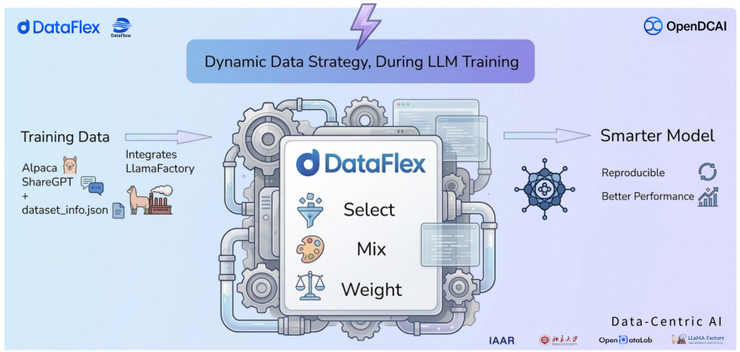
DataFlex: A Unified Framework for Data-Centric Dynamic Training of Large Language Models

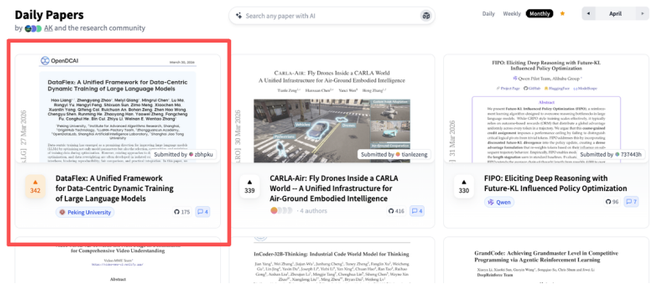
Hao Liang, Zhengyang Zhao, Meiyi Qiang, Mingrui Chen, Lu Ma, Rongyi Yu, Hengyi Feng, Shixuan Sun, Zimo Meng, Xiaochen Ma, Xuanlin Yang, Qifeng Cai, Ruichuan An, Bohan Zeng, Zhen Hao Wong, Chengyu Shen, Runming He, Zhaoyang Han, Yaowei Zheng, Fangcheng Fu, Conghui He, Bin Cui, Zhiyu Li, Weinan E, Wentao Zhang

INTRODUCTION

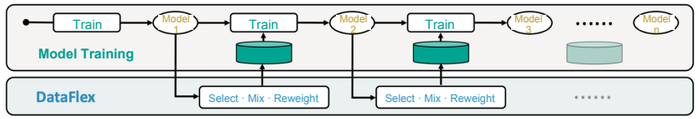
The teams of Prof. Wentao Zhang and Academician Weinan E at Peking University, together with the LLaMA-Factory Team, the Shanghai Institute for Algorithm Innovation, and other partners, introduced **DataFlex**, a data-centric framework for dynamic LLM training. Beyond individual algorithms or collections of scripts, it provides unified training infrastructure built on LLaMA-Factory. It integrates **dynamic sample selection, dynamic data mixing, and dynamic sample reweighting** directly into training, turning how data participates in training from an empirical configuration into a controllable, optimizable, and reproducible system capability.



Following its release, DataFlex attracted widespread attention on Hugging Face Daily Papers and ranked **first on the monthly leaderboard**. This response marks the community's recognition of a milestone: bringing data-centric dynamic training from theory into a complete engineering workflow.



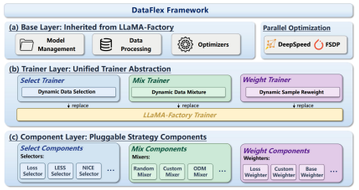
DESIGN PHILOSOPHY



Move beyond static data feeding and make data scheduling a ready-to-use system capability, guided by three principles:

- **Unity:** Bring three representative data-centric training paradigms into one framework.
- **Compatibility:** Integrate with existing large-scale training infrastructure without introducing a separate workflow.
- **Extensibility:** Implement and compare new data-centric algorithms with low engineering overhead.

SYSTEM ARCHITECTURE

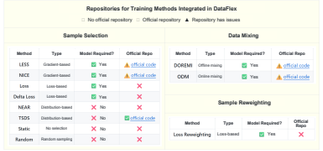


DataFlex retains LLaMA-Factory's clear, accessible design while making key architectural upgrades. Within the existing training ecosystem, it turns data-centric training into a unified, extensible, reproducible, and deployable system capability. The framework comprises three layers:

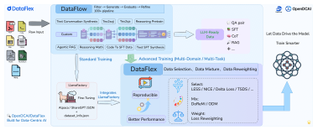
- **Base Layer:** Inherited from LLaMA-Factory, this layer handles model management, data processing, optimizers, and other general training capabilities. It preserves familiar workflows while focusing extensions on data-centric training, lowering the cost of migration to DataFlex.
- **Trainer Layer:** Instead of a single original trainer, training is abstracted into three data-centric modes: data selection, data mixing, and sample reweighting. Trainers therefore handle both data decisions and parameter optimization.
- **Component Layer:** Pluggable selectors, mixers, and weighters encapsulate each algorithm's strategy and expose a unified interface to the trainers.

ALGORITHMS & EXTENSIBILITY

DataFlex integrates representative methods such as LESS, DoReMi, ODM, and Loss Reweighting across its three trainer types. Each method is a pluggable component behind a unified interface, enabling fair comparisons under controlled conditions. Many influential dynamic training methods lack official repositories or have implementations that are difficult to reproduce. Through systematic refactoring, DataFlex restores production-level usability to these unavailable or partially stalled algorithms.



SYNERGY WITH DATAFLOW



DataFlex and the team's other core project, DataFlow (3k+ stars), establish a complete workflow for data use in the AI era. DataFlow refines vast, unstructured real-world information into data with high intelligence density. DataFlex takes this refined data deep into the model training process. Beyond accelerating convergence, it controls model generalization and knowledge acquisition at the system level, translating the potential of data into stronger practical AI capabilities.

PERFORMANCE VALIDATION

1. Data Selection & Reweighting

Experiments on an Open-Hermes-2.5 subset show that most dynamic data-centric methods outperform static full-data training on both Mistral-7B and Llama-3.2-3B. The results indicate that, when model capacity is limited, dynamic selection that responds to the model's current state is crucial to reaching its performance ceiling.

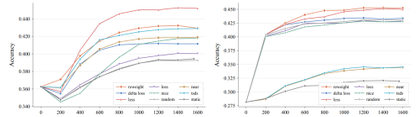


Figure 3 Accuracy comparison across different data selection and sampling strategies on two backbone models. **Left:** Mistral-7B. **Right:** Llama-3.2-3B.

2. Data Mixing

Under both the 6B and 30B SlimPajama settings, DoReMi and ODM show clear advantages. Benefits are already evident at 6B tokens: ODM achieves higher accuracy on general capability evaluations than the default static mixture, while DoReMi achieves better overall perplexity. These results show that dynamically adjusting the proportions of data domains can improve training outcomes.

Table 2 Comparison of data mixing strategies. MMLU accuracy (Acc, ↑) and perplexity (PPL, ↓) across different data domains.

Method	Acc ↑	Perplexity (PPL) ↓						
	MMLU	ALL	CC	C4	SE	Wiki	GitHub	ArXiv	Book
Slim-Pajama-6B									
Baseline	25.27	4.217	4.278	4.532	3.402	**3.546**	**2.640**	3.508	4.778
DoReMi	25.84	**4.134**	**4.108**	**4.358**	3.788	3.997	3.420	3.413	4.661
ODM	**26.04**	4.244	4.326	4.555	**3.243**	3.699	2.704	**2.904**	**4.613**
Slim-Pajama-30B									
Baseline	25.51	3.584	3.723	3.505	2.850	3.215	3.163	4.540	5.329
DoReMi	**25.97**	3.562	3.731	**3.503**	2.706	2.985	2.973	4.441	5.214
ODM	25.63	**3.429**	**3.598**	3.519	**2.382**	**2.713**	**2.255**	**3.487**	**4.746**

3. System Efficiency

DataFlex also improves efficiency. With LESS on a single GPU, it reduces training time across multiple sampling ratios. At a ratio of 1.0, training time falls from **30,239 s to 28,734 s**, while accuracy rises from **40.38% to 42.37%**. With eight H20 GPUs, training time is reduced by **57.13%**.

Table 3 Efficiency and accuracy comparison between DataFlex and LESS evaluated on the MMLU test set. Each model is trained independently on a subset of Open-Hermes-2.5, sampled from a fixed pool of 100,000 samples at ratios of 0.05, 0.1, 0.5, and 1.0. Time reduction is calculated relative to the LESS baseline at the same ratio. The 8-GPU configuration highlights the parallel scalability of our framework.

Sample Ratio	Method	Accuracy(%)	Training Time (s)	Reduction ↓
0.05	LESS	34.91	1,640	-
	DataFlex	**38.35**	**1,579**	**3.72%**
0.1	LESS	37.97	3,735	-
	DataFlex	**40.25**	**3,573**	**4.34%**
0.5	LESS	**41.57**	14,398	-
	DataFlex	40.93	**13,377**	**7.09%**
1.0	LESS	40.38	30,239	-
	DataFlex	**42.37**	**28,734**	**4.98%**
1.0	DataFlex (8-GPUs)	**43.01**	**12,965**	**57.13%***

*Compared to DataFlex (Single-GPU) at 1.0 ratio.

For offline selection methods such as TSDS, DataFlex's reimplementation also delivers consistent speedups of **1%–3.5%** across different data scales.

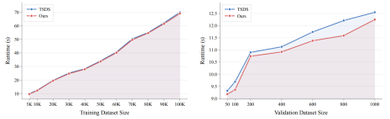


Figure 4 Efficiency comparison between the original TSDS operator and our implementation under different dataset scaling strategies. Left: the validation set size is fixed while the training set size varies. Right: the training set size is fixed while the validation set size varies.

ABOUT THE AUTHORS Peking University DCAI Team, focusing on LLM data systems and Data-Centric AI infrastructure.

WeChat Group | GitHub Repository | Bilibili Tutorials | Illustrated Guides