PiKE : Adaptive Data Mixing for Multi-Task Learning Under Low Gradient Conflicts

Zeman Li[1,2], Yuan Deng[2], Peilin Zhong[2], Meisam Razaviyayn[1,2], Vahab Mirrokni[2]

Full Paper

[1]University of Southern California, [2]Google Research

NeurIPS 2025 Spotlight

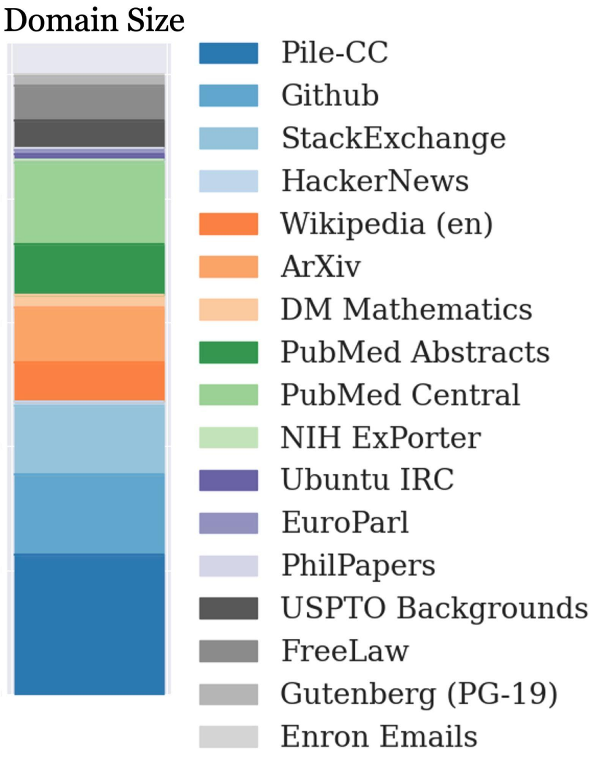
Data Mixing in Pre-training Large Language Models (LLMs)

LLM data come from K data source (Wikipedia, web, books, etc.)

Question: For a fixed budget, how much of each should we train on?

Key Challenges of Finding Optimal Data Mixture

1. Training LLMs is **costly** and **done in one run**
2. Sources of data are **diverse** (e.g. 18+ sources)
3. No target data distribution is known



Data composition of the Pile dataset (Gao et al., 2020)

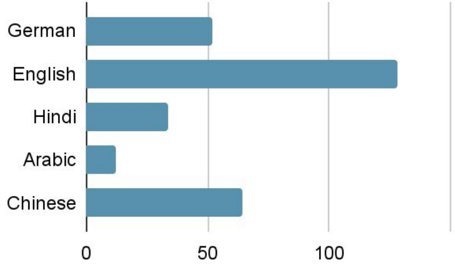
Data Mixing in Pre-training LLMs: A Multi-task Learning (MTL) Perspective

K Data Domains $\mathcal{D}_k$

Hello, World!
Hallo, Welt!
हैलो वर्ल्ड!
مرحبا بالعالم!
你好，世界！
……

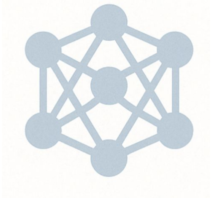
Fixed/Adaptive Weights

Per Task Samples b_k



Form a batch and update the parameters

Pre-training LLM $\boldsymbol{\theta}_t$



We can formulate this as a MTL problem:

Per-task loss function

Main Objective: $$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \mathcal{L}(\boldsymbol{\theta}) := \sum_{k=1}^{K} \mathbb{E}_{x\sim\mathcal{D}_k}[\ell_k(\boldsymbol{\theta};x)],$$

MTL Gradient: $$\mathbf{g}_t = \sum_{k=1}^{K} \mathbf{g}_{t,k},$$

Per-task gradient

Question: What is a good data mixing strategy from MTL perspective?

Do Any Methods From Previous MTL Literature Apply?

Unfortunately, the short answer is **NO.**

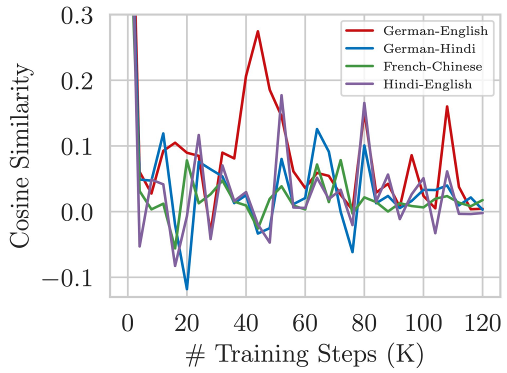
Challenges/Problems with previous MTL methods:

1) **Previous MTL methods mostly cannot scale** (e.g. PCGrad (2020), CAGrad (2021), NashMTL (2022) …)

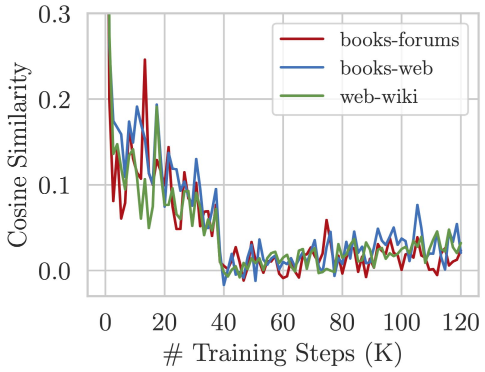
Standard pre-training 1B model only needs 8 Viperfish TPUs, while previous MTL methods require **K times more** TPUs

2) **Many MTL methods are designed to remove gradient conflict**

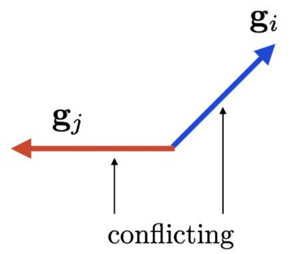
However, we observe that pre-training LLMs typically results in **very low gradient conflicts**



Pre-train 1B GPT-2 style model with multilingual-C4 (de), mC4 (fr), mC4 (zh) and mC4 (hi).



Pre-train 750M GPT-2 style model with GLaM Datasets (6 different sources e.g. Web, WiKi, Books,...)



An illustration of gradient conflict

What Makes a Good Mixing Algorithm When There's No Gradient Conflict?

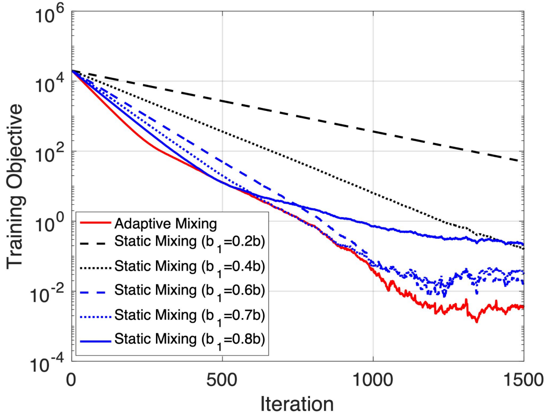
Our initial insights start with a simple model (Two tasks linear regression)

Case Study: Consider two loss functions $\ell_1(\boldsymbol{\theta}; x_1) = \frac{1}{2}(\boldsymbol{\theta}^\top e_1)^2 + x_1^\top \boldsymbol{\theta}$ and $\ell_2(\boldsymbol{\theta}; x_2) = \frac{1}{2}(\boldsymbol{\theta}^\top e_2)^2 + x_2^\top \boldsymbol{\theta}$.

Analyzing this simple case gives us two insights

1. Optimal batch composition should **change** over iterates

2. Data sampling weights depend on the **magnitude** and **variance** of per-task gradients



Adaptive mixing consistently outperform static mixing in the case study.

PiKE : Positive gradient Interaction-based K-task weights Estimator

We analysis the large-scale nonconvex models (billions+ params) based on no/low gradient conflicts

Every T_0 (e.g. 1,000 steps), **PiKE adaptively** adjusts the sampling weights based on

$$w_k \leftarrow w_k \exp\left(\zeta_1 \overset{\text{Task k gradient norm}}{\|\nabla \mathcal{L}_k(\boldsymbol{\theta})\|^2} - \zeta_2 \overset{\text{Task k gradient variance}}{\sigma_k^2}\right)$$

mirror descent

hyperparams

The data sampling rate of one domain should increase ($\uparrow$) if

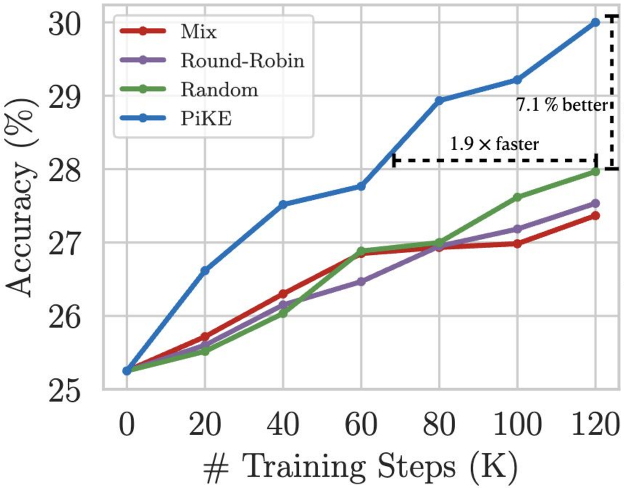
Task k gradient norm $\|\nabla \mathcal{L}_k(\boldsymbol{\theta})\|^2$ $\Uparrow$ **or** Task k gradient variance σ_k^2 $\Downarrow$

Theoretically, PiKE maximizes a **tight** lower bound on the objective decrease amount

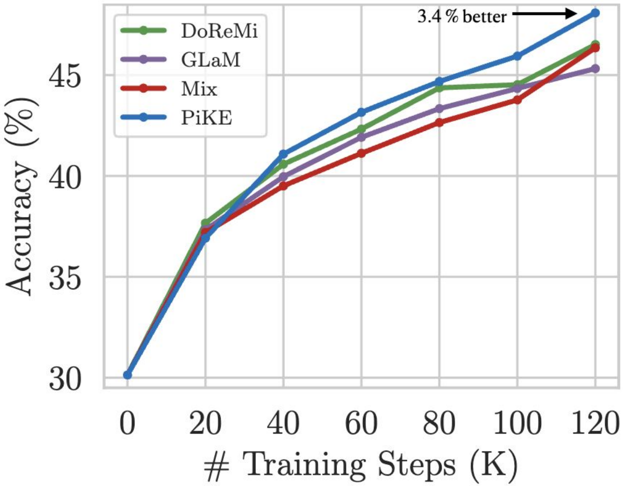
Pre-training Results Using PiKE

Key Features of PiKE:

1. **Efficient Scaling:** almost no memory and training (~1.2%) overhead.
2. **Balanced Learning: Promotes fairness across different tasks.**
3. **Improved Performance: 3.4%** better than previous complex data mixing.



Pre-train 1B language model with C4 (English) and C4 (Hindi) datasets



Pre-train 750M language model with GLaM Datasets (6 different sources e.g. Web, WiKi, Books,...)

Du, Nan, et al. "Glam: Efficient scaling of language models with mixture-of-experts."ICML (2022).
Xie, Sang Michael, et al. "Doremi: Optimizing data mixtures speeds up language model pretraining." Neurips (2023):

Thank You!

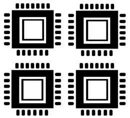
Previous Data Mixture Approaches in LLMs

Heuristic Rules / Manual Tuning

1. Manual Tuning (Llama 3, PaLM, …)
2. Data Abundance (mT5) [Xue et al., 2020]

Training Small Proxy Models

1. Downstream Performance (GLaM) [Du et al., 2021]
2. Group DRO (DoReMi) [Xie et al., 2021]



Problems with Previous Approaches

1. Lack of theoretical backup
2. Small proxy model may **not transfer** to larger ones [Ye et al., 2024]
3. The computational overhead is not **negligible**
4. **Fixed data mixture** weights may be **suboptimal** (as we will show)

Conceptual PiKE

Our analysis shows two insights

1. Optimal batch composition should **change** over iterates

2. Data sampling weights depend on the **magnitude** and **variance** of per-task gradients

Our analysis also shows that we should include **more data** (⬆) from one's task in one batch if

Magnitude (⇧) and **variance** (⇩) of one task's gradients

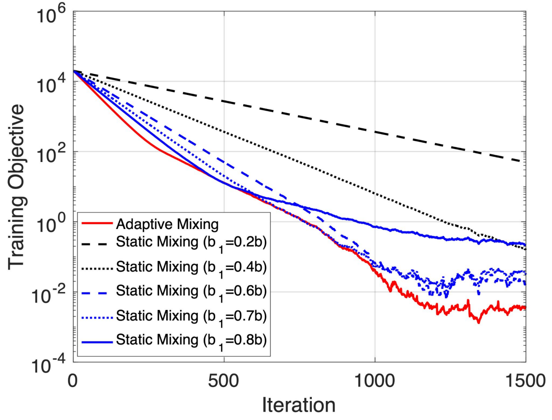
Intuitively speaking,

Task k gradient norm

$\|\nabla \mathcal{L}_k(\boldsymbol{\theta})\|^2$ measures how much **progress** one can make by following this gradient

Task k gradient variance

σ_k^2 measures how much **confidence** we have for the gradient estimate



Adaptive mixing consistently outperform static mixing in our case study.

When two tasks have similar gradient magnitudes, we should prioritize the task with smaller gradient variance.

⇨ This approach ensures that we prioritize tasks where confidently good progress can be made.

Appendix / Extra Slides

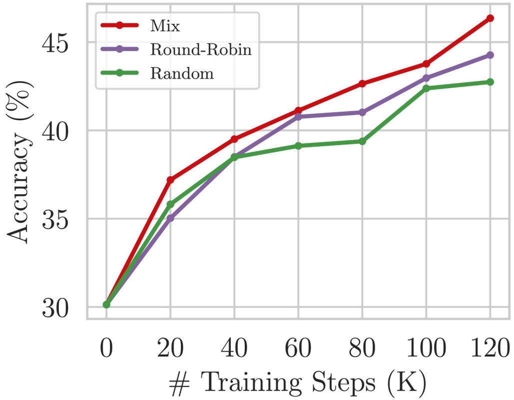
Mixing Domains in Each Batch Improves LLM Generalization

Assume static, uniform sampling weights (i.e., 1/K per task), we compare three standard strategies:

Number of samples from each tasks

$$\mathbf{b}_t = \begin{cases} b \cdot \mathbf{e}_{k^*} & \textit{Random}, \text{ where } k^* \sim \text{Uniform}(\{1, \ldots, K\}) \\ b \cdot \mathbf{e}_{(t \bmod K)+1} & \textit{Round-Robin} \\ b \cdot \left(\frac{1}{K}, \ldots, \frac{1}{K}\right) & \textit{Mix} \end{cases}$$

where $\mathbf{e}_k \in \mathbb{R}^K$ iis the k-th standard basis vector.



Pre-train 750M GPT-2 model with GLaM Datasets (6 domains)

Mix strategy typically yields better results

Conceptual PiKE

Minimizing the RHS of previous Theorem and relaxing $w_k = b_k/b$ gives us the **Conceptual PiKE**

Theorem (Tightness, Informal)

Assume task gradients are L-Lipschitz, unbiased, have bounded variance and when $\underline{c}$ and $\bar{c}$ are small

There exist loss functions $\{\ell_k(\cdot,\cdot)\}_{k=1}^{K}$, where those upper bound in previous theorem is tight.

Conceptual PiKE maximizes a **tight** lower bound on the objective decrease amount

Theorem (Convergence, Informal)

Assume task gradients are L-Lipschitz, unbiased, have bounded variance, $\underline{c}$-conflicted and $\bar{c}$-aligned and running **Conceptual PiKE** with with SGD at θ_0. Let $\Delta_L = \mathcal{L}(\boldsymbol{\theta}_0) - \min_{\boldsymbol{\theta}} \mathcal{L}(\theta)$ and then after $T = \frac{2\Delta_L}{\eta\beta\epsilon}$ iterations, **Conceptual PiKE** finds a point $\bar{\theta}$ such that

$$\mathbb{E}\left\|\nabla\mathcal{L}_k(\bar{\theta})\right\|^2 \leq \epsilon, \quad \forall k = 1, \ldots, K.$$

Conceptual PiKE

Remark 1.

The convergence rate of Conceptual PiKE is $T = O\left(1/\epsilon^2\right)$

Match the optimal rate for smooth, nonconvex stochastic optimization

Remark 2.

when $\underline{c}$ and $\bar{c}$ are small

We shows **improved iteration complexity** than Uniform (fixed) data mixing strategy

$$\bar{T}_{\text{Uniform}} = \frac{2L\Delta_L\left(\epsilon + \frac{\sigma^2}{b}K\right)}{\epsilon^2}. \qquad \text{v.s.} \qquad \bar{T}_{\text{PiKE}} = \frac{2L\Delta_L\left(\epsilon + \sigma^2_{\max}/b\right)}{\epsilon^2}$$

Conceptual PiKE requires estimating **Magnitude** and **variance** of per task's gradients **every step**

Not Practical!!!