DATA SELECTION VIA OPTIMAL CONTROL FOR LANGUAGE MODELS

Yuxian Gu[1,2]*, **Li Dong**[2], **Hongning Wang**[1] **Yaru Hao**[2], **Qingxiu Dong**[3]*, **Furu Wei**[2], **Minlie Huang**[1]†
[1]The CoAI Group, Tsinghua University [2]Microsoft Research [3]Peking University

Why do we need data selection?

- **Efficiency**: Reduce the computational cost during pre-training
- **Effective**: Training on the highest quality data can lead to stronger performance[1]

[1] GLaM: Efficient Scaling of Language Models with Mixture-of-Experts, *Nan Du et al. Proceedings of the 39th International Conference on Machine Learning*, PMLR 162:5547-5569, 2022.

Why do we need data selection?

- **Efficiency**: Reduce the computational cost during pre-training
- **Effective**: Training on the highest quality data can lead to stronger performance[1]

How to define and measure data quality?

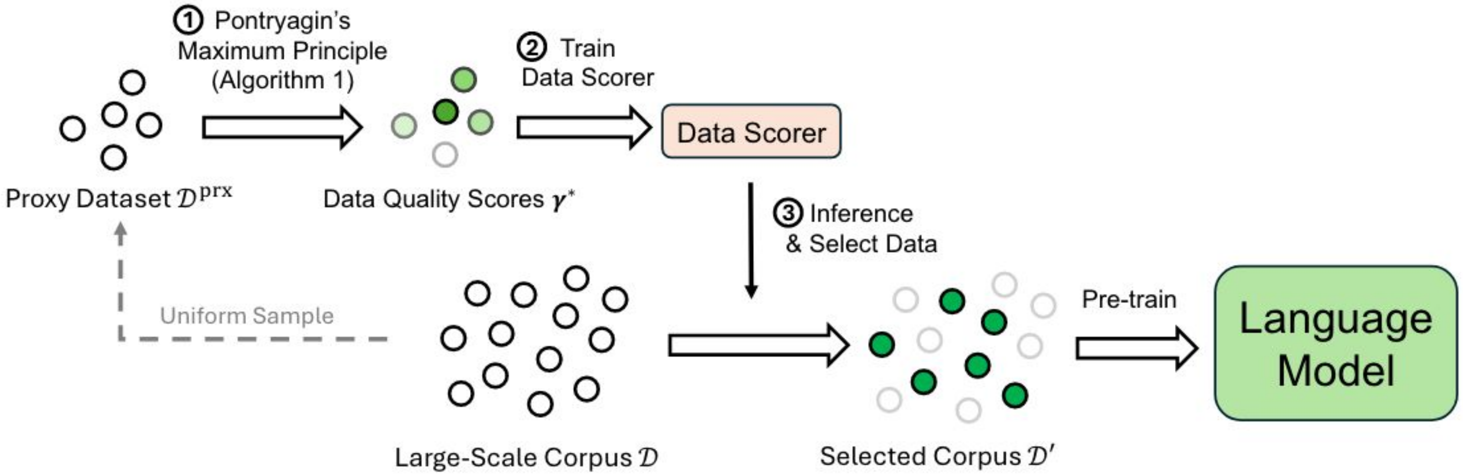
How to select data based on such quality measurements?

[1] GLaM: Efficient Scaling of Language Models with Mixture-of-Experts, *Nan Du et al. Proceedings of the 39th International Conference on Machine Learning*, PMLR 162:5547-5569, 2022.

How this work approaches these questions…

Map data selection as discrete **optimal control problem**

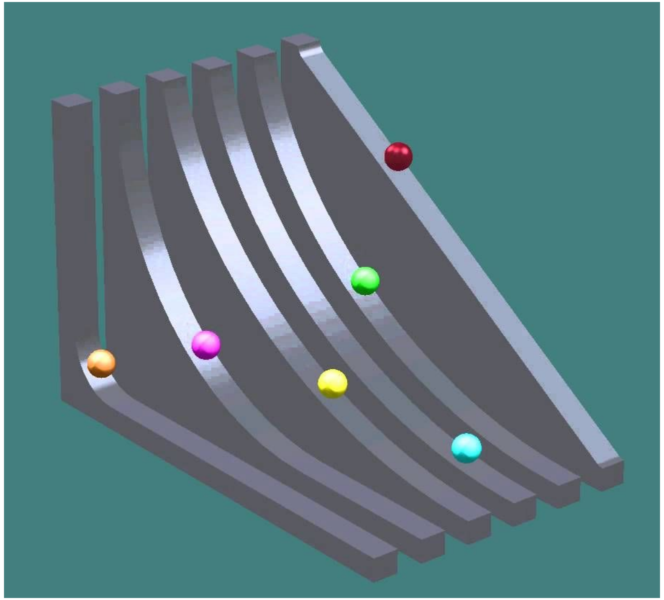
Data selection	LM pre-training	LM's downstream performance
Control variables	Dynamic systems	Objective



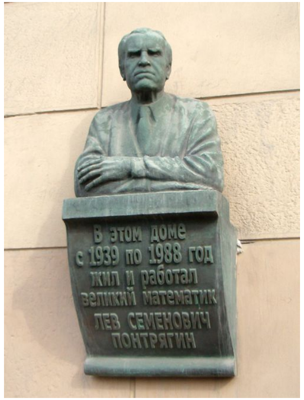
What is optimal control theory?

“a branch of control theory that deals with finding a **control** for a **dynamical system** over a period of time such that an **objective function** is optimized.”

-- Wikipedia



Brachistochrone problem(1696)



Lev Pontryagin
Pontryagin’s Maximum Principle

Richard Bellman
Dynamic programming

Formulation of optimal control problem

Objective function e.g. deviation from trajectory of a rocket

$$\boxed{\min_{\gamma_t} \sum_{t=0}^{T-1} \mathcal{J}(\boldsymbol{\theta}_t, \gamma_t) + J(\boldsymbol{\theta}_T),}$$

$$s.t. \ \boxed{\boldsymbol{\theta}_{t+1} = f(\boldsymbol{\theta}_t, \gamma_t)}, \quad \gamma_t \in U,$$

Dynamics of a system, how state evolves over time e.g. rocket acceleration depends on thrust and gravity

State variable, e.g. Position, velocity, mass of the rocket

Control variables, e.g. Thrust direction and magnitude

Optimal control VS Reinforcement learning(bonus slide)

Optimal control emphasizes mathematical models and assumes known dynamics (model -> action)

Reinforcement learning adapts online, making it flexible under unknown or changing environments (data -> action)

Problem formulation for data selection

Data quality score
$$\boldsymbol{\gamma} = \left[\gamma_1, \gamma_2, \cdots, \gamma_{|\mathcal{D}|}\right]^\top$$

Intuition: higher quality score indicates the corresponding sample is more helpful to reduce J

$$U = \left\{ \left[\gamma_1, \gamma_2, \cdots, \gamma_{|\mathcal{D}|}\right]^\top \middle| \textstyle\sum_{n=1}^{|\mathcal{D}|} \gamma_n = 1 \text{ and } \gamma_n \geq 0 \text{ for } 1 \leq n \leq |\mathcal{D}| \right\}$$

Pretraining loss
$$L(\boldsymbol{\theta}, \boldsymbol{\gamma}) = \sum_{n=1}^{|\mathcal{D}|} \gamma_n l(x_n, \boldsymbol{\theta})$$

$$l(x_n, \boldsymbol{\theta}) = -\log p_{\boldsymbol{\theta}}(x_n)$$

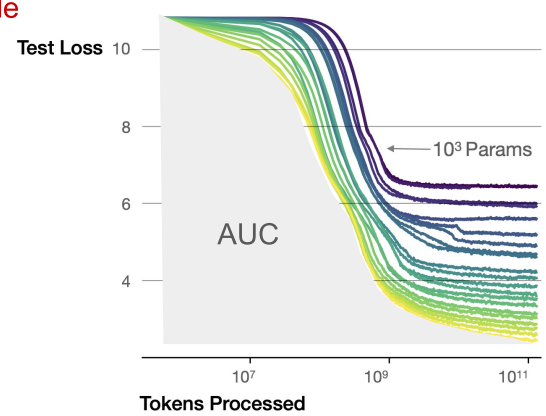
Single data sample

Parameter update in GD
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla L(\boldsymbol{\theta}_t, \boldsymbol{\gamma})$$

Optimization goal
$$\min_{\boldsymbol{\gamma}} \sum_{t=1}^{T} J(\boldsymbol{\theta}_t),$$

Minimize the area under curve approx. by cumulative sum of downstream task loss J

$$\text{s.t. } \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla L(\boldsymbol{\theta}_t, \boldsymbol{\gamma}), \quad \boldsymbol{\gamma} \in U.$$



How to map data selection to optimal control theory

Data selection	LM pre-training	LM's downstream performance
↓	↓	↓
Control variables	Dynamic systems	Objective

Formulation for **optimal control**

$$\min_{\gamma_t} \sum_{t=0}^{T-1} \mathcal{J}(\boldsymbol{\theta}_t, \gamma_t) + J(\boldsymbol{\theta}_T),$$

$$s.t.\ \boldsymbol{\theta}_{t+1} = f(\boldsymbol{\theta}_t, \gamma_t),\quad \gamma_t \in U,$$

Formulation for **data selection**

$$\min_{\gamma} \sum_{t=1}^{T} J(\boldsymbol{\theta}_t),$$

$$\text{s.t. } \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla L(\boldsymbol{\theta}_t, \gamma),\quad \gamma \in U.$$

Pontryagin's maximum principle

- Just the **first order necessary conditions** for an optimum deterministic (discrete) optimal control problem
- Plug in the Hamilton function into the lagrangian and take derivatives

- Although it's a necessary condition, the authors claim that previous studies have show PMP can successfully lead to fairly good solutions

Problem formulation:

$$\min_{\gamma_t} \sum_{t=0}^{T-1} \mathcal{J}(\boldsymbol{\theta}_t, \gamma_t) + J(\boldsymbol{\theta}_T),$$

$$s.t.\ \boldsymbol{\theta}_{t+1} = f(\boldsymbol{\theta}_t, \gamma_t), \quad \gamma_t \in U,$$

Optimal solution:

$$\begin{aligned}
\boldsymbol{\theta}^*_{t+1} &= \nabla_{\boldsymbol{\lambda}} H(\boldsymbol{\theta}^*_t, \boldsymbol{\lambda}^*_{t+1}, \gamma^*_t), \quad \boldsymbol{\theta}^*_0 = \boldsymbol{\theta}_0, \\
\boldsymbol{\lambda}^*_t &= \nabla_{\boldsymbol{\theta}} H(\boldsymbol{\theta}^*_t, \boldsymbol{\lambda}^*_{t+1}, \gamma^*_t), \quad \boldsymbol{\lambda}^*_T = \nabla J(\boldsymbol{\theta}_T), \\
\gamma^*_t &= \arg\min_{\gamma_t} H(\boldsymbol{\theta}^*_t, \boldsymbol{\lambda}^*_{t+1}, \gamma_t), \quad \gamma_t \in U,
\end{aligned}$$

$$H(\boldsymbol{\theta}, \boldsymbol{\lambda}, \gamma) = \mathcal{J}(\boldsymbol{\theta}, \gamma) + \boldsymbol{\lambda}^\top f(\boldsymbol{\theta}, \gamma).$$

Derived by plugging Hamiltonian into Lagrangian with some arithmetic trick

Data selection as optimal control

Theorem 2.1 (PMP Conditions for Data Selection). *Let γ^* solve the problem in Eq. (3), and $\boldsymbol{\theta}_t^*$ denote the LM parameters trained with γ^*. For $0 \leq t < T$, there exists a vector $\boldsymbol{\lambda}_t^* \in \mathbb{R}^N$ such that*

$$\boldsymbol{\theta}_{t+1}^* = \boldsymbol{\theta}_t^* - \eta \nabla L(\boldsymbol{\theta}_t^*, \boldsymbol{\gamma}^*), \quad \boldsymbol{\theta}_0^* = \boldsymbol{\theta}_0, \tag{4}$$

$$\boldsymbol{\lambda}_t^* = \boldsymbol{\lambda}_{t+1}^* + \nabla J(\boldsymbol{\theta}_t^*) - \eta \nabla^2 L(\boldsymbol{\theta}_t^*, \boldsymbol{\gamma}^*) \boldsymbol{\lambda}_{t+1}^*, \quad \boldsymbol{\lambda}_T^* = \nabla J(\boldsymbol{\theta}_T^*), \tag{5}$$

$$\boxed{\boldsymbol{\gamma}^*} = \arg\max_{\boldsymbol{\gamma}} \sum_{n=1}^{|\mathcal{D}|} \gamma_n \left[\sum_{t=0}^{T-1} {\boldsymbol{\lambda}_{t+1}^*}^\top \nabla l(x_n, \boldsymbol{\theta}_t^*) \right], \quad \boldsymbol{\gamma} \in U, \tag{6}$$

We get the optimal data score here

where $\nabla^2 L(\boldsymbol{\theta}_t^*, \boldsymbol{\gamma}^*)$ denotes the Hessian matrix of $L(\boldsymbol{\theta}, \boldsymbol{\gamma}^*)$ with respect to $\boldsymbol{\theta}$ evaluated at $\boldsymbol{\theta} = \boldsymbol{\theta}_t^*$.

Note: Due to the offline nature of data selection, data selection problem need to add additional invariant constraints to control variables, i.e. $\gamma_0 = \gamma_1 = \cdots = \gamma_{T-1}$

Understand PMP condition for data selection

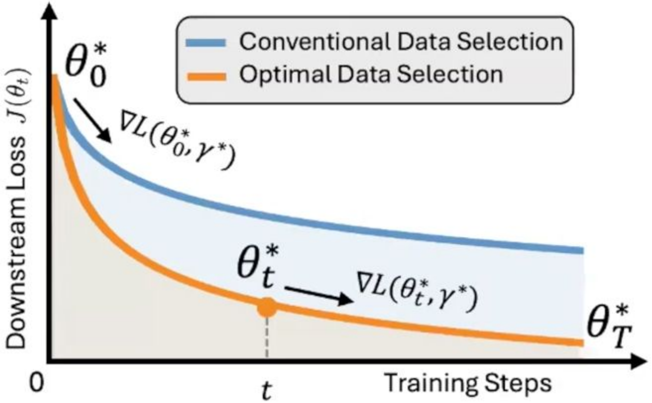
$$(1)\quad \boxed{\theta^*_{t+1}} = \theta^*_t - \eta\nabla L(\theta^*_t, \boxed{\gamma^*}), \quad \theta^*_0 = \theta_0,$$

Model Parameters

(Optimal) Data Quality Scores

- The first equation is exactly the parameter updating policy of training LMs

- Constrains the *parameters θ^*_t* are still reachable with GD *under the optimal data selection*
  - (or Adam, see Appendix C in out paper for derivations)



Slide source: https://www.youtube.com/watch?v=sDeOJdI_M-M

Understand PMP condition for data selection

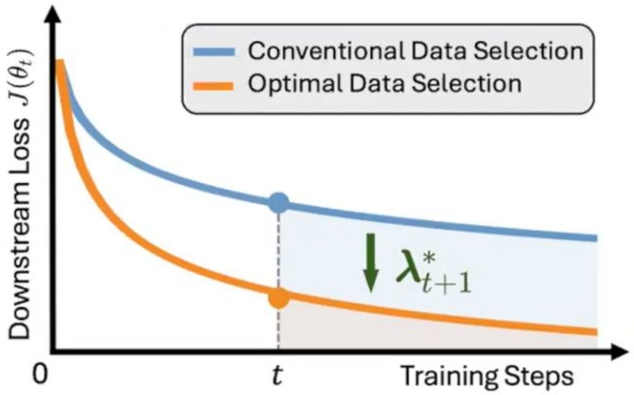
Information of the downstream loss

$$(2)\quad \boxed{\boldsymbol{\lambda}_t^*} = \boldsymbol{\lambda}_{t+1}^* + \boxed{\nabla J(\boldsymbol{\theta}_t^*)} - \eta \boxed{\nabla^2 L(\boldsymbol{\theta}_t^*, \boldsymbol{\gamma}^*)\boldsymbol{\lambda}_{t+1}^*}, \quad \boldsymbol{\lambda}_T^* = \nabla J(\boldsymbol{\theta}_T^*),$$

Target gradient direction

Information of learning dynamics of LMs

- The second equation defines λ_t, **the "target" gradient direction** of the high-quality data points.
  - Same size of the model parameters.
  - *"Compass" for high-quality data.*
- Target gradient direction includes information of downstream loss and LM's learning dynamics.



Derived from PMP

Slide source: https://www.youtube.com/watch?v=sDeOJdI_M-M

Understand PMP condition for data selection

Gradient of a single data point

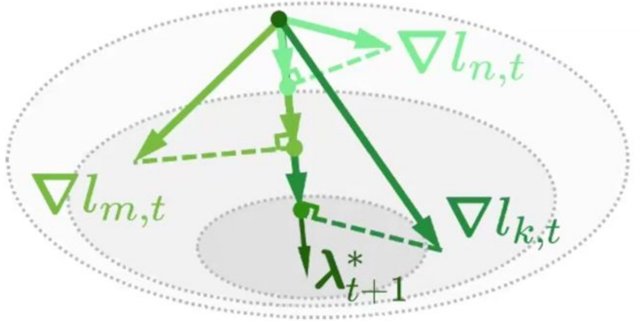
$$(3)\quad \boxed{\gamma^*} = \arg\max_{\gamma} \sum_{n=1}^{|\mathcal{D}|} \gamma_n \left[\sum_{t=0}^{T-1} \boxed{\boldsymbol{\lambda}^*_{t+1}}^{\top} \boxed{\nabla l(x_n, \boldsymbol{\theta}^*_t)} \right], \quad \gamma \in U,$$

Data Quality Scores

Target gradient direction

$$\gamma_1 + \gamma_2 + \cdots \gamma_{|D|} = 1$$
$$\gamma_1, \gamma_2, \dots \gamma_{|D|} \geq 0$$

- The third equation indicates that examples with **closer gradients to λ_t** should **have higher scores.**



$$\sum_t {\lambda^*_{t+1}}^{\top} \nabla l_{n,t} < \sum_t {\lambda^*_{t+1}}^{\top} \nabla l_{m,t} < \sum_t {\lambda^*_{t+1}}^{\top} \nabla l_{k,t}$$

Data Quality Scores: $\gamma_n < \gamma_m < \gamma_k$

Solve PMP conditions

- **PMP Conditions:**

Model Parameters

$$
\begin{cases}
(1)\ \boxed{\boldsymbol{\theta}_{t+1}^*} = \boldsymbol{\theta}_t^* - \eta \nabla L(\boldsymbol{\theta}_t^*, \boldsymbol{\gamma}^*), \quad \boldsymbol{\theta}_0^* = \boldsymbol{\theta}_0, \\
(2)\ \boxed{\boldsymbol{\lambda}_t^*} = \boldsymbol{\lambda}_{t+1}^* + \nabla J(\boldsymbol{\theta}_t^*) - \eta \nabla^2 L(\boldsymbol{\theta}_t^*, \boldsymbol{\gamma}^*) \boldsymbol{\lambda}_{t+1}^*, \quad \boldsymbol{\lambda}_T^* = \nabla J(\boldsymbol{\theta}_T^*), \\
(3)\ \boxed{\boldsymbol{\gamma}^*} = \arg\max_{\boldsymbol{\gamma}} \sum_{n=1}^{|\mathcal{D}|} \gamma_n \left[\sum_{t=0}^{T-1} \boxed{\boldsymbol{\lambda}_{t+1}^*}^{\top} \nabla l(x_n, \boldsymbol{\theta}_t^*) \right], \quad \boldsymbol{\gamma} \in U,
\end{cases}
$$

Data Quality Scores

Target gradient direction

- **How to Solve:**

Step 1: *Forward* Loop with (1): $\theta_0 \rightarrow \theta_1 \rightarrow \cdots\cdots \rightarrow \theta_T$

$\boldsymbol{\lambda}_T^* = \nabla J(\boldsymbol{\theta}_T^*)$

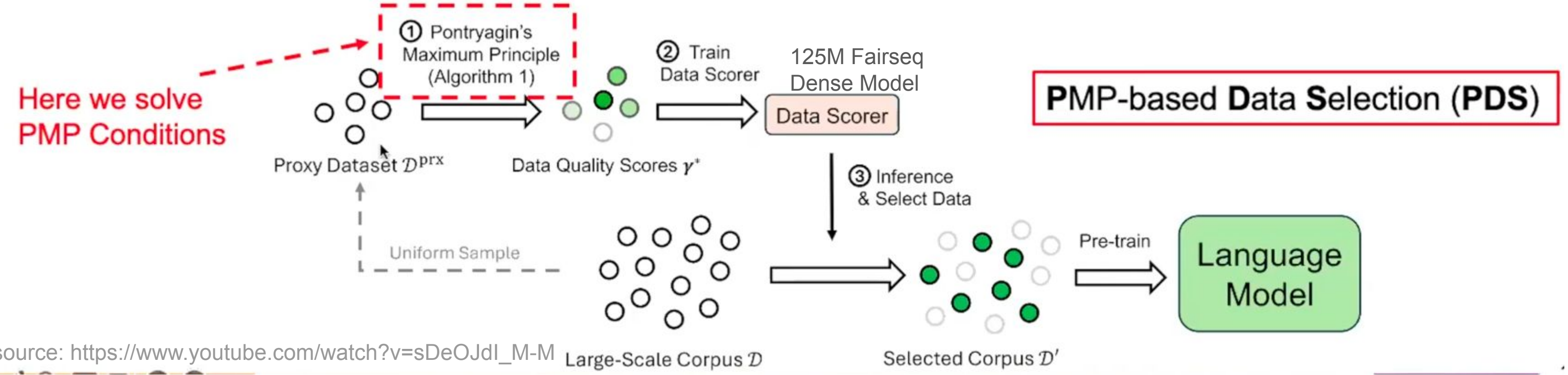
Step 2: *Reverse* Loop with (2): $\lambda_0 \leftarrow \lambda_1 \leftarrow \cdots\cdots \leftarrow \lambda_T$

Step 3: Compute γ with (3): γ

Slide source: https://www.youtube.com/watch?v=sDeOJdI_M-M

Efficient Implementation

- Forward and Reverse loops are computationally intensive
  - Training the full model. Hessian computation.
- Efficient “proxy to large” implementation
  - On downstream task (LIMA)
  - Solve the data scores on a small model (e.g., 140M) and small data (e.g., 160M tokens)
  - Fit the scores with a data scorer (e.g., a 140M LM with a regression head)
  - Infer data scores on the whole dataset (e.g., 100B tokens). Train large model (e.g., 1.7B).



Slide source: https://www.youtube.com/watch?v=sDeOJdI_M-M

Experiments

Can PDS fulfill our expectations on data selection techniques?

- Reduce the computational cost during pre-training
- Training on the highest quality data can lead to stronger performance

Investigate PDS performance under data constraints

Simulated setting when exact data quality score is attainable(skipped)

Ablation studies (skipped)

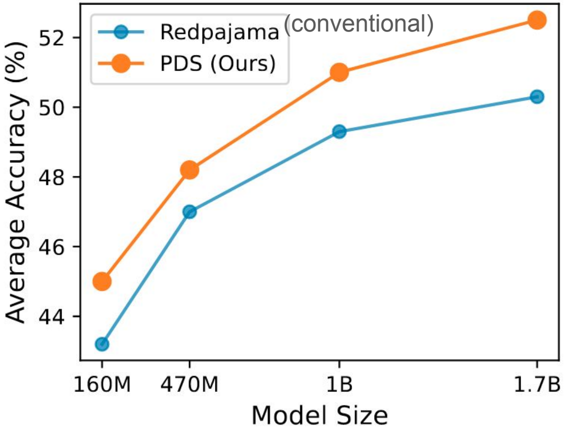
Experiment setup

- Training & evaluation
  - Pre-training LM from scratch
  - Evaluate zero-shot problem
- Data setups
  - Pre-training data: Commoncrawl from Redpajama (100B tokens)
  - Downstream loss(used for PMP): loss on LIMA (1k instruction pairs)
  - Evaluations: widely used NLP benchmarks
- Model
  - Mistral architecture: 160M, 470M, 1B, 1.7B
  - Extend to larger sizes (400B) with scaling laws

Can we have better performance with PDS selected data?

- Select 50B tokens from 125B token corpus
- Match the total training steps with the baselines

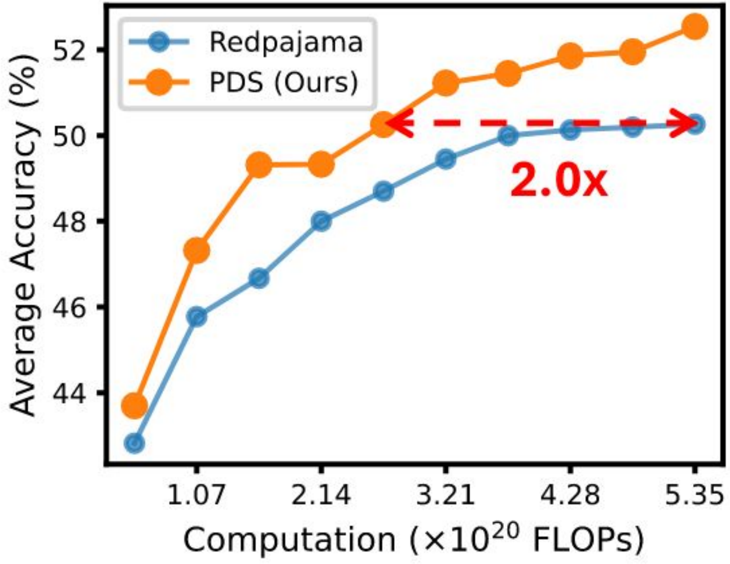
	HS	LAMB	Wino.	OBQA	ARC-e	ARC-c	PIQA	SciQ	BoolQ	Avg.
Model Size = 470M										
Conventional	36.7	41.4	52.4	**30.4**	44.8	25.2	61.0	70.6	60.4	47.0
RHO-Loss	36.6	42.4	53.0	29.4	43.7	25.2	60.4	72.8	59.8	47.0
DSIR	36.4	42.6	51.7	29.8	46.0	24.7	61.0	72.0	55.8	46.7
IF-Score	36.6	41.8	**53.4**	29.6	44.7	25.1	60.8	68.8	58.7	46.6
PDS	**37.9**	**44.6**	52.3	29.8	**46.5**	**25.8**	**61.8**	**73.8**	**61.4**	**48.2**
Model Size = 1B										
Conventional	39.9	47.6	52.4	30.6	49.3	26.4	63.1	73.7	60.9	49.3
RHO-Loss	39.8	47.0	53.0	30.8	48.0	26.4	62.9	71.1	**61.0**	48.9
DSIR	40.8	47.8	53.0	31.2	49.8	26.8	62.7	76.6	58.0	49.6
IF-Score	39.4	47.0	52.6	28.6	49.4	26.4	63.5	74.0	60.5	49.0
PDS	**42.1**	**48.8**	**54.0**	**33.4**	**51.3**	**28.0**	**64.1**	**78.5**	58.7	**51.0**



Slide source: https://www.youtube.com/watch?v=sDeOJdI_M-M

Can we reduce computational cost?

2.0x acceleration on 1.7B models



PDS is efficient and offline

		Complexity
PDS	Proxy γ-solver	$O(N^{\text{prx}}D + 4MN^{\text{prx}}D^{\text{prx}})$
	Data Scorer	$O(3N^{\text{score}}D^{\text{prx}} + N^{\text{score}}D)$
	Data Selection	$O(D)$
Pre-Training		$O(ND)$

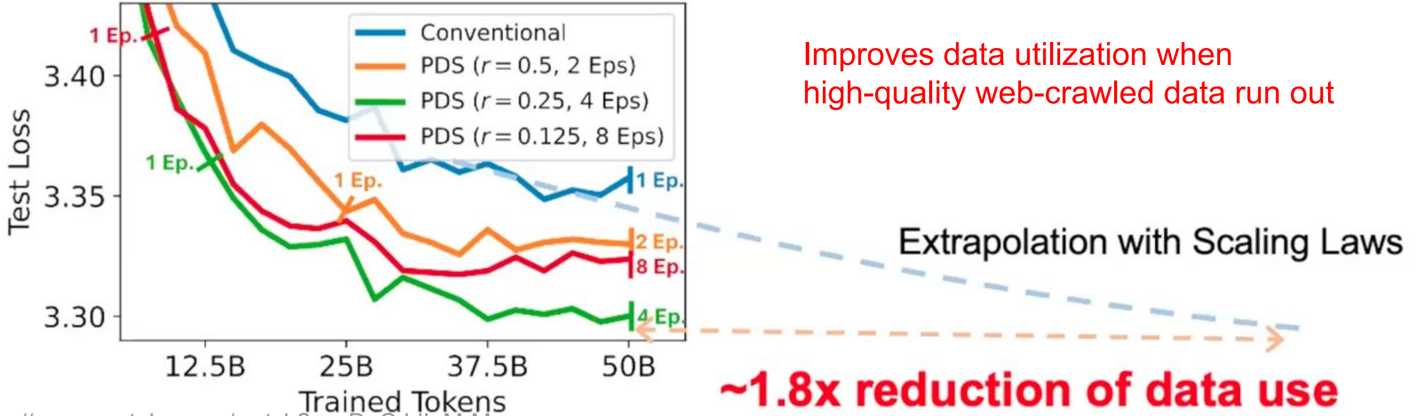
		FLOPs ($\times 10^{20}$)	Actual Time
PDS	Proxy γ-solver	0.49	15.2 Hours
	Data Scorer	0.063	1.50 Hours
	Data Selection	0.0	10.2 Minutes
Pre-Training		5.1	144 Hours

Slide source: https://www.youtube.com/watch?v=sDeOJdI_M-M

Data Utilization Improvement

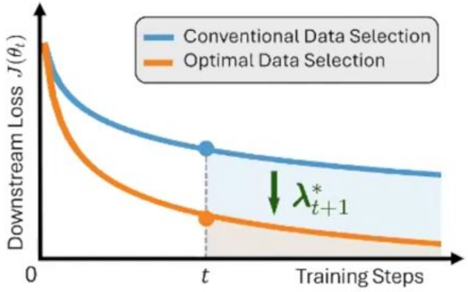
- Performance improvement with limit data (50B tokens)

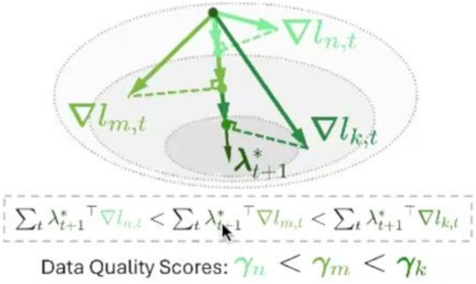
Improves data utilization when
high-quality web-crawled data run out

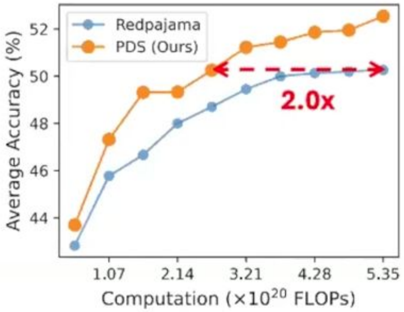


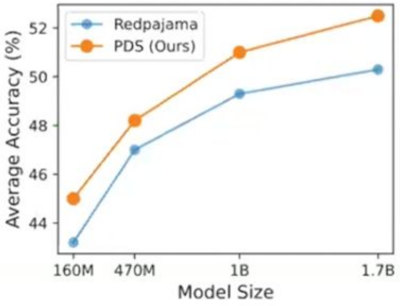
Conclusion

- A novel perspective for Data selection: Optimal Control problem









- Good theoretical guarantees ✅
- Efficient Implementation ✅
- Sound empirical results ✅

A **rigorous, theory-driven alternative** to the ad-hoc practices that currently dominate LM pre-training

Slide source: https://www.youtube.com/watch?v=sDeOJdI_M-M