Exploring Learning Complexity for Efficient Downstream Dataset Pruning

Wenyu Jiang[1, 2] Zhenlong Liu[1] Zejian Xie[1] Songxin Zhang[1]
Bingyi Jing[1] Hongxin Wei[1]

[1]Department of Statistics and Data Science, Southern University of Science and Technology

[2]State Key Laboratory for Novel Software Technology, Nanjing University

Outline

1. Background
   - Prohibitive Model Development Costs from Neural Scaling Laws
   - Beating Neural Scaling Laws via Dataset Pruning
   - Concerns of Efficiency and Effectiveness
2. Method
   - DLC: Distorting-based Learning Complexity
   - FlexRand: Flexible Under-sampling with Randomness
   - Extension to Instruction Dataset Pruning
3. Experiments
   - Main Results
   - Ablation Studies
4. Discussions
   - Relationship with the pre-trained models
5. Summary

Background

Prohibitive Model Development Cost from Neural Scaling Laws

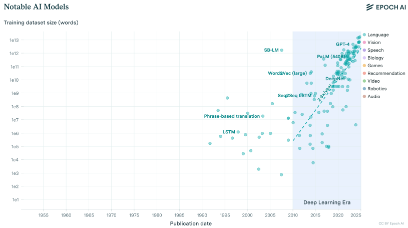
The paradigm of pre-training and fine-tuning (PT-FT) is increasingly popular with the rapid advancements in foundation models. Unfortunately,

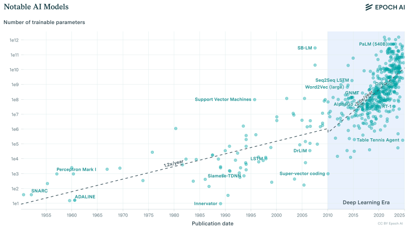
- **Training costs are ever-increasing** due to the neural scaling laws

$$(N^{\mathrm{opt}}, D^{\mathrm{opt}}) = \mathrm{argmin}_{N,D}\mathcal{L}(N, D) \tag{1}$$

$$\text{s.t.} \quad \mathrm{FLOPs}(N, D) = C \tag{2}$$

, where N is the model parameter, D is the training dataset and C is computing power. We aim to allocate C between N and D optimally.



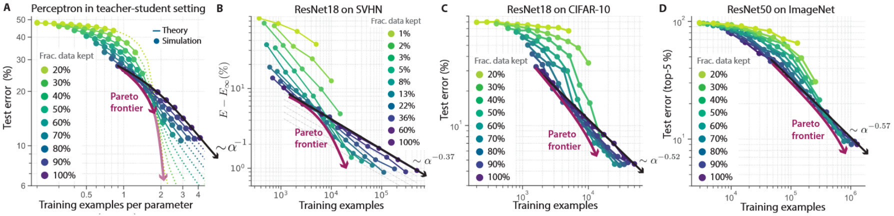


Background

Beating Neural Scaling Laws via Dataset Pruning

From the perspective of dataset size, recent work [1] demonstrates that we can break beyond power laws and potentially even reduce it to exponential scaling instead if we have access to

- **A high-quality dataset pruning metric** that ranks the order in which training examples should be discarded to achieve any size.



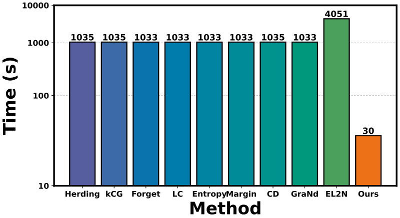
Background

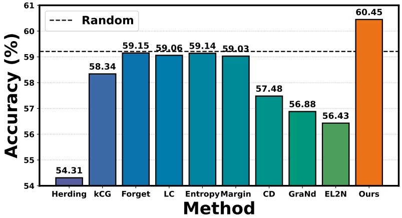
Concerns of Efficiency and Efficacy

Existing dataset pruning methods propose diverse metrics to quantify the sample importance through a scoring function $S(\boldsymbol{x})$, which are generally designed for training from scratch.

When transferring to the PT-FT, we face the following challenges:

- **Inefficiency:** Fine-tuning the entire dataset for estimating $S(\boldsymbol{x})$ can be prohibitively expensive due to the computationally intensive back-propagation on the huge parameters.
- **Ineffectiveness:** Existing methods empirically show inferior performance than the random on the diverse PT-FT benchmarks.





Method

DLC: Distorting-based Learning Complexity (I)

To design an efficient and effective $S(\boldsymbol{x})$, we first define the Learning Complexity from the hardness perspective based on:

Learning Path

A sequence of model parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}(t) \mid t \in \mathcal{T}\}$ can be defined as a learning path if there exists a positive constant $r < \mathcal{R}_{\mathcal{L}}(f_{\boldsymbol{\theta}(0)})$ such that $\lim_{t \to \infty} \mathcal{R}_{\mathcal{L}}(f_{\boldsymbol{\theta}(t)}) = r$.

, and the above Learning Complexity can be formally defined as follows:

Learning Complexity

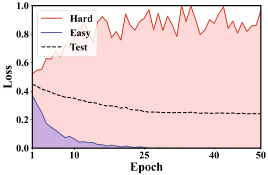
$$\mathrm{S}_{\mathrm{LC}}((\boldsymbol{x}, y)) = \int_{t \in \mathcal{T}} \mathcal{L}(f(\boldsymbol{x}; \boldsymbol{\theta}(t)), y) dt$$

Existing optimization-based learning complexity quantifies the sample hardness from the perspective that:

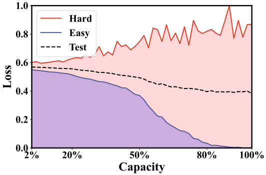
- Easy samples have larger $\frac{d\mathcal{L}}{dC}$ than the hard as shown in Figure (a).

Instead, we implement the learning complexity with a lightweight distorting process for efficiency, by the observation that:

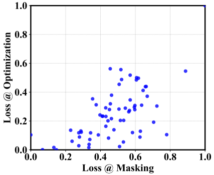
- Easy samples have larger $\frac{d\mathcal{L}}{dN}$ than the hard as shown in Figure (b).



(a)



(b)



(c)

In Figure (c), the ranking of DLC is **highly correlated** with the learning complexity based on optimization.

Method

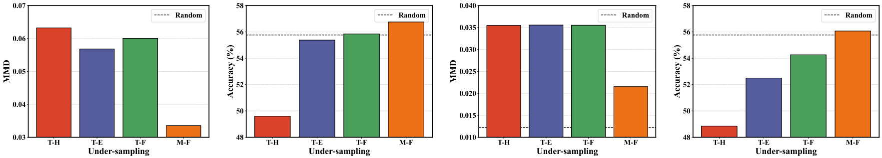
FlexRand: Flexible Under-sampling with Randomness

To identify informative samples with DLC, we further design a flexible under-sampling strategy with randomness, named **FlexRand** as follows:

$$p((\boldsymbol{x}, y)) = \begin{cases} \frac{M}{2N*\gamma}, & \text{if } \mathrm{S_{LC}}((\boldsymbol{x}, y)) < \mathrm{S}_\gamma \\ \frac{M}{2N*(1-\gamma)}, & \text{if } \mathrm{S_{LC}}((\boldsymbol{x}, y)) \geq \mathrm{S}_\gamma \end{cases}$$

which offers two advantages

- **FlexRand can adapt to different data regimes.**
- **FlexRand avoids severe distribution shift.**



When extending the learning complexity to prune instruction datasets for efficient LLMs fine-tuning, we replace the original loss function with:

$$\mathrm{S_{LC}}((\boldsymbol{x}, \boldsymbol{y})) = \frac{1}{C} \sum_{j=0}^{C-1} \mathcal{L}(f(y_{j-1} : \ldots : y_0 : \boldsymbol{x}; \boldsymbol{\theta}(t)), y_j)$$

where C is the length of the output $\boldsymbol{y}$, due to the prediction difference in image classification and text generation. Empirical results are presented in the following table:

Base Model	Alpaca Cleaned				Dolly & HH-RLHF			
	Humanity	Social Science	STEM	Other	Humanity	Social Science	STEM	Other
Mistral 7B	52.44 / **54.75**	71.89 / **72.64**	51.74 / **52.74**	68.88 / **70.20**	52.50 / **53.82**	69.58 / **71.47**	51.18 / **53.30**	68.01 / **68.91**
Llama3 8B	54.24 / **56.75**	71.99 / **72.05**	52.33 / **54.84**	69.78 / **70.20**	52.52 / **53.94**	69.13 / **72.60**	49.92 / **53.60**	68.39 / **69.65**
Gemma2 9B	56.37 / **58.79**	73.29 / **75.25**	54.14 / **56.30**	71.13 / **71.52**	55.21 / **56.08**	71.43 / **73.32**	50.23 / **52.99**	69.51 / **70.60**

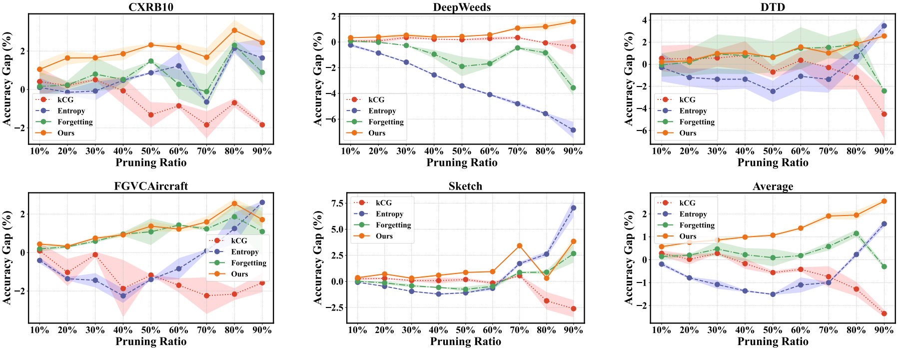
We can find that **our method consistently outperforms the random with various pre-trained models and instruction fine-tuning datasets**.

Experiments

Main Results

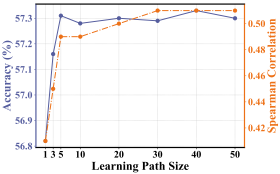
In the downstream image dataset pruning benchmark, the empirical results demonstrate that our method

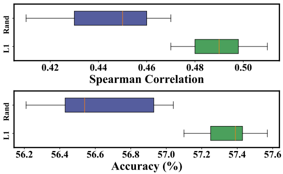
- **achieves superior accuracy with different architectures.**
- **consistently outperforms the random with diverse setups.**
- **significantly reduces the time cost of dataset pruning by 35x.**

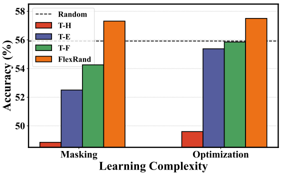


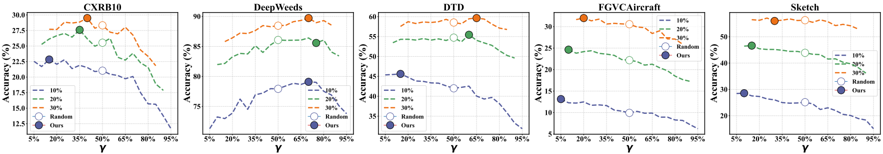
To better understand why our method can obtain superior efficiency and efficacy, we perform extensive ablation studies on the

- **Learning path size**
- **Weight masking principle**
- **Under-sampling strategy**
- **Interval splitting**









- **Is the proposed method affected by the pre-training dataset?**

Method	10%∼30%					40%∼60%					70%∼90%					Average
	CXR	DW	DTD	FA	Sk	CXR	DW	DTD	FA	Sk	CXR	DW	DTD	FA	Sk	
Random	27.27	87.42	54.67	32.15	56.89	32.80	92.91	66.32	55.36	72.81	34.68	94.45	69.44	66.97	77.35	61.43
Herding	24.77	49.83	32.31	25.32	44.74	31.90	78.43	56.41	50.32	67.55	36.28	91.89	67.55	65.71	76.40	53.29
kCG	25.48	87.30	49.13	28.50	55.83	31.27	93.25	65.67	54.74	73.63	35.30	94.66	70.29	67.42	77.60	60.67
Forgetting	28.60	86.53	54.99	**35.94**	**58.75**	34.17	91.89	66.86	59.86	73.12	36.18	94.33	70.29	68.39	76.92	62.45
Least Conf	27.13	83.48	55.60	30.63	58.07	32.93	89.71	64.49	51.20	71.09	36.20	93.26	69.18	64.24	76.68	60.26
Entropy	28.42	82.95	55.52	31.09	58.45	32.88	89.93	64.41	51.28	71.23	35.72	93.55	68.86	64.27	76.80	60.36
Margin	28.12	83.20	55.80	30.52	58.43	33.92	89.85	64.21	50.59	71.35	36.52	93.41	68.63	64.30	76.61	60.36
CD	23.00	87.07	52.97	34.49	54.85	32.03	94.13	66.14	59.53	**73.73**	36.07	**95.11**	70.52	68.67	**78.20**	61.77
GraNd	21.77	86.40	53.43	35.08	54.24	30.70	94.04	66.02	60.38	73.38	36.03	94.85	70.35	**68.91**	78.00	61.57
EL2N	19.62	87.13	52.91	34.45	53.70	31.70	**94.31**	66.24	**60.94**	73.13	36.17	95.08	70.35	**68.91**	77.75	61.49
Ours	**29.69**	**88.49**	**57.45**	34.67	58.41	**34.79**	93.40	**67.12**	56.97	73.65	**37.42**	94.90	**70.66**	67.29	77.79	**62.85**

- **Can we use small models to select samples for large models?**

$\mathrm{S}((\boldsymbol{x}, y))$	ResNet-50					ViT-Small					ViT-Base					Average
	CXR	DW	DTD	FA	Sk	CXR	DW	DTD	FA	Sk	CXR	DW	DTD	FA	Sk	
Random	31.73	91.99	67.44	46.75	63.63	33.13	92.60	66.56	48.02	66.27	31.88	91.86	64.81	44.94	64.10	60.38
Original	**32.90**	**92.50**	**68.29**	**47.61**	**64.48**	34.12	**92.99**	67.29	**48.41**	**66.93**	32.63	**92.17**	**65.26**	**45.63**	**65.06**	**61.08**
Transfer (RN18)	31.61	90.43	68.03	46.78	63.98	**34.71**	92.93	**67.45**	48.37	66.69	**32.94**	91.82	65.12	44.99	64.61	60.70

Summary

- **Contribution**
  - We propose Distorted-based Learning Complexity (**DLC**), a novel and straightforward hardness score without relying on fine-tuning.
  - We design the **FlexRand** under-sampling, which can adapt to different data regimes while avoiding severe distribution shifts.
  - Comprehensive experiments verify the effectiveness and efficiency of the proposed method on comprehensive benchmarks.
- Learn More!
  - Paper: `https://arxiv.org/pdf/2402.05356.pdf`

References I

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos.
Beyond neural scaling laws: beating power law scaling via data pruning.
In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Daquan Zhou, and Yang You.
Infobatch: Lossless training speed up by unbiased dynamic data pruning.
arXiv preprint arXiv:2303.04947, 2023.

References II

Max Welling.
Herding dynamical weights to learn.
In *International Conference on Machine Learning*, pages 1121–1128, 2009.

Ozan Sener and Silvio Savarese.
Active learning for convolutional neural networks: A core-set approach.
In *International Conference on Learning Representations*, 2018.

Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora.
Contextual diversity for active learning.
In *European Conference on Computer Vision*, pages 137–153, 2020.

References III

Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q. Weinberger.
Multi-scale dense networks for resource efficient image classification.
In *International Conference on Learning Representations*, 2017.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy J. Lin.
Deebert: Dynamic early exiting for accelerating bert inference.
In *Annual Meeting of the Association for Computational Linguistics*, 2020.