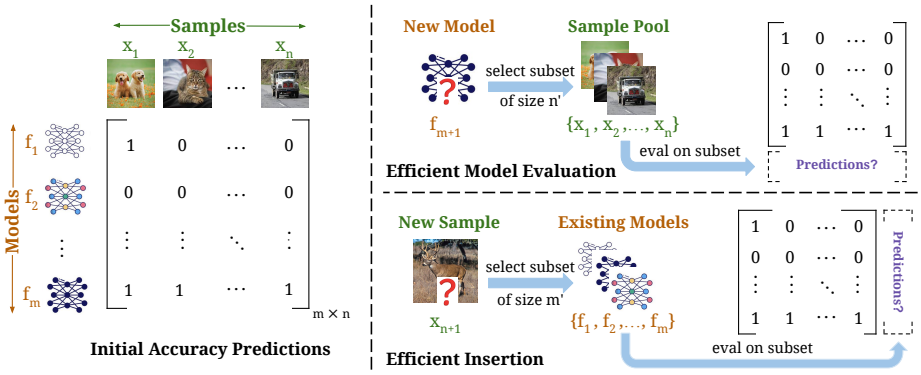


Figure 4: **Our proposed *Lifelong Benchmarking* setup.** We assume access to an initial pool of n samples and m models that have been evaluated on these samples *(left)*. Our goal is to efficiently evaluate a new model at sub-linear cost *(right top)* and efficiently insert a new sample into the lifelong benchmark by determining sample difficulty at sub-linear cost *(right bottom)*.

8 Lifelong Benchmarks: Overview

Table 1: **Overview of our Lifelong Benchmarks.** We list the constituent source datasets (deduplicated) and their statistics for constructing our lifelong benchmarks here. Our benchmarks encompass a wide-range of natural and synthetic domains, sources and distribution shifts, making for a comprehensive lifelong testbed.

Dataset	#Test Samples	#Domains	#Unique Sources	Synthetic/Natural	Corrupted/Clean
Lifelong-CIFAR10	1,697,682	22	9	Both	Both
CIFAR10.1 Recht et al. (2018)	2,000	1	1	Natural	Clean
CIFAR10 Krizhevsky et al. (2009)	10,000	1	1	Natural	Clean
CIFAR10.2 Lu et al. (2020)	12,000	1	1	Natural	Clean
CINIC10 Darlow et al. (2018)	210,000	1	1	Natural	Clean
CIFAR10-W Sun et al. (2023)	513,682	3	8	Both	Clean
CIFAR10-C Hendrycks et al. (2021b)	950,000	19	1	Natural	Corrupted
Lifelong-ImageNet	1,986,310	43	9	Both	Both
ImageNet-A Hendrycks et al. (2021c)	7,500	1	3	Natural	Clean
ObjectNet Barbu et al. (2019)	18,514	1	1	Natural	Clean
OpenImagesNet Kuznetsova et al. (2020)	23,104	1	1	Natural	Clean
ImageNet-V2 Recht et al. (2019)	30,000	1	1	Natural	Clean
ImageNet-R Hendrycks et al. (2021a)	30,000	13	1	Natural	Clean
ImageNet Deng et al. (2009)	50,000	1	1	Natural	Clean
Greyscale-ImageNet Taori et al. (2020)	50,000	1	1	Natural	Clean
StylizedImageNet Geirhos et al. (2018)	50,000	1	1	Synthetic	Corrupted
ImageNet-Sketch Wang et al. (2019b)	50,889	1	1	Natural	Clean
SDNet Bansal and Grover (2023)	98,706	19	1	Synthetic	Clean
LaionNet Shirali and Hardt (2023)	677,597	1	1	Natural	Clean
ImageNet-C Hendrycks and Dietterich (2019)	900,000	19	1	Natural	Corrupted

9 Algorithms in *Sort & Search*

```
def sort_by_sum(A):
    sum_ranking = A.sum(axis=0)
    order = np.flip(np.argsort(sum_ranking))
    return order

def two_stage_sort_by_sum(A, idx):
    #Step 1: Sum
    order = sort_by_sum(A)
    #Step 1: Search
    thresh = dp_search(A[:, order])

    #Iterate over bins
    bins_ordered = sum_bins[order]
    uniq_bins = np.unique(bins_ordered)

    for bin in uniq_bins:
        idx = np.nonzero(bins_ordered==bin)[0]
        bin_thresh = np.nonzero(np.all([[bins_ordered >= idx.min()],  [bins_ordered <= idx.max()]],
        ↪  axis=0))[1]
        At = A[thresh][:, order[idx]]
        #Step 2: Sum
        new_order = sort_by_sum(At)
        # Replace current ordering within new in bin
        order[idx] = order[idx[new_order]]
    return order
```

Listing 1: Sort Algorithms

```
def uniform_sampling(query, num_p):
    # idx -> num_p uniformly sampled points
    idx = np.arange(0, len(query),
            len(query)//num_p)[1:]
    return idx

def dp_search(query):
    # query is 1 x k (from a row of PA)
    # (k can be assigned := n, n', m, m')
    query[query==0] = -1
    cumsum = np.cumsum(query)
    idx = np.argmax(cumsum)
    return idx
```

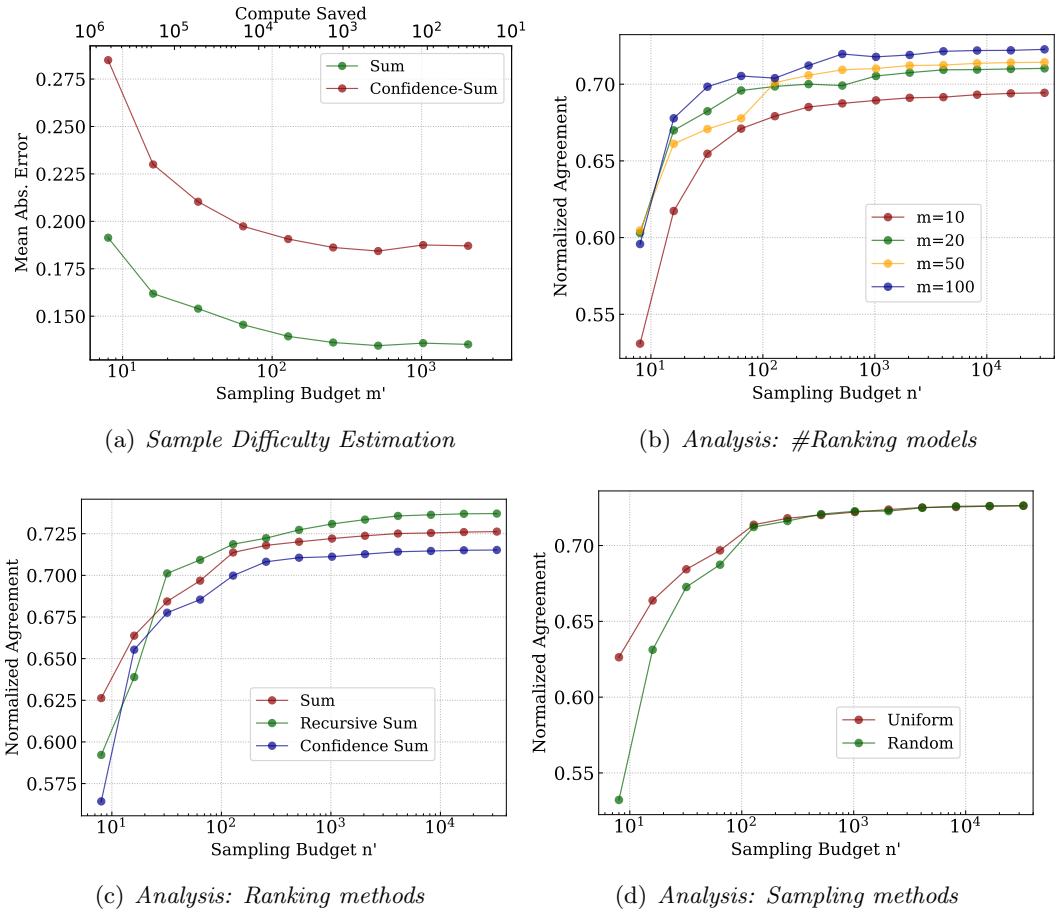
Listing 2: Search Algorithms

10 Design choices for experiments

Table 2: **Design Choices.** Alternatives for *Sorting & Searching*.

Sorting	**Searching**
Sum	Uniform
Confidence Sum	Random
Recursive Sum	

11 Breaking down *Sort & Search*



(a) *Sample Difficulty Estimation*

(b) *Analysis: #Ranking models*

(c) *Analysis: Ranking methods*

(d) *Analysis: Sampling methods*

Figure 5: **Additional Analyses.** *(a)* We achieve accurate sample difficulty estimates (<0.15 MAE) at a fraction of the total number of models to be evaluated, thereny enabling an efficient insertion of new samples into the ordered set of samples in the benchmark.*(b,c,d)* We analyse three of the design choice axes for gaining a better understanding of the *S&S* method.

Here, we analyse the different design choices used in our *S&S* framework, and compare their induced efficiency gains and accuracies.

Effect of number of models used for ranking. In Fig. 5(b), we analyse the effect of the number of models used for computing the initial ranking (*i.e.*, m) on the final performance prediction on *Lifelong-ImageNet*. Having access to more models seems to be a key factor in improving accuracy prediction power, since the *S & S* method using lower number of models for ranking (m=10) converges to a smaller normalised agreement. Interestingly, the m used for ranking does not have any effect on speed of convergence itself, but rather only on the predictive power.

Different ranking methods. On comparing the three different ranking methods used in our framework on *Lifelong-ImageNet* Fig. 5(c), we note no substantial benefits to using

the continual relaxation of the accuracy prediction values as confidence values, in fact, this degrades the predictive power of our method. However, using the multi-step recursive correction of rankings provides significant boosts due to its ability to locally correct ranking errors that the global sum method is unable to.

Different sampling methods. Finally, we compare the method used for sub-selecting the samples to evaluate on in Fig. 5(d), comparing between uniform and random sampling. Both methods converge very quickly and at similar budgets to their optimal values and start plateauing. Worth noting however is that uniform sampling provides large boosts over random sampling when the sampling budget is miniscule—this can be attributed to its "diversity-seeking" behaviour which helps cover samples from all difficulty ranges and hence better represent the entire benchmark evaluation samples than an unrepresentative random set.

Other Evaluations. We present additional evaluations such as decomposition of the mean absolute error into reducible and irreducible components, in the Appendix. Most of the error is induced due to generalization issues of the optimal ranking matrix $\mathbf{P}^*$ rather than sampling small subsets.

12 Decomposing the errors of *S&S*

The total mean absolute error $E(\mathbf{a}_{gt}, \mathbf{y}_{pred})$ can be decomposed into a component irreducible by further sampling, referred to as the Aleatoric Sampling Error ($E_{aleatoric}$), and a component which can be improved by querying larger fraction of samples n', referred to as the Epistemic Sampling Error ($E_{epistemic}$).

Let $\mathbf{y}^* = y'$ when $n' = n$, i.e. it is the global minima of errors across all subsampled thresholds. However, some error still remains between $\mathbf{y}^*$ and $\mathbf{a}_{gt}$. This error, caused by imperfect generalization of the permutation matrix $\mathbf{P}^*$ cannot be reduced by increasing the sample budget n'. More formally,

$$\begin{aligned} E_{aleatoric}(\mathbf{a}_{gt}, \mathbf{y}) &= \min_{\mathbf{y}} \|\mathbf{a}_{gt}\mathbf{P} - \mathbf{y}\| \\ &= \|\mathbf{a}_{gt}\mathbf{P} - \mathbf{y}^*\| \end{aligned} \tag{5}$$

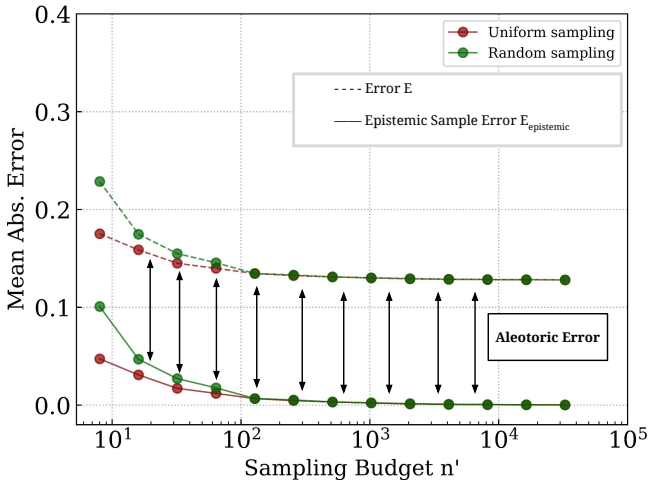


Figure 6: **Error Decomposition on *Lifelong-ImageNet***

On the contrary, the gap between the optimal ranking prediction $\mathbf{y}^*$ and $\mathbf{y}_{pred}$ is reducible by increasing sample size n'. This gap, referred to as Epistemic Sampling Error is formally defined as:

$$E_{aleatoric}(\mathbf{y}^*, \mathbf{y}_{pred}) = \|\mathbf{y}^* - \mathbf{y}_{pred}\| \tag{6}$$

Now, we can analyse the effectiveness of sampling in Lifelong CIFAR-10 and Lifelong-ImageNet by studying the Epistemic Sampling Error and Aleatoric Sampling Error (see Fig. 6). Note that this decomposition can be similarly defined for normalized agreement metric (κ) simply by $\kappa_{aleatoric}(\mathbf{a}_{gt}, \mathbf{y}^*)$ and $\kappa_{epistemic}(\mathbf{y}^*, \mathbf{y}_{pred})$

13 Correlation plots between estimated and ground-truth accuracies

In this section, we expand the Figure 3 from the main paper with more datapoints.

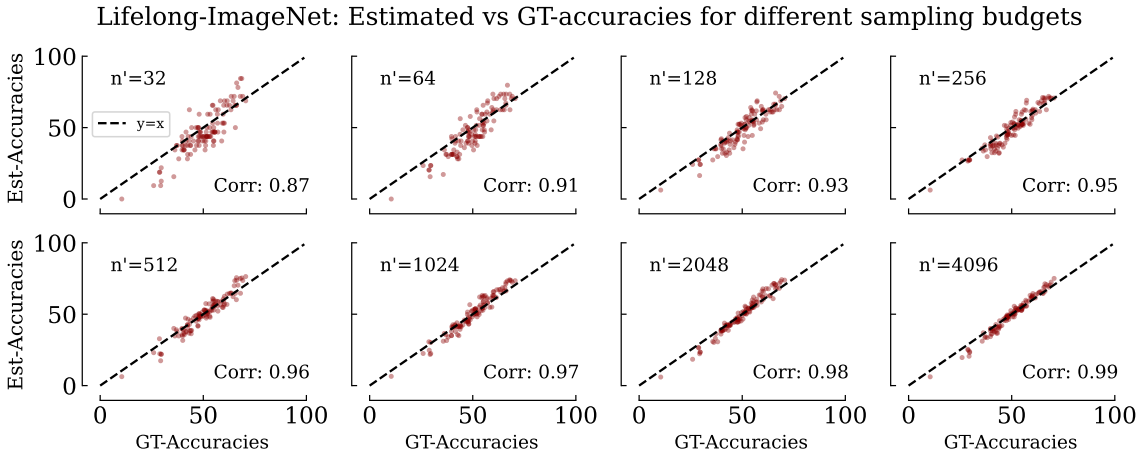


Figure 7: **Estimated v/s Ground-Truth accuracies on Lifelong-ImageNet.** For different sampling budgets($n' = 32 - 4096$), our estimated accuracies for 117 models are surprisingly close to the true ground-truth accuracies ($\rho = 0.94 - 1.0$).

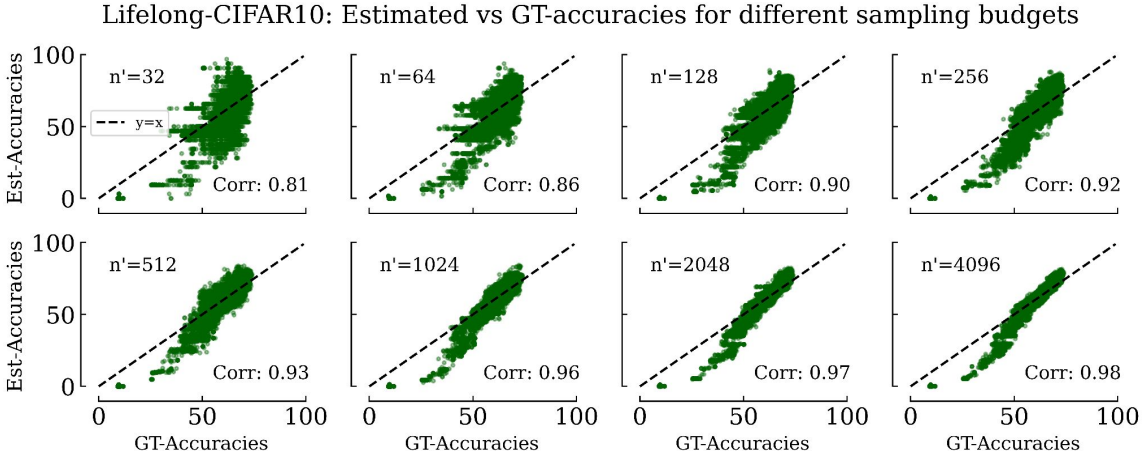


Figure 8: **Estimated v/s Ground-Truth accuracies on Lifelong-CIFAR10.** For different sampling budgets ($n' = 32 - 4096$), our estimated accuracies for 25,250 models are surprisingly close to the true ground-truth accuracies ($\rho = 0.81 - 0.98$).

14 Related Work

14.1 Closest relevant literature

While the lifelong benchmarking setup introduced here is quite unique, the sub-challenge of efficiently evaluating models has received limited attention. We comprehensively draw connections across different directions in the Appendix and briefly present the most similar works here. Model Spider (Zhang et al., 2023) efficiently ranks models from a pre-trained model zoo. LOVM (Zohar et al., 2023) and Flash-HELM (Perlitz et al., 2023) similarly rank foundation models efficiently on unseen datasets. However, these approaches only solve an easier task of predicting dataset-level accuracy and not predicting sample-level accuracies. Predicting sample-level accuracies is far harder as it involves not only calculating the average number of correct predictions but also accurately assigning the predictions to the full set of evaluation samples. Concurrent to our work, APS (Vivek et al., 2023) proposes an efficient sample-level evaluation by creating a core-set from the test data. However, their proposed method requires memory and time complexity of $\mathcal{O}(n^2)$ with the number of samples, preventing comparisons on datasets bigger than a few thousand samples. This is a far cry from our lifelong benchmarks having over 1.5 million test samples each. Our *Sort & Search* approach, in contrast, requires memory and time complexity of $\mathcal{O}(n \cdot logn)$ with the number of samples, and can scale up to billion-sized test sets.

14.2 Extended related works

Here, we expand on the brief literature review from above for a more expansive coverage of related topics.

Comprehensive Benchmarks. Benchmarking has become ubiquitous in the machine learning world in the last few years (Raji et al., 2021). It has gained further traction in the recent past with the release of foundation models like GPT-4 (Bubeck et al., 2023) and CLIP (Radford et al., 2021). A popular direction taken by efforts like GLUE (Wang et al., 2018), BigBench (Srivastava et al., 2022), HELM (Liang et al., 2022) *etc.* is to have a benchmark of benchmarks, reporting the average accuracy over the constituent datasets. This approach now spans across several domains including fact-based question-answering (Hendrycks et al., 2021b), language understanding (Wang et al., 2019a), zero-shot classification of vision-language models (Gadre et al., 2023), large-scale vision model evaluation (Zhai et al., 2019), multi-modal model evaluation (Yue et al., 2023; Zhou et al., 2022), and text-to-image generation (Bakr et al., 2023; Lee et al., 2023). Despite these benchmarks having vast coverage of testing concepts, the obvious downsides are two-fold: (1) they are static in nature and hence can always be susceptible to test-set contamination (Magar and Schwartz, 2022), and (2) their large sizes renders them very expensive resources to run full model evaluations on.

Adversarial Dynamic Benchmarks. One necessary aspect essential for lifelong benchmarks is collecting harder samples, which has been pursued by two strands of works. Adversarial methods to augment benchmarks (Wallace et al., 2022; Nie et al., 2020; Kiela et al., 2021; Potts et al., 2021; Shirali et al., 2022) aim to automatically curate samples that all tested models reliably fail on. These methods usually involve an iterative optimisation procedure to find such adversarial samples. The second strand of work in curating adversarial samples are efforts revolving around red-teaming (Ganguli et al., 2022; Perez et al., 2022) that aim to

explicitly elicit certain sets of behaviours from foundation models; primarily these approaches look at the problem of adversarial benchmarking from a safety perspective. Further, a host of benchmarks that aim to stress-test models are making their way on the horizon—their primary goal is to create test sets for manually discovered failure modes (Yuksekgonul et al., 2022; Parcalabescu et al., 2021; Thrush et al., 2022; Udandarao et al., 2023; Hsieh et al., 2023; Kamath et al., 2023; Bitton-Guetta et al., 2023; Bordes et al., 2023). However, while they are sample efficient, they are criticized as unfair. To mitigate this, a strand of automatic error discovery (Chen et al., 2023; Eyuboglu et al., 2022; Wiles et al., 2022; Peychev et al., 2023) or their human-in-the-loop variants (Wang et al., 2021; d'Eon et al., 2022; Gao et al., 2023) have been developed. This is complementary to our work, as we primarily explore model testing.

Active Testing. Efforts such as (Kossen et al., 2021, 2022) aim to identify "high-quality", representative test instances from a large amount of unlabeled data, which can reveal more model failures with less labeling effort. The key assumption underlying these works is that they assume access to a host of unlabeled data at a relatively cheap cost. However, they assume that the cost of label acquisition is a bottleneck. However, these assumptions can break down when doing multiple forward passes on a single batch of data with a large-scale foundation model is necessitated. Albeit similar in spirit to the task of actively acquiring a subset of samples for testing models, an important distinction of our method is that we want to minimise the number of forward-passes through a model—we believe that the cost of running a model on several test samples is substantial, and hence needs to be reduced for efficient evaluation in terms of time, resources and capital.

Ideas for Replacing Benchmarks. Recently, there have been a surge of methods introducing creative ways of benchmarking models (Liao et al., 2021; Roelofs et al., 2019; Kaplun et al., 2023; Gardner et al., 2020; Rodriguez et al., 2021; Rofin et al., 2022; Mania et al., 2019; Hutchinson et al., 2022; Bowman and Dahl, 2021; Tian et al., 2023; Ott et al., 2022; Garrido et al., 2023; Roelofs et al., 2019; Rodriguez et al., 2021) including hosted competitions (Blum and Hardt, 2015), self-supervised evaluation (Jain et al., 2023) and newer metrics (Geirhos et al., 2020). Further, recently ELO style methods have been gaining a lot of attention (Bitton et al., 2023; Zheng et al., 2023) due to their scalability of deployment to millions of users in a peer-to-peer manner. The ELO algorithm is used to compute ranks for different models based on human-in-the-loop preferences. However, despite its utility ELO is heavily dependent on the choice of user inputs and can be a very biased estimator of model rankings (Shi et al., 2023). Another interesting idea proposed by (Corneanu et al., 2020) is to assume access to the pre-training data of models and compute topological maps to give predictions of test error; this however requires running expensive forward passes over the training data or modifying the training protocol, which might be not be scalable to pre-trained models.

Computerized Adaptive Testing. Computerized Adaptive Testing (CAT) is a framework that allows for efficient testing of human examinees. The idea is to lower the burden of students taking tests by only asking them a subset of questions from the entire pool. There have been few main directions of solutions: model-agnostic strategies for selection (Bi et al., 2020), bi-level optimization (Ghosh and Lan, 2021; Zhuang et al., 2022; Feng et al., 2023), multi-objective optimization (Mujtaba and Mahapatra, 2021; Huang et al., 2019; Wang et al., 2023), retrieval-augmented adaptive search (Yu et al., 2023). One key

challenge in CAT is the lack of a stable ground-truth. Since the goal in CAT is to estimate the proficiency of an examinee, and the examinee's true ground-truth proficiency is not provided, how would one evaluate the true proficiency of an examinee? Thereby, existing CAT methods cannot explicitly optimise for predicting ability directly *i.e.* they cannot do exact ability estimation. Hence, CAT methods are not usually guaranteed to converge to the true examinee abilities under certain conditions. The biggest distinction of our work from CAT is the access to the ground-truth targets for the tasks we consider. In both *Lifelong-ImageNet* and *Lifelong-CIFAR10*, we have access to the ground-truth and hence can compute grounded metrics that can be optimised towards, unlike in CAT, where every method has to inherently be label-free.

Curriculum Learning. This refers to the problem of finding a curriculum of input samples such that the optimisation objective of an algorithm becomes easier. The most intuitive explanation from curriculum learning comes from how humans learn (Khan et al., 2011). In the context of machine learning, the idea behind curriculum learning is to find the "difficulty" of samples, where difficulty is usually defined in terms of the ease of classifying that sample correctly. Some recent works in this direction utilise estimating variance of gradients (Agarwal et al., 2022) and other information theoretic properties (Ethayarajh et al., 2022) to estimate sample difficulty. These approaches are complementary to our *Sum* component in *S&S* since these can be easily integrated into our framework directly.

15 Proof of Theorem 4.1

Proof Let us restate the problem. We first say minimizing

$$\begin{aligned} &\min_{\mathbf{P},\mathbf{Y}} \|\mathbf{AP} - \mathbf{Y}\|, \\ &\text{s.t. } \mathbf{P} \text{ is a permutation matrix,} \\ &\mathbf{Y}_{ij} \text{ is such that if } \mathbf{Y}_{ij} = 1 \text{ then } \mathbf{Y}_{ij'} = 1 \forall j \leq j \\ &\text{and if } \mathbf{Y}_{ij} = 0 \text{ then } \mathbf{Y}_{ij'} = 1 \forall j \geq j \end{aligned} \tag{7}$$

is equivalent to the following problem:

$$\begin{aligned} &\min_{\mathbf{P},\mathbf{X}} \|\mathbf{AP} - \mathbf{X}\Psi\|, \\ &\text{s.t. } \mathbf{P} \in \{0,1\}^{m \times m}, \mathbf{P}\mathbf{1}_m = \mathbf{1}_m, \mathbf{1}_m^\top \mathbf{P} = \mathbf{1}_m \\ &\mathbf{X}\mathbf{1} = \mathbf{1}_m, \mathbf{X} \in \{0,1\}^{m \times n}, \end{aligned} \tag{8}$$

where $\Psi \in 0, 1^{n \times n}$ such that it $\Psi_{i:} = [\mathbf{1}_i^\top, \mathbf{0}_{n-i}^\top]$.

Note that $\mathbf{1}_i^\top$ and $\mathbf{0}_{n-i}^\top$ are a row vectors of ones and zeros with sizes i and $n-i$, respectively. Moreover, note that the constraints on $\mathbf{P}$ that it has to be binary and that the rows and the columns independently sum to 1 enforces $\mathbf{P}$ to be doubly stochastic and hence a permutation matrix. At last observe that for any choice of indexing for $\mathbf{Y}$, we have that there $\exists$ $\mathbf{X}$ such that $\mathbf{X}\Psi = \mathbf{Y}$.

To solve Equation (1), one can iterate between updating the permutation matrix for the data $\mathbf{A}$ for a given thresholding operator $\mathbf{X}$, and then iterating back on the thresholding function $\mathbf{X}$ for a given permutation matrix. The general solver $f(X, P)$ has the following form :

$$\begin{cases} \mathbf{X}^{k+1} = \text{argmin}_{\mathbf{X}} \|\mathbf{A}\mathbf{P}^k - \mathbf{X}\Psi\|, \\ \qquad \text{s.t. } \mathbf{X}\mathbf{1} = \mathbf{1}_m, \mathbf{X} \in \{0,1\}^{m \times n} \\ \mathbf{P}^{k+1} = \text{argmin}_{\mathbf{P}} \|\mathbf{A}\mathbf{P} - \mathbf{X}^{k+1}\Psi\|, \\ \qquad \text{s.t. } \mathbf{P} \in \{0,1\}^{m \times m}, \mathbf{P}\mathbf{1}_m = \mathbf{1}_m, \mathbf{1}_m^\top \mathbf{P} = \mathbf{1}_m \end{cases}$$

We notice that for any k,

$$\begin{aligned} f(X^{k+1}, P^k) &\leq f(X^k, P^k) \\ f(X^{k+1}, P^{k+1}) &\leq f(X^{k+1}, P^k) \end{aligned} \tag{9}$$

Hence, $f(X^{k+1}, P^{k+1}) \leq f(X^k, P^k), \forall k$ i.e. the general solver $f(X, P)$ forms a monotonically non-increasing function and therefore the sequence has a limit f^* at $k = \infty$. Hence, this function converges. ■

16 Proof of Theorem 4.2

Proof First, using the same decomposition as Equation 6, we reduce the theorem problem to the following:

$$\mathbf{y'}^* = \text{argmin}_{\mathbf{y'}} \|\mathbf{a'}\mathbf{P}^* - \mathbf{y'}\| \tag{10}$$

Note that $\mathbf{y'}$ essentially constructs a vector of all ones up to some index with the rest where $\mathbf{x}$ is nonzero with the rest being zero. Therefore, $\mathbf{y'}_i$ is a vector of all ones up to index i with the rest being zero. Let $\mathbf{b} = \mathbf{a'}\mathbf{P}^*$ be the sorted vector according to the permutation matrix. Thus, the objective function has the following error:

$$\mathbf{e}(\mathbf{y'}_i) = \left(i - \sum_{k=1}^{i} \mathbf{b}_k \right) + \sum_{k=i+1}^{n} \mathbf{b}_k. \tag{11}$$

Observe that the first term is the number of zeros to the left of index i (inclusive) in $\mathbf{b}$, while the second term is the number of 1s in $\mathbf{b}$ to the right of index i.

Proposition 3. *If $\mathbf{y'}_i$ is a minimizer to Theorem 4.2, then, the following holds:*

$$\sum_{k=i+1}^{n} \mathbf{b}_k \leq (n - i) - \sum_{k=i+1}^{n} \mathbf{b}_j.$$

Proof Let $j < i$ and that $\mathbf{y'}_i$ and $\mathbf{y'}_j$ are feasible solutions for Theorem 4.2. However, let that $\mathbf{y'}_i$ be such that the inequality in Proposition 3 while it is not the case for $\mathbf{y'}_j$. Then, we compare the differences in the objective functions $\mathbf{e}(\mathbf{y'}_i)$ and $\mathbf{e}(\mathbf{y'}_j)$. We have that:

$$\begin{aligned} \mathbf{e}(\mathbf{y'}_j) - \mathbf{e}(\mathbf{y'}_i) &= \left[\left(j - \sum_{k=1}^{j} \mathbf{b}_j \right) + \sum_{k=j+1}^{n} \mathbf{b}_k \right] - \left[\left(i - \sum_{k=1}^{i} \mathbf{b}_k \right) + \sum_{k=i+1}^{n} \mathbf{b}_k \right] \\ &= 2 \sum_{k=j+1}^{i} \mathbf{b}_k - (i - j). \end{aligned}$$

However, we know from the assumptions that $2\sum_{i+1}^{n} \mathbf{b}_k \leq n - i$ and that $2\sum_{j+1}^{n} \mathbf{b}_k \geq n - j$. Subtracting the two inequalities we have $2\sum_{k=j+1}^{n} \mathbf{b}_k \geq i - j$ which implies that $\mathbf{y}'(\mathbf{s}_j) \geq \mathbf{e}(\mathbf{y'}_i)$ which implies that $\mathbf{y'}_i$ is a better solution to any other $\mathbf{y'}_j$ not satisfying the inequality in Proposition 3. ■

The inequality condition in proposition 3, implies that for the choice of index i, the number of zeros in $\mathbf{a}$ to the right of index i is more than the number of 1s to the right of of index i. Since any solution, i.e. $\mathbf{y'}_i$ or in general thresholding index i, either statisfies property in proposition 3 or not, and since proposition demonstrated that the set of indices that satisfy this property are better, in objective value (lower), than all those that do not satisfy it, then this condition achieves optimality. ■

17 167 Models used for Lifelong-ImageNet experiments

We use the following models (as named in the timm (Wightman, 2019) and ImageNet-Testbed (Taori et al., 2020) repositories):

1. BiT-M-R101x3-ILSVRC2012
2. BiT-M-R50x1-ILSVRC2012
3. BiT-M-R50x3-ILSVRC2012
4. FixPNASNet
5. FixResNet50
6. FixResNet50CutMix
7. FixResNet50CutMix_v2
8. FixResNet50_no_adaptation
9. FixResNet50_v2
10. alexnet
11. alexnet_lpf2
12. alexnet_lpf3
13. alexnet_lpf5
14. bninception
15. bninception-imagenet21k
16. cafferesnet101
17. densenet121
18. densenet121_lpf2
19. densenet121_lpf3
20. densenet121_lpf5
21. densenet161
22. densenet169
23. densenet201
24. dpn107
25. dpn131
26. dpn68
27. dpn68b
28. dpn92
29. dpn98
30. efficientnet-b0
31. efficientnet-b0-autoaug
32. efficientnet-b1
33. efficientnet-b1-advprop-autoaug
34. efficientnet-b1-autoaug
35. efficientnet-b2
36. efficientnet-b2-advprop-autoaug
37. efficientnet-b2-autoaug
38. efficientnet-b3
39. efficientnet-b3-advprop-autoaug
40. efficientnet-b3-autoaug
41. efficientnet-b4
42. efficientnet-b4-advprop-autoaug
43. efficientnet-b4-autoaug
44. efficientnet-b5
45. efficientnet-b5-advprop-autoaug
46. efficientnet-b5-autoaug
47. efficientnet-b5-randaug
48. efficientnet-b6-advprop-autoaug
49. efficientnet-b6-autoaug
50. efficientnet-b7-advprop-autoaug
51. efficientnet-b7-autoaug
52. efficientnet-b7-randaug
53. efficientnet-b8-advprop-autoaug
54. fbresnet152
55. inceptionresnetv2
56. inceptionv3
57. inceptionv4
58. instagram-resnext101_32x16d
59. instagram-resnext101_32x32d
60. instagram-resnext101_32x8d

61. mnasnet0_5
62. mnasnet1_0
63. mobilenet_v2
64. mobilenet_v2_lpf3
65. mobilenet_v2_lpf5
66. nasnetalarge
67. nasnetamobile
68. polynet
69. resnet101
70. resnet101_cutmix
71. resnet101_lpf2
72. resnet101_lpf3
73. resnet101_lpf5
74. resnet152
75. resnet18
76. resnet18-rotation-nocrop_40
77. resnet18-rotation-random_30
78. resnet18-rotation-random_40
79. resnet18-rotation-standard_40
80. resnet18-rotation-worst10_30
81. resnet18-rotation-worst10_40
82. resnet18_lpf2
83. resnet18_lpf3
84. resnet18_lpf5
85. resnet18_ssl
86. resnet18_swsl
87. resnet34
88. resnet34_lpf2
89. resnet34_lpf3
90. resnet34_lpf5
91. resnet50
92. resnet50_adv-train-free
93. resnet50_augmix
94. resnet50_aws_baseline
95. resnet50_cutmix
96. resnet50_cutout
97. resnet50_deepaugment
98. resnet50_deepaugment_augmix
99. resnet50_feature_cutmix
100. resnet50_l2_eps3_robust
101. resnet50_linf_eps4_robust
102. resnet50_linf_eps8_robust
103. resnet50_lpf2
104. resnet50_lpf3
105. resnet50_lpf5
106. resnet50_mixup
107. resnet50_ssl
108. resnet50_swsl
109. resnet50_trained_on_SIN
110. resnet50_trained_on_SIN_and_IN
111. resnet50_trained_on_SIN_and_IN_then_finetuned_on_IN
112. resnet50_with_brightness_aws
113. resnet50_with_contrast_aws
114. resnet50_with_defocus_blur_aws
115. resnet50_with_fog_aws
116. resnet50_with_frost_aws
117. resnet50_with_gaussian_noise_aws
118. resnet50_with_greyscale_aws
119. resnet50_with_jpeg_compression_aws
120. resnet50_with_motion_blur_aws
121. resnet50_with_pixelate_aws
122. resnet50_with_saturate_aws
123. resnet50_with_spatter_aws
124. resnet50_with_zoom_blur_aws
125. resnext101_32x16d_ssl
126. resnext101_32x4d
127. resnext101_32x4d_ssl
128. resnext101_32x4d_swsl
129. resnext101_32x8d
130. resnext101_32x8d_ssl
131. resnext101_32x8d_swsl
132. resnext101_64x4d
133. resnext50_32x4d
134. resnext50_32x4d_ssl
135. resnext50_32x4d_swsl
136. se_resnet101
137. se_resnet152
138. se_resnet50
139. se_resnext101_32x4d
140. se_resnext50_32x4d
141. senet154
142. shufflenet_v2_x0_5
143. shufflenet_v2_x1_0
144. squeezenet1_0
145. squeezenet1_1
146. vgg11
147. vgg11_bn
148. vgg13
149. vgg13_bn
150. vgg16
151. vgg16_bn
152. vgg16_bn_lpf2
153. vgg16_bn_lpf3
154. vgg16_bn_lpf5
155. vgg16_lpf2
156. vgg16_lpf3
157. vgg16_lpf5
158. vgg19
159. vgg19_bn
160. wide_resnet101_2
161. xception
162. resnet50_imagenet_subsample_1_of_16_batch64_original_images
163. resnet50_imagenet_subsample_1_of_2_batch64_original_images
164. resnet50_imagenet_subsample_1_of_32_batch64_original_images
165. resnet50_imagenet_subsample_1_of_8_batch64_original_images
166. resnet50_with_gaussian_noise_contrast_motion_blur_jpeg_compression_
167. resnet50_imagenet_100percent_batch64_original_images

18 Limitations and Future Directions

Although showcasing very promising results in enhancing the efficiency of evaluating Lifelong Benchmarks, *S&S* faces certain key limitations: (1) *One-Step Process*: Currently, our approach is restricted to one-step sample ranking and model evaluation, whereas ideal lifelong evaluation would need simultaneous optimization of these steps. (2) *Ranking Imprecision*: Our error decomposition analysis in the Appendix suggests that the ordering of samples while evaluating new models is the bottleneck in reducing prediction errors. Generalizable sample ordering is a complex task, with potential biases and a lack of representation across diverse scenarios. (3) *Identifying Difficult Samples*: Finding and labeling challenging examples is an essential task for lifelong benchmarks, which is not investigated completely in this work. Studying adversarial sample selection approaches with lifelong benchmarking is a promising direction.