TiKMiX: Efficient Semi-Dynamic Data Mixture via Data Influence for LLM Pre-training

Yifan Wang, Binbin Liu, Fengze Liu, Yuanfan Guo,
Jiyao Deng, Xuecheng Wu, Weidong Zhou, Xiaohuan Zhou*, Taifeng Wang
ByteDance
yifanyfwang@gmail.com, zhouxiaohuan@bytedance.com

Abstract

The data mixture used in the pre-training of a language model is a cornerstone of its final performance. Static data mixing strategies in Large Language Model (LLM) pre-training are often suboptimal as they fail to adapt to the model's evolving learning states. Conversely, fully online dynamic updates, while adaptive, incur prohibitive computational costs. To bridge this gap, we propose TiKMiX, an efficient semi-dynamic data mixing framework. Our approach is grounded in a key observation of influence ranking invariance: the relative importance of data domains exhibits strong temporal stability over long training intervals. Leveraging this insight, we propose Group Influence, an efficient approach for quantifying domain impact, and formulate data mixing as a periodic, low-overhead influence maximization problem. Compared with REGMIX, the proposed method reduces computational overhead by 80% and achieves an average performance gain of 2% across nine downstream benchmarks, thereby effectively mitigating data under-digestion.

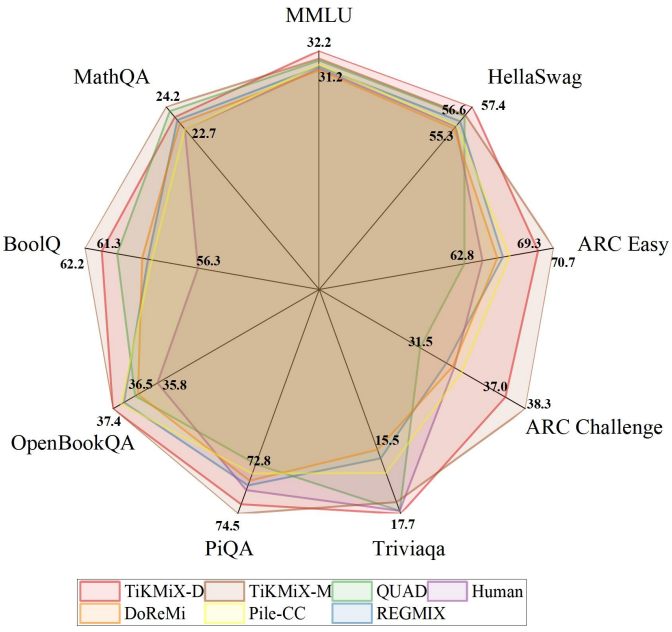

Figure 1: Performance comparisons of our TiKMiX with current state-of-the-art data mixing strategies for pre-training a 1B parameter Language Model with 1T tokens.

1 Introduction

The availability of large-scale public datasets has been a key factor in the creation of LLMs. The pre-training data for LLMs is predominantly sourced from the internet (Wettig et al., 2025; Yu et al., 2025a), encompassing a wide range of materials such as academic papers (Tirumala et al., 2023), books (Tirumala et al., 2023), and more. The mixture ratio of data from different domains plays a crucial role in determining the capabilities of large language models (LLMs) (Zhang et al., 2025b; Liu et al., 2025b). For example, the developers of GPT-3 (Floridi and Chiriatti, 2020) regard Wikipedia as a source of exceptionally high-quality data and increase its proportion within the training dataset. However, relying on such manual heuristics is often suboptimal and labor-intensive. To address this, research has shifted towards automated data mixing strategies. Representative methods like REGMIX (Liu et al., 2024) leverage results from small-scale experiments to automatically optimize mixing ratios.

Prior research heavily relies on small proxy models to derive static data weights (Xie et al., 2023; Fan et al., 2023; Team, 2024b) or employs clustered importance sampling to efficiently align generalist data with target tasks (Grangier et al., 2024). However, these static or one-off selection strategies are not only computationally expensive due to the need for extensive proxy pre-training or domain-specific statistics, but also fundamentally limited by scale mismatch: proxies and static statistics often

*Corresponding author.
Yifan Wang completed this work during an internship at ByteDance.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5777–5793

fail to accurately approximate the evolving preferences of larger target models (Liu et al., 2024), inevitably leading to suboptimal alignment. Conversely, fully online approaches and unified optimization frameworks (Albalak et al., 2023; Chen et al., 2024a) attempt to capture these dynamics through real-time adjustment and improved parameter estimation. Yet, their requirement for frequent, iterative updates introduces complex architectural dependencies and prohibitive computational overhead that scale poorly with modern pre-training demands (Jin et al., 2024; Wang et al., 2025). Given that leading LLMs (Team, 2025b,a, 2024a) typically employ multi-stage training without efficient re-weighting mechanisms, a critical gap emerges between static rigidity and dynamic cost. This raises a pivotal research question: *Is it possible to devise a semi-dynamic strategy that rapidly computes optimal data mixtures capable of remaining effective over extended training intervals, thereby achieving both accurate preference alignment and computational efficiency?*

Recent quantitative frameworks, such as Data Mixing Laws(Ye et al., 2024), optimize data mixtures by modeling the relationship between mixture proportions and validation loss. However, relying solely on loss can be unstable, particularly when extrapolating from early training stages. In contrast, influence functions offer a robust alternative, exhibiting strong rank invariance and high correlation with final performance(Koh et al., 2019). To leverage this stability without incurring prohibitive computational costs, we introduce Group Influence, an efficient metric to quantify the long-term benefits of data sources. Building on this, we propose TiKMiX, which formulates dynamic data mixing as an influence maximization problem. We develop two variants: TiKMiX-D, which directly determines optimal ratios by optimizing weighted influence sums; and TiKMiX-M, an advanced approach that refines the TiKMiX-D initialization via regression-based perturbation modeling to predict the globally optimal mixture.

With the proposed TiKMiX framework, we are able to dynamically adjust the data mixture strategy throughout the entire pre-training cycle, adapting to changes in both model scale and training stage. In line with previous work (Bai et al., 2024; Kang et al., 2024; Diao et al., 2025; Tao et al., 2025), we conducted experiments on models with varying parameter sizes and scaled training up to 1 trillion tokens. TiKMiX-D surpasses state-of-the-art methods such as REGMIX, achieving comparable or superior performance while requiring only 20% of the computational resources. TiKMiX-M further yields an average performance improvement of 2% across nine downstream benchmarks, as illustrated in Fig. 1. Additionally, we discuss the feasibility and implications of applying TiKMiX to even larger-scale models. Our experiments reveal several key findings: (1) a model's data preferences evolve as training progresses; (2) models of different scales exhibit distinct patterns of preference change; (3) dynamic adjustment of the data mixture facilitates more comprehensive learning of the data by the model. In summary, the main contributions of this paper are as follows:

- We propose **Group Influence**, a novel and efficient method for observing and quantifying the dynamic preferences of Large Language Models for different data domains during the pre-training process.
- We designed **TiKMiX**, a dynamic data mixture framework that leverages the observations from Group Influence to adaptively adjust data ratios, aiming to balance the model's performance across multiple tasks.
- Extensive experiments demonstrate that our method not only significantly enhances model performance but also provides profound insights into how a model's data preferences evolve with the training process and model scale, thereby validating the effectiveness of dynamically adjusting data proportions.

2 Related Work

2.1 Influence Function

Influence Functions offer a mathematically grounded method to estimate the effect of training data on model predictions without costly retraining (Koh and Liang, 2017). Their application to high-dimensional models like Large Language Models (LLMs) has been hampered by the computational challenge of inverting the Hessian matrix. Recent work has overcome this barrier through scalable approximation techniques. Notably, the work by Anthropic (Grosse et al., 2023) adapted EK-FAC (George et al., 2018), an efficient Hessian approximation, to successfully apply influence functions to 50B-parameter Transformer models. This breakthrough established influence functions

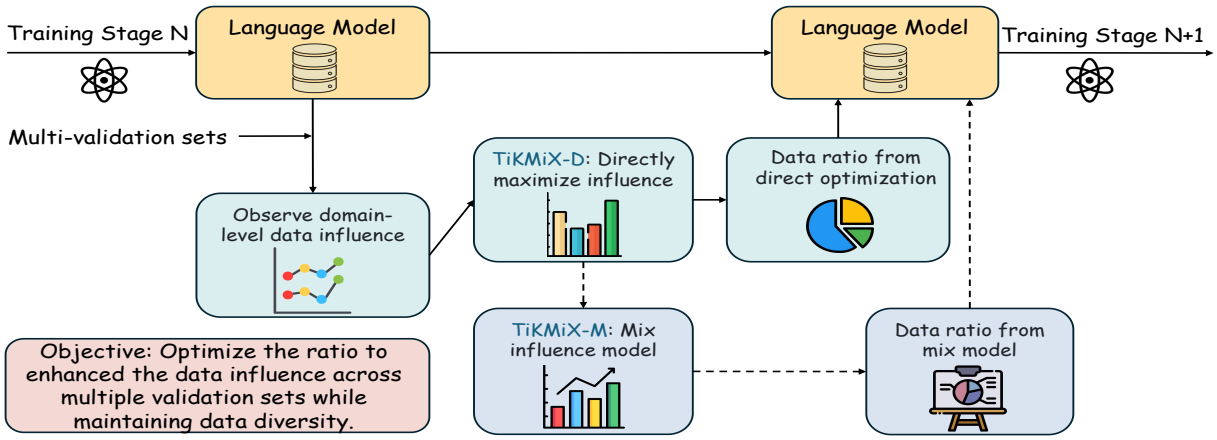

Figure 2: The process involves periodically measuring domain contributions via Group Influence and adjusting the data mixture to maximize learning efficiency.

as a viable tool for performing data attribution at the scale of modern LLMs, enabling the identification of specific pre-training data that drives model outputs (Kou et al., 2025; Choe et al., 2024; Lin et al., 2024a). However, computation at the sample level incurs prohibitive overhead in large-scale pre-training scenarios. Therefore, we propose Group Influence, which extends influence functions to groups of data. By leveraging gradient accumulation techniques, Group Influence can efficiently evaluate the collective impact of the data domain with relatively low computational cost. This allows us to quantify the model's current data preferences.

2.2 Data Selection and Mixing

Strategic curation of training data significantly enhances model performance (Koh and Liang, 2017; Albalak et al., 2023). For pre-training Large Language Models (LLMs), data curation methods are commonly categorized by granularity: **Token-level Selection:** The most fine-grained approach, which filters individual tokens according to specific criteria (Lin et al., 2024b). **Sample-level Selection:** Methods include heuristic-based approaches (Sharma et al., 2025; Soldaini et al., 2024) and learning-based techniques employing optimization algorithms (Chen et al., 2024b; Shao et al., 2024). Additionally, approaches such as MATES (Yu et al., 2024) utilize model-derived signals to inform selection (Marion et al., 2023; Ankner et al., 2024). **Group-level Selection:** Earlier work relied on manually defined ratios, while recent advances favor learning-based strategies. Offline methods like REGMIX (Liu et al., 2024) and DoReMi (Xie et al., 2023) use proxy models to assign static group weights, whereas dynamic methods such as Quad (Zhang et al., 2025a) and ODM (Albalak et al., 2023) iteratively adjust weights during training. Current mainstream pre-training pipelines are typically divided into multiple stages but often lack a mechanism to dynamically adjust the data mixture ratio based on the model's state in different stages. Our proposed method, TiKMiX, is a semi-offline, group-level selection approach that dynamically adjusts the data mixture ratio across multiple training stages. Unlike fully dynamic methods that require repeated iterative updates, TiKMiX directly optimizes the mixture ratio based on the model's current data preferences, enabling efficient adaptation without multiple rounds of adjustment.

3 Methodology

In this section, we introduce TiKMiX, a framework for dynamically optimizing the data mixture during large language model pre-training, as shown in Fig. 2. To capture the granular characteristics of domain-level contributions, we propose Group Influence. As illustrated in Fig. 3, utilizing this metric allows for explicitly observing how the impact of different domains evolves throughout the training process. By formulating the dynamic data mixture problem as an optimization task aimed at maximizing Group Influence, we develop two distinct methods: TiKMiX-D, which directly optimizes based on influence scores, and TiKMiX-M, which leverages a regression model for a computationally efficient approximation. We begin by defining the problem setup and Group Influence before detailing these two optimization strategies.

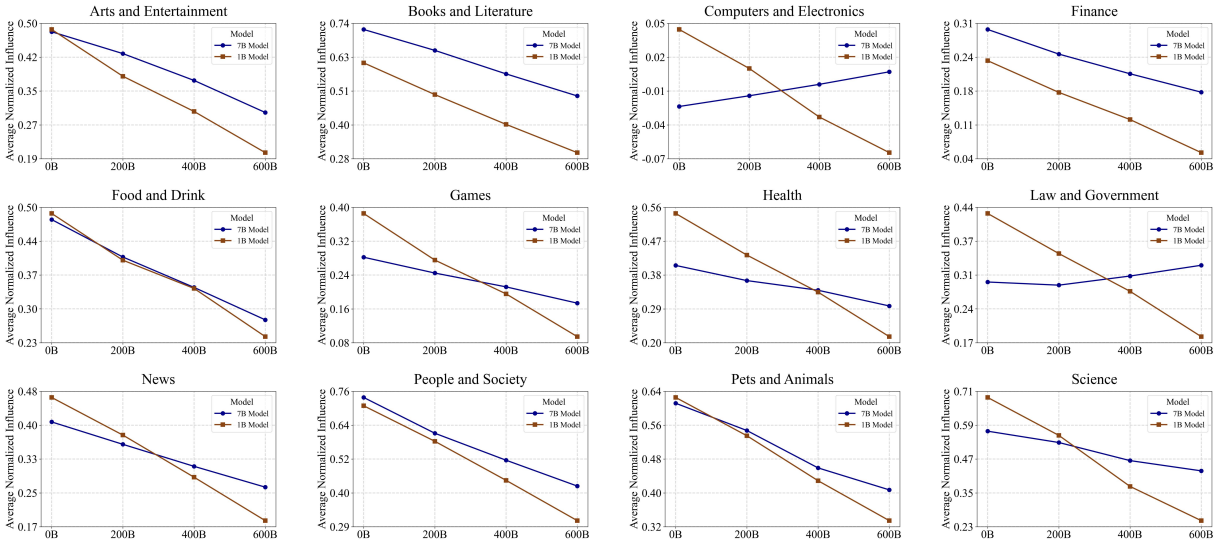

Figure 3: The influence of different domains on the validation set as the model training progresses.

3.1 Group Influence

Existing methods of data mix method like Data Mixing Laws(Ye et al., 2024) rely on validation loss as a direct proxy for data utility. However, loss trajectories can be volatile during the early stages of training, making it difficult to predict the long-term contribution of specific data sources to the final model performance. Unlike transient loss values, influence functions quantify the counterfactual effect of training data on validation performance. Crucially, empirical studies demonstrate that influence scores exhibit strong rank invariance (Koh et al., 2019): the relative importance of data sources tends to remain consistent throughout the training process. This property allows us to estimate the final utility of a domain accurately.

However, applying influence functions at the instance level is computationally prohibitive for large-scale pre-training due to the cost of Hessian-vector products. Furthermore, instance-level granularity often fails to capture the collective semantics of a data source. To address these challenges, we generalize the framework to Group Influence. While standard influence targets individual points, Group Influence aggregates gradient information over cohesive subsets of data. This formulation provides two distinct advantages: it significantly reduces computational overhead by avoiding per-sample calculations, and it better encapsulates domain-level characteristics—such as topic distribution and stylistic patterns—that emerge only from the collective effect of semantically related instances.

Derivation. Let $\mathcal{D}_{\text{train}} = \{z_i\}_{i=1}^{N}$ denote the training dataset, where each sample $z_i = (x_i, y_i)$ consists of an input and its corresponding label. We identify a target subset $S \subseteq \mathcal{D}_{\text{train}}$ representing a coherent group of instances, such as a specific data domain or a semantic cluster. To rigorously quantify the collective impact of this group on the learned representations, we adopt a counterfactual perspective: we examine how the optimal model parameters would shift if the influence of S were marginally amplified. Formally, we model this scenario as a perturbed empirical risk minimization problem, where the aggregate loss contribution of the subset S is up-weighted by a small scalar ϵ:

$$\hat{\theta}_{\epsilon,S} = \arg\min_{\theta\in\Theta} \left(\frac{1}{N} \sum_{z_i \in \mathcal{D}_{\text{train}}} \mathcal{L}(z_i, \theta) + \epsilon \sum_{z_j \in S} \mathcal{L}(z_j, \theta) \right). \tag{1}$$

The optimal parameters $\hat{\theta}_{\epsilon,S}$ satisfy the first-order stationarity condition $\nabla_\theta \mathcal{L}_{\text{total}}(\hat{\theta}_{\epsilon,S}) = 0$. Applying the Implicit Function Theorem around the unperturbed solution $\theta^* = \hat{\theta}_{0,S}$, we derive the sensitivity of the parameters with respect to ϵ:

$$\left.\frac{d\hat{\theta}_{\epsilon,S}}{d\epsilon}\right|_{\epsilon=0} = -H_{\theta^*}^{-1} \sum_{z_j \in S} \nabla_\theta \mathcal{L}(z_j, \theta^*) \triangleq -H_{\theta^*}^{-1} g_S. \tag{2}$$

where $H_{\theta^*} = \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta^2 \mathcal{L}(z_i, \theta^*)$ is the Hessian of the training objective, and g_S represents the accumulated gradient of the group S. Consequently,

the influence of group S on a validation objective $f(\theta) = \mathcal{L}(\mathcal{D}_{\text{val}}, \theta)$ is derived via the chain rule:

$$\mathcal{I}(S) \triangleq \left.\frac{df(\hat{\theta}_{\epsilon,S})}{d\epsilon}\right|_{\epsilon=0} = -\nabla_\theta f(\theta^*)^\top H_{\theta^*}^{-1} g_S. \tag{3}$$

Equation 3 demonstrates that calculating Group Influence is mathematically equivalent to summing the individual influence scores of all instances in S, owing to the linearity of the inverse-Hessian operator.

Computational Efficiency Analysis. The computational tractability of influence functions is governed by the Inverse-Hessian-Vector Product (IHVP). Let C_{HVP} denote the cost of a single Hessian-vector product (typically approximated via LiSSA or conjugate gradient), and C_{grad} denote the cost of a gradient backpropagation. A naive aggregation of instance-level influence requires computing the IHVP for each sample in the subset, resulting in a complexity of:

$$\mathcal{T}_{\text{naive}} = |S| \times \left(C_{\text{HVP}} + C_{\text{grad}}\right) \approx \mathcal{O}(|S| \cdot C_{\text{HVP}}). \tag{4}$$

In contrast, our formulation (Eq. 3) exploits the distributivity of the linear operator. By pre-accumulating the gradients into a single vector g_S, we perform the expensive IHVP operation only once:

$$\mathcal{T}_{\text{group}} = 1 \times C_{\text{HVP}} + |S| \times C_{\text{grad}} \approx \mathcal{O}(C_{\text{HVP}}). \tag{5}$$

Given that $C_{\text{HVP}} \gg C_{\text{grad}}$ in deep models, the proposed Group Influence achieves a theoretical speedup factor of approximately $|S|$. This efficiency renders the analysis of large-scale pre-training corpora (where $|S| \sim 10^6$) computationally feasible.

3.2 TiKMiX-D: Directly maximize influence

Based on the Group Influence metric, which quantifies the effect of each data domain on model performance, we aim to optimize the data mixture by determining a weight vector w that maximizes overall influence. We propose TiKMiX-D, which formulates this as a multi-objective optimization problem, dynamically adjusting w during training to balance performance and maintain data diversity. The Group Influence scores are organized into an $n \times m$ matrix S, where n is the number of validation tasks and m is the number of data domains, with S_{ij} denoting the influence of domain d_j on task v_i. The expected influence for each task is $P = S \cdot w$, and to ensure comparability across tasks, we normalize as follows:

$$\hat{P}_i = \frac{P_i}{\max_j S_{ij} + \epsilon}, \tag{6}$$

ϵ denotes a small positive constant added for numerical stability. The optimization objective of TiKMiX-D is defined as a unified function $L(w)$ that integrates three components: (1) **influence uniformity**, measured by the standard deviation $\text{std}(\hat{P})$, promoting balanced improvements across tasks; (2) **overall influence gain**, quantified by the sum $\sum \hat{P}_i$, to maximize aggregate performance; and (3) **data diversity**, measured by the entropy $H(w) = -\sum_{j=1}^{m} w_j \log(w_j)$, encouraging a uniform weight distribution. The trade-offs among these objectives are controlled by hyperparameters α, β, and γ, which are set to 1 in our experiments for equal weighting.

The complete optimization problem is subject to several constraints to ensure a valid and beneficial solution. The weights must be non-negative ($w_j \geq 0$) and sum to one ($\sum w_j = 1$). Furthermore, to guarantee continuous improvement, we enforce a Pareto improvement constraint, ensuring that the influence generated by the new mixture w is no less than that of the prior mixture w_{prior} for any task, i.e., $S \cdot w \geq S \cdot w_{\text{prior}}$. This leads to the final constrained non-linear optimization problem:

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \alpha \cdot \text{std}(\hat{P}) - \beta \sum_{i=1}^{n} \hat{P}_i - \gamma \cdot H(w) \\ \text{subject to} \quad & \sum_{j=1}^{m} w_j = 1, \\ & w_j \geq 0, \quad \forall j \in \{1, \ldots, m\}, \\ & S \cdot w \geq S \cdot w_{\text{prior}}. \end{aligned} \tag{7}$$

We employ the Sequential Least Squares Quadratic Programming algorithm (Gupta and Gupta, 2018) to solve this problem, initializing the weights with a uniform distribution. The resulting optimal vector, w_{best}, serves as the dynamic data mixture for the subsequent training stage.

3.3 TiKMiX-M: Mix influence model

While TiKMiX-D provides an efficient strategy for data mixing through direct optimization, it operates on the assumption that the influences of data domains are linearly additive. This simplification may

overlook non-linear cross-domain interactions that arise when different data sources are combined. To better capture these mixture effects, we introduce TiKMiX-M, which optimizes mixture proportions by modeling interactions within domain mixtures.

To explore the model's performance across a diverse range of domain weightings, we generate a set of N candidate mixture vectors. Our approach is anchored by an empirically determined prior weight vector, $w_{\text{orig}} \in \mathbb{R}^D$, where D is the number of domains. For each domain i, we define a plausible sampling interval by scaling the original weight. We employ Latin Hypercube Sampling (Loh, 2021) within this D-dimensional hyperrectangle to efficiently generate candidate vectors, ensuring a uniform of the parameter space.

Each candidate vector w_{cand} is subsequently normalized to satisfy the constraint ($\sum_{i=1}^{D} w_i = 1$), yielding a normalized vector $w_{\text{norm}} = w_{\text{cand}} / \sum_{j=1}^{D} w_{\text{cand},j}$. However, this normalization can shift components outside their predefined intervals. Therefore, we implement a rejection sampling scheme, where a normalized vector w_{norm} is accepted into our final set only if it satisfies the boundary constraints for all dimensions, i.e., $w_{\text{norm},i} \in [l_i, h_i]$ for all $i \in \{1, \ldots, D\}$. This iterative process is repeated until N valid weight vectors that meet both the summation and boundary conditions have been collected, resulting in a robust and well-distributed set of weights for subsequent analysis. For each generated candidate mixture w_i, we calculate its true aggregate influence score, y_i, across all validation sets using the Group Influence evaluation method $\sum \hat{P}_i$.

Following these steps, we obtain a training set $D_{\text{train}} = \{(w_i, y_i)\}_{i=1}^{N}$. Inspired by REGMIX(Liu et al., 2024), we select LightGBM (Ke et al., 2017), an efficient gradient boosting decision tree model, as our regression surrogate. This model, f_{LGBM}, is trained to predict the aggregate influence y for given data mixture w, i.e., $y = f_{\text{LGBM}}(w)$.We leverage it to efficiently explore the mixture space without performing expensive, true influence evaluations. We design an iterative search algorithm that balances exploration and exploitation to find the optimal mixture.

The process is detailed in Algorithm 1. We start from the ratio from TiKMiX-D, $w_{\text{best-D}}$. At each step, we sample candidate mixtures on the current best solution. The distribution's concentration parameter is annealed over steps, beginning with a large value to encourage global exploration and gradually decreasing to promote local exploitation near the optimum. We employ the surrogate model to evaluate all sampled candidates. The center for the next iteration is then updated to be the average of the top-k candidates with the highest predicted scores. This procedure is repeated until convergence or a maximum number of iterations is reached. TiKMiX-M not only accounts for non-linear cross-domain interactions but also significantly enhances search efficiency through the surrogate model, enabling it to discover superior solutions within the vast mixture space.

Algorithm 1 Iterative Search via TiKMiX-M

Input: Surrogate f_{sur}, initial mix $w^{(0)}$, iters T, samples N, exploration $[\alpha_{\min}, \alpha_{\max}]$, top-$k$.
Output: Optimized mixture w^*.
$w_{\text{best}} \leftarrow w^{(0)}$
Generate exploration strengths $\{\alpha_t\}_{t=1}^{T}$
for $t = 1$ to T **do**
  Sample N domain mixture candidates $\{w_i\}$
  compute the Group influence score of w_i.
  Select indices $I_{\text{top-k}}$ of k mixtures with highest Group influence scores.
  Update $w_{\text{best}} = \frac{1}{k} \sum_{i \in I_{\text{top-k}}} w_i$.
end for
return w_{best}

4 Experiments

This section presents a comprehensive set of experiments designed to validate the effectiveness of our TiKMiX framework. We first outline the experimental setup, including evaluation benchmarks, datasets, and baseline methods. Subsequently, we demonstrate that: (1) the pre-training data mixture significantly impacts downstream task performance; (2) our proposed Group Influence is an effective predictor of downstream performance; and (3) the TiKMiX framework, particularly TiKMiX-D and TiKMiX-M, markedly improves model performance and surpasses existing SOTA methods.

4.1 Experimental Setup

Datasets and Models Data mixture design for web-scale corpora has become an important component in large language model pre-training, as it directly affects both data utilization efficiency and downstream generalization performance. However, due to the substantial diversity and heterogeneity of web data, identifying an appropriate mixture

remains a nontrivial problem. To study this issue under a controlled setting, we conduct experiments on the RefinedWeb dataset (Penedo et al., 2023), which contains 26 distinct data domains. Following prior work, we adopt the model architecture introduced by (Zhang et al., 2024), and instantiate models at the 1B and 7B scales. Unless otherwise specified, pre-training is conducted on up to 1 trillion tokens.

We compare TiKMiX with several representative data mixing methods. **REGMIX** (Liu et al., 2024) learns a regression model to estimate validation loss and uses it to optimize the mixture ratio. **DoReMi** (Xie et al., 2023) is a representative dynamic data reweighting approach that relies on a proxy model for mixture adaptation. **QUAD** (Zhang et al., 2025a) performs dynamic data selection during training based on clustered data groups. All baselines are implemented following their original settings. For TiKMiX, we adopt a semi-dynamic training protocol consistent with its design principle. Specifically, we divide the full training process into two stages according to the total token budget, and update the data mixture once at the stage boundary using the proposed algorithm.

Downstream Task Evaluation To comprehensively evaluate our proposed method, we curated a diverse set of 9 widely recognized downstream benchmarks, which were strategically divided into two categories: in-domain and out-of-domain. This division allows for a rigorous assessment of both the model's core capabilities and its generalization prowess. Our **in-domain** evaluation suite was designed to cover a wide spectrum of reasoning and knowledge-based tasks. It includes **MMLU** (Hendrycks et al., 2020), a challenging benchmark measuring knowledge; **HellaSwag** (Zellers et al., 2019), a commonsense reasoning task that involves choosing the most plausible continuation for a given context; **ARC** (Clark et al., 2018), which we evaluate on both the Easy (**ARC-E**) and the more difficult Challenge (**ARC-C**) sets of grade-school science questions; and **TriviaQA** (Joshi et al., 2017), a reading comprehension benchmark requiring models to locate answers within lengthy documents. To evaluate the generalization capabilities of our method, we selected a set of out-of-domain benchmarks. These include **PiQA** (Bisk et al., 2020), a commonsense benchmark focused on physical interactions; **OpenBookQA** (Mihaylov et al., 2018), a question-answering task requiring reasoning over a given set of science facts; **BoolQ** (Clark et al., 2019), a dataset of naturally occurring yes/no questions; and **MathQA** (Amini et al., 2019), a mathematical reasoning benchmark with multi-step problems.

4.2 Group Influence as an Effective Predictor of Performance

The core hypothesis of our proposed TiKMiX framework is that maximizing Group Influence can effectively enhance overall downstream task performance. As a preliminary observation, Fig. 3 illustrates the evolution of influence across different domains during the 600B-token training process. To empirically validate our hypothesis, we calculated the impact of 10 different data mixtures on various benchmarks. Specifically, we trained a 1B-parameter model on 500B tokens using these mixtures. The normalized scores are presented in Fig. 4 in the Appendix. We observe a strong positive correlation (*i.e.*, Pearson correlation coefficient $\rho = 0.789$) between the total Group Influence and the average downstream scores. This indicates that mixtures generating stronger total influence almost invariably lead to better downstream performance. This finding not only confirms the validity of Group Influence as an optimization target but also provides a solid theoretical foundation for the design of TiKMiX-D and TiKMiX-M.

Building on the preceding findings, we formally evaluate the two implementations of our TiKMiX framework: TiKMiX-D and TiKMiX-M. We first followed the natural data distribution, then, using TiKMiX adjusted the data mixture between the two stages during the 1T-token pre-training process. As shown in Table 1, both of our methods significantly outperform all baselines. On average, across 9 benchmarks, TiKMiX-D and TiKMiX-M improved performance by **1.6%** and **2.0%**, respectively, over the strongest baseline, REGMIX. Notably, on challenging tasks like ARC Easy and ARC Challenge, TiKMiX-M achieved a performance advantage of over 4.8%. The results of experiments conducted on larger-scale models are provided in Table 3.

4.3 Analysis of Computational Efficiency

The exact computation of the Hessian matrix in LLMs typically incurs extremely high computational costs. To mitigate this overhead, we draw upon recent studies on influence functions in LLMs(Grosse et al., 2023) and employ the Empirical Kronecker-Factored Approximate Curvature

Benchmark	Human	DoReMi	Average	QUAD	REGMiX	TiKMiX-D	TiKMiX-M
In-Domain Benchmarks							
MMLU (Hendrycks et al., 2020)	31.3	31.2	30.9	31.7	31.5	**32.2**	31.8
HellaSwag (Zellers et al., 2019)	55.5	55.3	55.9	56.5	56.0	**57.4**	56.6
ARC Easy (Clark et al., 2018)	64.4	65.7	64.1	62.8	66.2	69.3	**70.7**
ARC Challenge (Clark et al., 2018)	33.7	33.6	32.1	33.5	33.2	37.0	**38.3**
Triviaqa (Joshi et al., 2017)	17.6	15.5	17.3	17.6	15.8	**17.7**	17.3
Out-of-Domain Benchmarks							
PiQA (Bisk et al., 2020)	73.5	73.1	71.5	72.4	73.3	74.1	**74.5**
OpenBookQA (Mihaylov et al., 2018)	35.8	36.5	34.6	36.6	37.0	**37.4**	**37.4**
Boolq (Clark et al., 2019)	56.3	59.2	58.3	60.5	58.9	61.3	**62.2**
MathQA (Amini et al., 2019)	22.7	23.1	23.7	23.9	23.3	23.5	**24.2**
Estimated FLOPs	0	4.2e19	0	2.3e18	3.7e18	7.2e17	3.2e18
Average Perf.	43.4	43.7	43.2	43.9	43.9	45.5	45.9
Best On	0/9	0/9	0/9	0/9	0/9	4/9	6/9

Table 1: Comparison of 1B Parameter Models Trained on 1T Tokens Across Various Benchmarks. The best-performing model on each benchmark is highlighted in bold.

Table 2: The ablation study of Loss and TiKMiX on different data sizes.

Benchmark	Loss		TiKMiX	
	5B	10B	0.1B	0.5B
In-Domain Benchmarks				
MMLU (Hendrycks et al., 2020)	31.4	31.2	32.2	32.1
HellaSwag (Zellers et al., 2019)	56.3	56.4	57.4	57.6
ARC Easy (Clark et al., 2018)	67.3	65.6	69.3	69.1
ARC Challenge (Clark et al., 2018)	34.4	33.4	37.0	37.1
TriviaQA (Joshi et al., 2017)	16.5	16.9	17.7	17.9
Out-of-Domain Benchmarks				
PiQA (Bisk et al., 2020)	73.2	73.5	74.1	74.2
OpenBookQA (Mihaylov et al., 2018)	36.4	36.6	37.4	37.3
BoolQ (Clark et al., 2019)	59.4	59.7	61.3	61.5
MathQA (Amini et al., 2019)	23.9	23.7	23.5	23.6
Average Perf.	44.3	44.1	45.5	45.6

(EKFAC) method to approximate the Hessian matrix. EKFAC reduces computational and memory requirements by partitioning the Hessian and applying Kronecker factorization, thereby transforming complex high-dimensional matrix operations into computations within lower-dimensional subspaces.

Consequently, TiKMiX demonstrates superior computational efficiency. In contrast to methods such as MATES(Yu et al., 2024), Group-MATES(Yu et al., 2025b), and REGMIX, which require the additional overhead of training small proxy models, the Group Influence calculation and optimization process in TiKMiX is highly efficient and does not involve such auxiliary training procedures. In our 1B model experiments, the total computational overhead for **TiKMiX-D** to determine the next-stage mixture (including influence calculation and regression model inference) was only about **20%** of that required by the **RegMix** method, while achieving comparable or even superior performance. This high efficiency makes TiKMiX a practical and powerful tool for large-scale LLM training.

4.4 Ablation Study

We conduct a series of ablation studies, with the results presented in Table 2. Our primary investigation focused on the efficacy of using **group influence** and **TiKMiX** for preference observation and data mixture adjustments. As shown in Table 2, our approach allows for the accurate observation of model preferences using only 0.1B tokens and requires no model training, leading to a significant performance improvement over the loss. This highlights the superiority of our method in efficiently identifying and correcting data biases. Additionally, we find that setting hyperparameters α, β, γ to 1 yields optimal balance between in-domain and out-of-domain performance. We attribute this to the limitations of current validation sets in capturing the model's full capabilities, necessitating explicit diversity constraints to prevent overfitting to specific validation metrics. We further discuss the effectiveness of our model on a larger scale in the appendix.

5 Conclusion and Discussions

In this work, we introduce TiKMiX, a novel framework that dynamically adjusts the data mixture based on Group Influence, a highly efficient metric to evaluate the contribution of data domains to the model's performance. By framing data mixing

as an influence-maximization problem, we developed two approaches: TiKMiX-D, which directly optimizes the mixture and surpasses state-of-the-art methods like REGMIX using only 20% of the computational resources, and TiKMiX-M, which uses a regression model to predict superior mixtures, achieving an average performance gain of 2% across 9 downstream benchmarks. We plan to conduct further experiments on larger-scale models and more diverse datasets to further validate the effectiveness of Group Influence and TiKMiX.

6 Limitations

While TiKMiX demonstrates superior efficiency and performance in semi-dynamic data mixing, we acknowledge two main avenues for further exploration. First, due to computational resource constraints, our empirical validation is primarily conducted on academic-scale models and datasets. Although the consistent performance gains across diverse benchmarks strongly suggest the generalizability of our approach, evaluating TiKMiX on larger-scale foundational models (e.g., >7B parameters) with trillion-token corpora remains a critical next step to verify its scalability and impact on emergent abilities. Second, our current work focuses on the efficacy of *Group Influence* for periodic mixture adjustments. We have not yet exhaustively explored the theoretical underpinnings of data mixing from the perspective of microscopic parameter evolution and gradient dynamics. Future work could investigate the correlation between data mixture ratios and model weight updates, aiming to provide a more rigorous interpretability for dynamic data scheduling and potentially unlock finer-grained, continuous adjustment strategies.

References

Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. 2023. Efficient online data mixing for language model pre-training. *arXiv preprint arXiv:2312.02406*.

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

Zachary Ankner, Cody Blakeney, Kartik Sreenivasan, Max Marion, Matthew L Leavitt, and Mansheej Paul. 2024. Perplexed by perplexity: Perplexity-based data pruning with small reference models. *arXiv preprint arXiv:2405.20541*.

Tianyi Bai, Ling Yang, Zhen Hao Wong, Jiahui Peng, Xinlin Zhuang, Chi Zhang, Lijun Wu, Jiantao Qiu, Wentao Zhang, Binhang Yuan, and Conghui He. 2024. Multi-agent collaborative data selection for efficient LLM pretraining. *CoRR*, abs/2410.08102.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Mayee F Chen, Michael Y Hu, Nicholas Lourie, Kyunghyun Cho, and Christopher Ré. 2024a. Aioli: A unified optimization framework for language model data mixing. *arXiv preprint arXiv:2411.05735*.

Xuxi Chen, Zhendong Wang, Daouda Sow, Junjie Yang, Tianlong Chen, Yingbin Liang, Mingyuan Zhou, and Zhangyang Wang. 2024b. Take the bull by the horns: Hard sample-reweighted continual training improves llm generalization. *arXiv preprint arXiv:2402.14270*.

Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Minsoo Kang, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, Jeff G. Schneider, Eduard H. Hovy, Roger B. Grosse, and Eric P. Xing. 2024. What is your data worth to gpt? llm-scale data valuation with influence functions. *CoRR*, abs/2405.13954.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan Su, Markus Kliegl, Zijia Chen, Peter Belcak, Yoshi Suhara, Hongxu Yin, Mostofa Patwary, Yingyan Lin, Jan Kautz, and Pavlo Molchanov. 2025. CLIMB: clustering-based iterative data mixture bootstrapping for language model pre-training. *CoRR*, abs/2504.13161.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. 2023. Doge: Domain reweighting with generalization estimation. *arXiv preprint arXiv:2310.15393*.

Luciano Floridi and Massimo Chiriatti. 2020. Gpt-3: Its nature, scope, limits, and consequences. *Minds and machines*, 30(4):681–694.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Thomas George, César Laurent, Xavier Bouthillier, Nicolas Ballas, and Pascal Vincent. 2018. Fast approximate natural gradient descent in a kronecker factored eigenbasis. *Advances in neural information processing systems*, 31.

David Grangier, Simin Fan, Skyler Seto, and Pierre Ablin. 2024. Task-adaptive pretrained language models via clustered-importance sampling. *arXiv preprint arXiv:2410.03735*.

Roger B. Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamile Lukosiute, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. 2023. Studying large language model generalization with influence functions. *CoRR*, abs/2308.03296.

Madhuri Gupta and Bharat Gupta. 2018. An ensemble model for breast cancer prediction using sequential least squares programming method (slsqp). In *2018 eleventh international conference on contemporary computing (IC3)*, pages 1–3. IEEE.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Xin Jin, Hongyu Zhu, Siyuan Li, Zedong Wang, Zicheng Liu, Juanxi Tian, Chang Yu, Huafeng Qin, and Stan Z Li. 2024. A survey on mixup augmentations and beyond. *arXiv preprint arXiv:2409.05202*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Feiyang Kang, Yifan Sun, Bingbing Wen, Si Chen, Dawn Song, Rafid Mahmood, and Ruoxi Jia. 2024. Autoscale: Automatic prediction of compute-optimal data compositions for training llms.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR.

Pang Wei W Koh, Kai-Siang Ang, Hubert Teo, and Percy S Liang. 2019. On the accuracy of influence functions for measuring group effects. *Advances in neural information processing systems*, 32.

Siqi Kou, Qingyuan Tian, Hanwen Xu, Zihao Zeng, and Zhijie Deng. 2025. Which data attributes stimulate math and code reasoning? an investigation via influence functions. *arXiv preprint arXiv:2505.19949*.

Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. 2024a. Data-efficient fine-tuning for llm-based recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 365–374.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. 2024b. Rho-1: Not all tokens are what you need. *CoRR*, abs/2404.07965.

Fengze Liu, Weidong Zhou, Binbin Li, Zhimiao Yu, Yifan Zhang, Haobin Lin, Yifeng Yu, Bingni Zhang, Xiaohuan Zhou, Taifeng Wang, and Yong Cao. 2025a. Quadmix: Quality-diversity balanced data selection for efficient LLM pretraining. *CoRR*, abs/2504.16511.

Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, Yuanxing Zhang, Zhuo Chen, Hangyu Guo, Shilong Li, Ziqiang Liu, Yong Shan, Yifan Song, Jiayi Tian, Wenhao Wu, and 18 others. 2025b. A comprehensive survey on long context language modeling. *CoRR*, abs/2503.17407.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. 2024. Regmix: Data mixture as regression for language model pre-training. *arXiv preprint arXiv:2407.01492*.

Wei-Liem Loh. 2021. On latin hypercube sampling. *The annals of statistics*.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. 2023. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data only. *Advances in Neural Information Processing Systems*, 36:79155–79172.

Yunfan Shao, Linyang Li, Zhaoye Fei, Hang Yan, Dahua Lin, and Xipeng Qiu. 2024. Balanced data sampling for language model training with clustering. *arXiv preprint arXiv:2402.14526*.

Vasu Sharma, Karthik Padthe, Newsha Ardalani, Kushal Tirumala, Russell Howes, Hu Xu, Po-Yao Huang, Daniel Li Chen, Armen Aghajanyan, Gargi Ghosh, and Luke Zettlemoyer. 2025. Text quality-based pruning for efficient training of language models. *J. Data-centric Mach. Learn. Res.*, 2:(13):1–13.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Raghavi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, and 17 others. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15725–15788. Association for Computational Linguistics.

Zhixu Silvia Tao, Kasper Vinken, Hao-Wei Yeh, Avi Cooper, and Xavier Boix. 2025. Merge to mix: Mixing datasets via model merging. *arXiv preprint arXiv:2505.16066*.

Kimi Team. 2025a. Kimi K2: open agentic intelligence. *CoRR*, abs/2507.20534.

Llama Team. 2024a. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Qwen Team. 2024b. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Qwen Team. 2025b. Qwen3 technical report. *CoRR*, abs/2505.09388.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari Morcos. 2023. D4: Improving llm pretraining via document de-duplication and diversification. *Advances in Neural Information Processing Systems*, 36:53983–53995.

Kun Wang, Guibin Zhang, Zhenhong Zhou, Jiahao Wu, Miao Yu, Shiqian Zhao, Chenlong Yin, Jinhu Fu, Yibo Yan, Hanjun Luo, Liang Lin, Zhihao Xu, Haolang Lu, Xinye Cao, Xinyun Zhou, Weifei Jin, Fanci Meng, Junyuan Mao, Hao Wu, and 63 others. 2025. A comprehensive survey in llm(-agent) full stack safety: Data, training and deployment. *CoRR*, abs/2504.15585.

Alexander Wettig, Kyle Lo, Sewon Min, Hannaneh Hajishirzi, Danqi Chen, and Luca Soldaini. 2025. Organize the web: Constructing domains enhances pre-training data curation. *arXiv preprint arXiv:2502.10341*.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2023. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. 2024. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *arXiv preprint arXiv:2403.16952*.

Shi Yu, Zhiyuan Liu, and Chenyan Xiong. 2025a. Craw4llm: Efficient web crawling for llm pretraining. *arXiv preprint arXiv:2502.13347*.

Zichun Yu, Spandan Das, and Chenyan Xiong. 2024. Mates: Model-aware data selection for efficient pretraining with data influence models. *Advances in Neural Information Processing Systems*, 37:108735–108759.

Zichun Yu, Fei Peng, Jie Lei, Arnold Overwijk, Wentau Yih, and Chenyan Xiong. 2025b. Data-efficient pretraining with group-level data influence modeling. *arXiv preprint arXiv:2502.14709*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Chi Zhang, Huaping Zhong, Kuan Zhang, Chengliang Chai, Rui Wang, Xinlin Zhuang, Tianyi Bai, Jiantao Qiu, Lei Cao, Ju Fan, Ye Yuan, Guoren Wang, and Conghui He. 2025a. Harnessing diversity for important data selection in pretraining large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Xuanliang Zhang, Dingzirui Wang, Longxu Dou, Qingfu Zhu, and Wanxiang Che. 2025b. A survey of table reasoning with large language models. *Frontiers of Computer Science*, 19(9):199348.

A Experimental Setup

Datasets and Models Web data serves as one of the core sources for pre-training large language models (LLMs), playing a crucial role in enhancing model capabilities due to its broad coverage and diversity. However, precisely because web data encompasses a wide range of domains—including news, encyclopedias, forums, and academic content—its highly diverse origins make it extremely challenging to achieve a balanced mixture across different domains. We follow the same experimental setup as prior studies on web data mixture (Wettig et al., 2025; Liu et al., 2025a), utilize the RefinedWeb dataset (Penedo et al., 2023), and employ the domain classifier (He et al., 2023) to categorize the data into 26 distinct domains. Our models, ranging in size from 1B to 7B parameters, are trained on up to 1 trillion tokens. The training process is divided into two distinct stages, each consisting of 500 billion tokens, with a strategic adjustment of the data mixture ratio at the transition point between stages. We compare TiKMiX against several representative data mixing strategies: **Pile-CC (Gao et al., 2021)**: The original data mixture proposed by the authors of The Pile based on heuristics. **REGMIX (Liu et al., 2024)**: SOTA method that uses a regression model to predict and optimize validation loss for determining the mixture. **DoReMi (Xie et al., 2023)**: a classic dynamic data mixing method that relies on a proxy model. **QUAD (Zhang et al., 2025a)**: a method for dynamic selection during training after clustering data We use the best-reported mixture from their paper, re-normalized to the domains available in our setup.

Our proposed TiKMiX method achieves a balance between dynamic adaptability and computational efficiency in data mixture strategies. Similar to other dynamic approaches such as DoReMi and QUAD, TiKMiX adjusts the data mixture ratios according to the current state of the model. However, unlike these methods, TiKMiX does not require multiple iterations, which significantly improves training efficiency. Furthermore, TiKMiX simplifies the data mixing process and reduces engineering complexity without sacrificing model performance.

To systematically evaluate the effectiveness of different data mixing strategies, we conduct large-scale experiments on the RefinedWeb dataset. Our models range in size from 1B to 7B parameters and are trained on up to 1 trillion tokens. The training process is divided into two distinct stages, each consisting of 500 billion tokens. At the transition point between these two stages, we strategically adjust the data mixture ratios to further assess the impact of mixing strategies on model performance.

B Downstream Task Evaluation

To conduct a comprehensive and rigorous evaluation of our proposed method, we curated a diverse suite of nine widely-recognized downstream benchmarks. This evaluation matrix is strategically divided into two categories: **in-domain** and **out-of-domain**. This bifurcation allows for a dual-faceted assessment of our model's capabilities: on one hand, to measure its proficiency on tasks closely aligned with its training objectives, and on the other, to critically examine its ability to generalize learned skills to novel tasks and knowledge domains. The consistent performance gains observed across both categories underscore our method's ability to enhance the model's foundational capabilities and foster robust generalization.

In-Domain Evaluation Our in-domain evaluation suite is designed to probe the model's core competencies in complex reasoning, commonsense understanding, and knowledge-intensive applications. These benchmarks are thematically aligned with our method's primary optimization goals and serve to quantify the depth of improvement in these critical areas.

- **MMLU (Massive Multitask Language Understanding)** (Hendrycks et al., 2020): A highly challenging multitask benchmark that assesses knowledge across 57 disparate subjects, ranging from elementary mathematics and U.S. history to computer science and law. MMLU demands not only a vast repository of knowledge but also the ability to perform precise, domain-specific reasoning, making it a key indicator of a model's comprehensive intellectual and academic capabilities.
- **HellaSwag** (Zellers et al., 2019): A commonsense reasoning benchmark that tasks the model with selecting the most plausible continuation for a given context. HellaSwag is distinguished by its use of adversarially-generated distractors, which are designed to be highly confusable for models that rely on superficial statistical cues. It therefore serves

as a robust test of a model's deeper understanding of causality and everyday situations.

- **ARC (AI2 Reasoning Challenge)** (Clark et al., 2018): This benchmark evaluates reasoning and comprehension on grade-school science questions. We assess performance on both its subsets: **ARC-Easy (ARC-E)**, which contains questions often solvable via information retrieval, and the more difficult **ARC-Challenge (ARC-C)**, which requires multi-step reasoning and synthesis of knowledge. Evaluating on both allows for a fine-grained analysis of the model's capabilities, from basic knowledge retrieval to complex scientific inference.

- **TriviaQA** (Joshi et al., 2017): A large-scale reading comprehension benchmark where questions are authored by trivia enthusiasts, leading to a high degree of diversity and complexity. The task requires models to locate answers within lengthy, evidence-rich documents, often amidst significant distractor information. It primarily evaluates the model's proficiency in long-context processing, precise information retrieval, and fact verification.

Out-of-Domain Evaluation To rigorously assess the generalization power of our method, we selected a set of out-of-domain benchmarks that are distinct from the in-domain tasks in terms of subject matter, format, or required reasoning skills. Performance on these benchmarks directly reflects the model's ability to transfer its learned meta-skills to new and unseen challenges.

- **PiQA (Physical Interaction QA)** (Bisk et al., 2020): A commonsense benchmark focused on physical reasoning. Presented in a question-answering format, it requires the model to understand the properties and affordances of everyday objects (e.g., "How can you cool a cup of water faster?"). PiQA probes the model's intuitive grasp of how the physical world operates, a domain of commonsense distinct from academic knowledge, making it an excellent test of generalization.

- **OpenBookQA** (Mihaylov et al., 2018): This benchmark simulates an "open-book" exam, requiring the model to answer questions using a given set of elementary science facts. Success demands not only reading comprehension but, more importantly, the ability to reason over and combine these facts to answer questions whose solutions are not explicitly stated. It critically evaluates the model's capacity for multi-step reasoning and knowledge application within a constrained context.

- **BoolQ (Boolean Questions)** (Clark et al., 2019): A dataset of naturally occurring yes/no questions, sourced from real user search queries. The challenge lies in the fact that the relationship between the question and the provided evidence passage is often implicit, requiring sophisticated syntactic and semantic analysis to arrive at a correct Boolean judgment. BoolQ effectively measures the model's fine-grained comprehension of natural, conversational language.

- **MathQA** (Amini et al., 2019): A mathematical reasoning benchmark featuring multi-step word problems. The task requires models to parse natural language descriptions, formulate a correct sequence of operations, and execute them to find a solution. Covering a diverse range of mathematical reasoning categories, MathQA is a crucial benchmark for evaluating a model's symbolic reasoning and logical chain-of-thought capabilities, representing a significant test of higher-order cognitive skills.

By systematically evaluating our method across this dual-category, nine-benchmark matrix, we demonstrate that our approach not only enhances performance in core competency areas (as shown by MMLU and ARC-C) but also significantly improves the transfer of these abilities to novel contexts (as evidenced by PiQA and MathQA). This comprehensive improvement across both in-domain and out-of-domain tasks provides strong evidence for the effectiveness and generalizability of our method.

To further investigate the impact of model scale on data utilization, we present a supplementary analysis in Figures 5 to 11. Our key finding is that models of different scales (1B and 7B) exhibit significantly different learning responses and form distinct preferences, even when trained on the exact same data. This phenomenon reveals a complex interplay between data utility and model scale. It provides a solid theoretical foundation for

Table 3: Ablation study of REGMIX and TiKMiX on 1B and 7B models.

	1B Model		7B Model	
Benchmark	**REGMIX**	**TiKMiX-D**	**REGMIX**	**TiKMiX-D**
In-Domain Benchmarks				
MMLU (Hendrycks et al., 2020)	31.5	32.2	40.7	41.5
HellaSwag (Zellers et al., 2019)	56.0	57.4	76.6	76.4
ARC Easy (Clark et al., 2018)	66.2	69.3	78.5	78.4
ARC Challenge (Clark et al., 2018)	32.2	37.0	49.4	50.2
TriviaQA (Joshi et al., 2017)	15.8	17.7	46.4	45.3
Out-of-Domain Benchmarks				
PiQA (Bisk et al., 2020)	73.3	74.1	79.1	79.2
OpenBookQA (Mihaylov et al., 2018)	37.0	37.4	43.2	45.4
MathQA (Amini et al., 2019)	23.2	23.5	28.8	29.9
Average Perf.	43.9	45.5	55.3	56.0

understanding and optimizing the data mixture for models of varying sizes.

C Experiments on models of different sizes

Considering computational overhead, for the 7B model, we adopted an experimental design similar to REGMIX(Liu et al., 2024), training with 500B tokens in the first stage and 200B tokens in the second stage. Table 3 presents the experimental results of our method on models of different scales. It can be observed that our proposed method significantly outperforms the current state-of-the-art approach, REGMIX, on both in-domain and out-of-domain benchmarks. The performance on the 7B model effectively demonstrates the scalability of our approach. Furthermore, we note that unlike the 1B model, the 7B model's performance on the benchmarks consistently improves throughout the training process. This suggests that the advantage of TiKMiX could be even more pronounced with additional training data.

D Observation of data mixing with Group Influence

To conduct a rigorous analysis of inter-domain interactions during mixed training, we designed an experiment to test the principle of influence additivity. Our hypothesis was that the influence of a mixed dataset on a validation set could be accurately predicted by a weighted sum of the influences from its individual constituent domains. To verify this, we first established a baseline mixing recipe using our TiKMiX-D method. We then systematically explored the local space around this recipe by generating 256 perturbed configurations, created by applying a random scaling factor between 0.5 and 2.0 to each domain's original proportion. After filtering out two sampling outliers, we

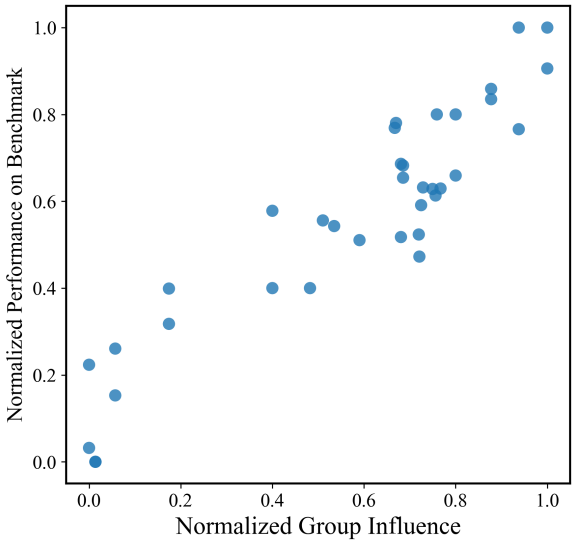

Figure 4: Analysis of the Group Influence and actual performance on the benchmark.

proceeded with 254 unique data mixture configurations. For each of these 254 points, we sampled a corresponding 0.1B token dataset and measured its direct influence. We then compared this empirical influence value against a predicted influence, which was calculated by summing the pre-computed influences of each individual domain, weighted by their respective proportions in the mixture. As depicted in Fig 13, this comparison revealed a strong linear correlation. Specifically, the Pearson correlation coefficients on the ARC(Clark et al., 2018), Hellaswag(Zellers et al., 2019), and TriviaQA(Joshi et al., 2017) benchmarks reached 0.845, 0.848, and 0.931, respectively, all of which are statistically highly significant ($p < 0.0001$). This result provides compelling evidence that the outcome of data mixing is highly predictable and can be modeled as a linear combination of inter-domain influences. Consequently, this finding offers a solid empirical justification for the theoretical soundness of our proposed two-stage optimization framework, encompassing both TiKMiX-D and TiKMiX-M.

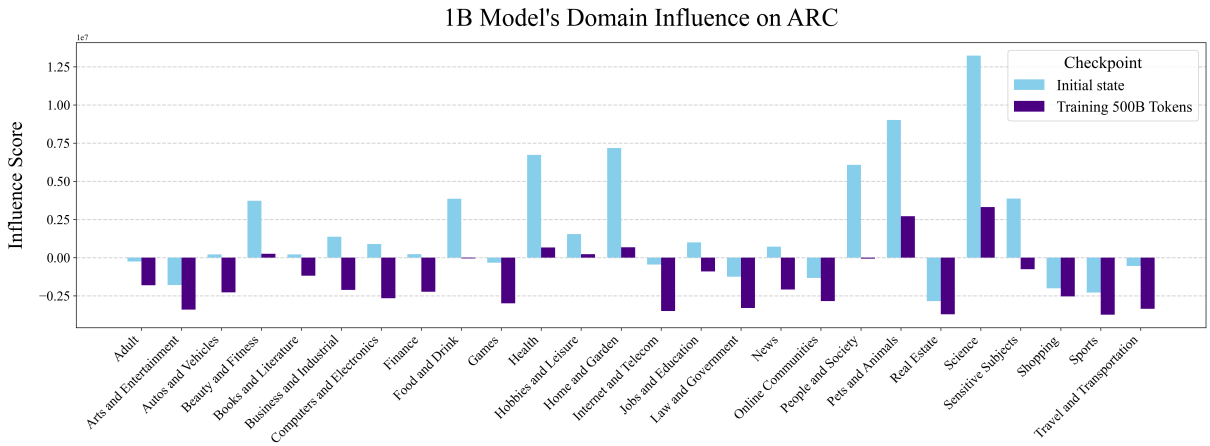

Figure 5: The impact of domains on a 1B model's performance on the ARC benchmark as training progresses.

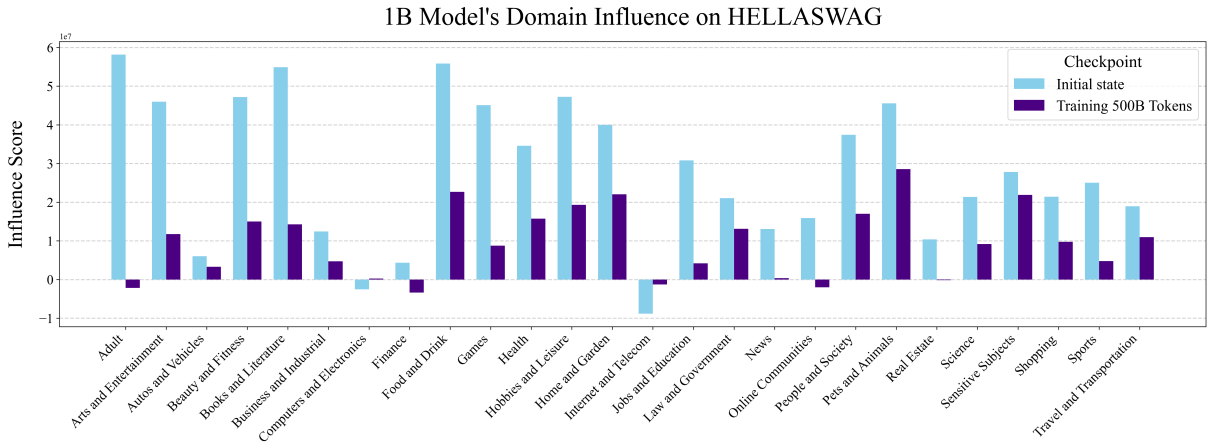

Figure 6: The impact of domains on a 1B model's performance on the HELLASWAG benchmark as training progresses.

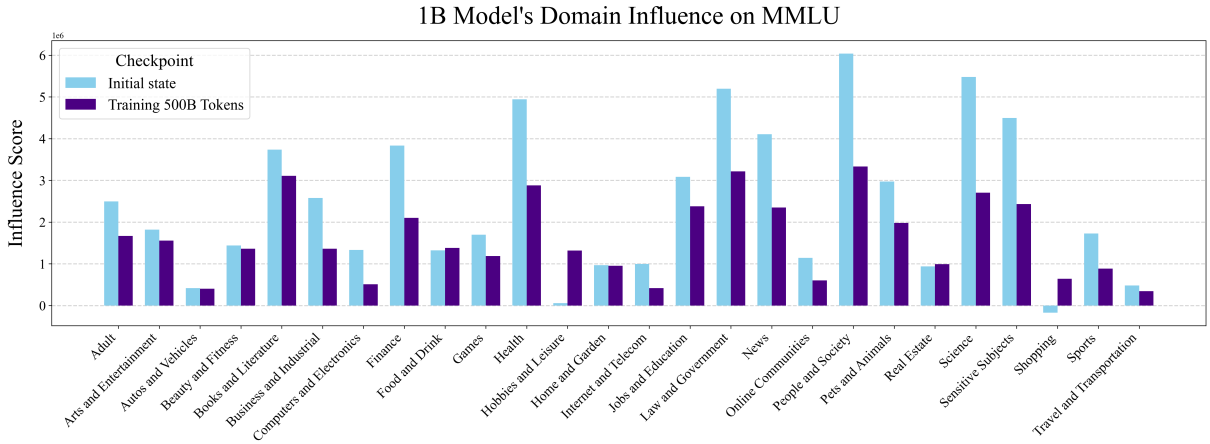

Figure 7: The impact of domains on a 1B model's performance on the MMLU benchmark as training progresses.

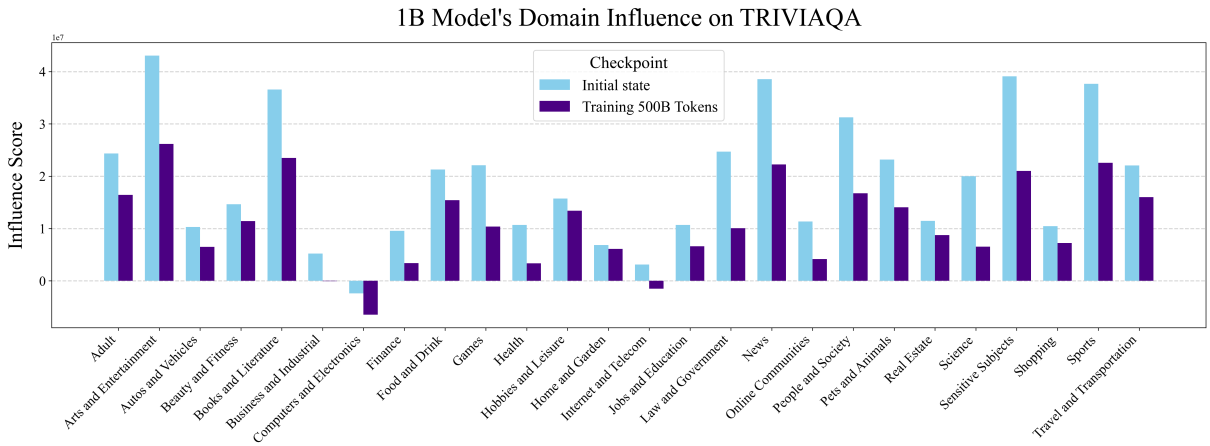

Figure 8: The impact of domains on a 1B model's performance on the TRIVIAQA benchmark as training progresses.

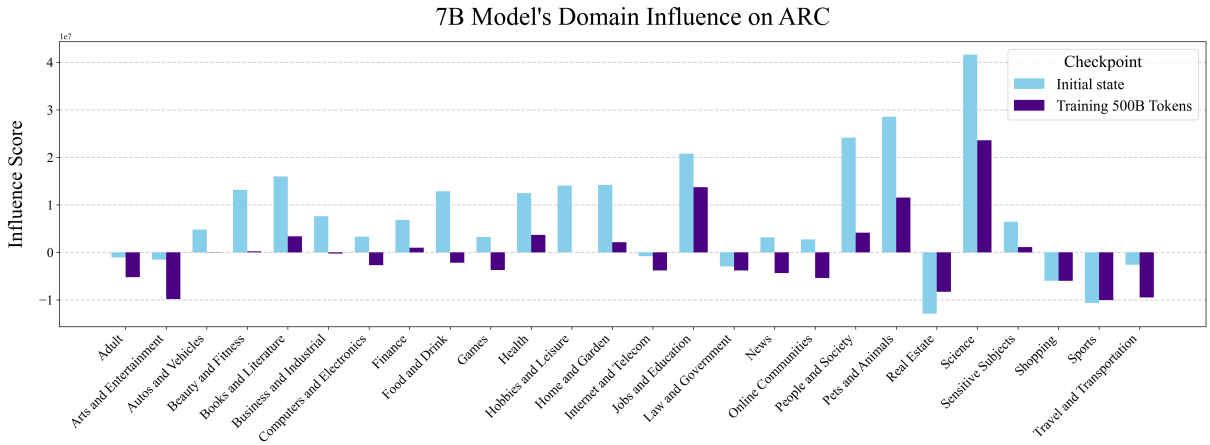

Figure 9: The impact of domains on a 7B model's performance on the ARC benchmark as training progresses.

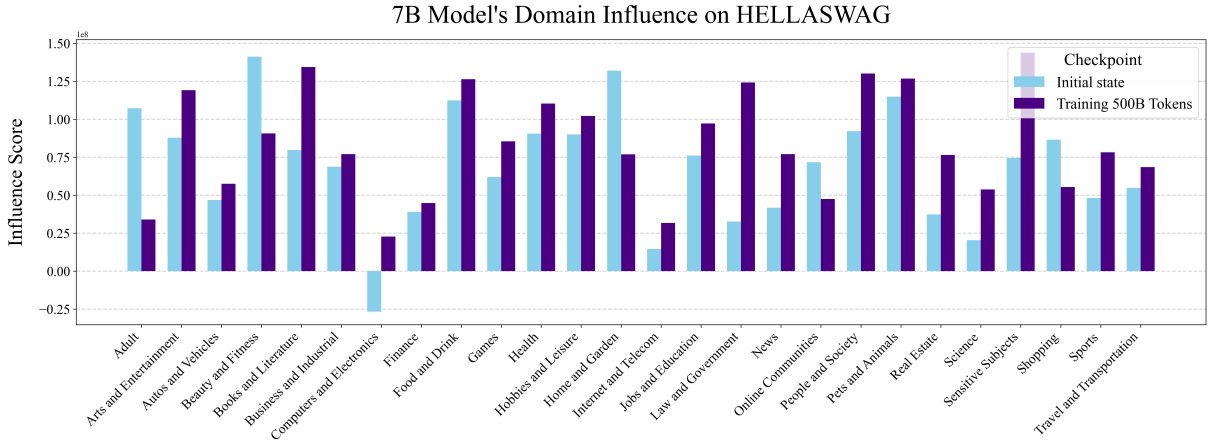

Figure 10: The impact of domains on a 7B model's performance on the HELLASWAG benchmark as training progresses.

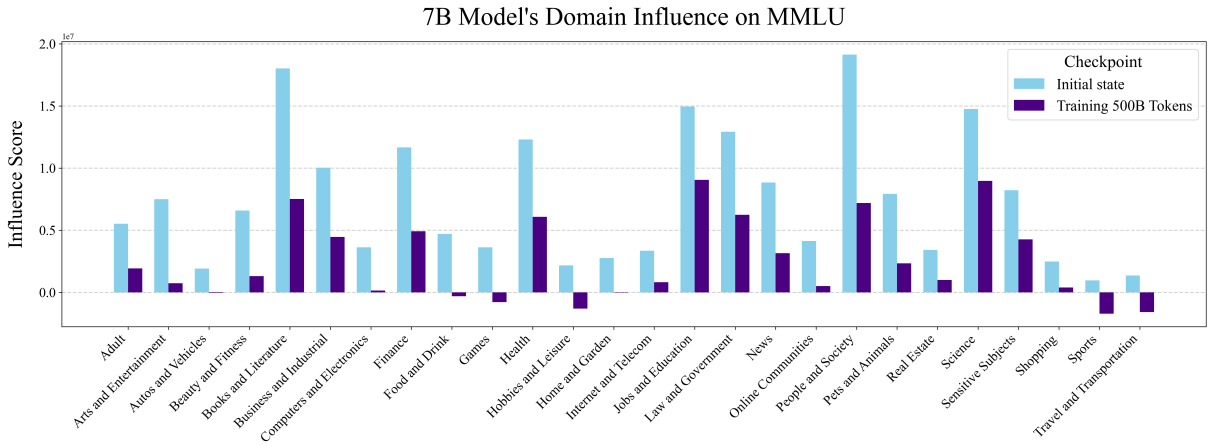

Figure 11: The impact of domains on a 7B model's performance on the MMLU benchmark as training progresses.

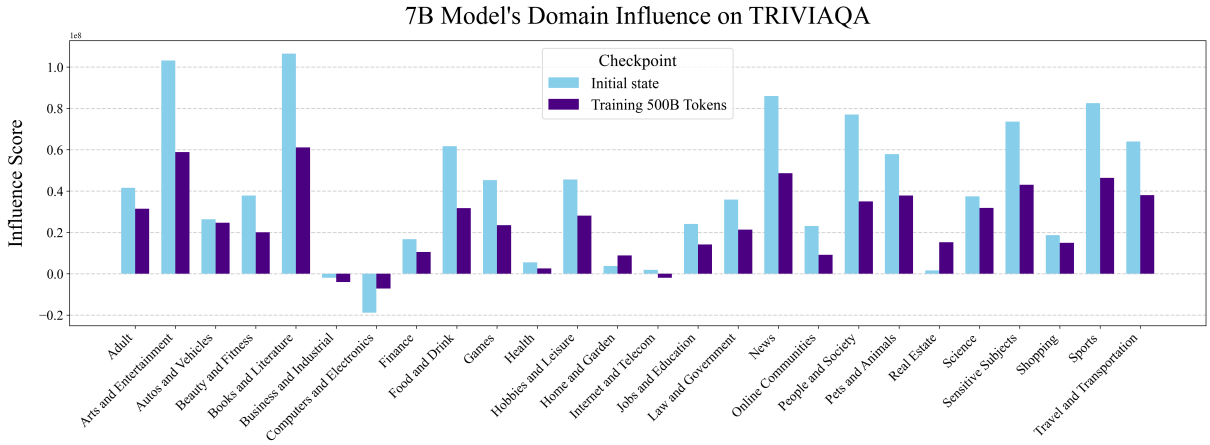

Figure 12: The impact of domains on a 7B model's performance on the TRIVIAQA benchmark as training progresses.

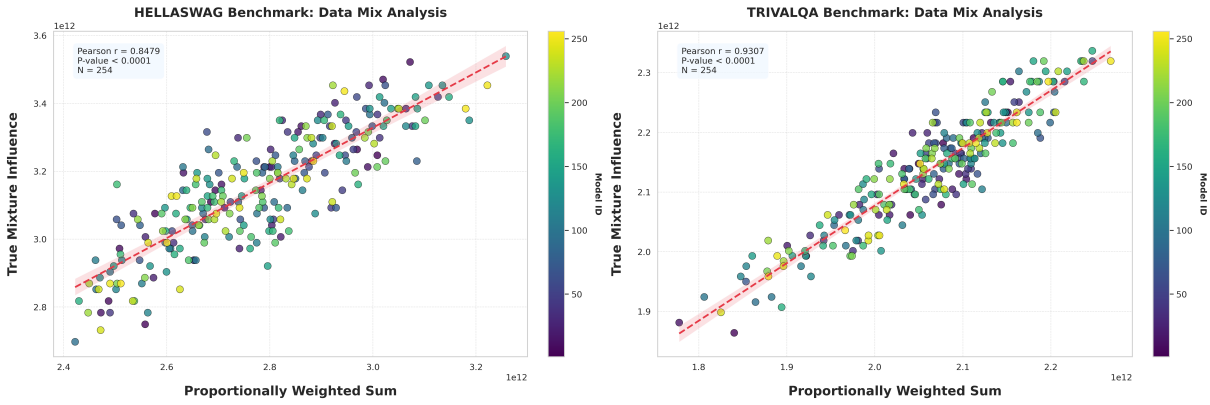

Figure 13: A Group Influence-based Analysis of Data Mixing Effects on Various Benchmarks.