DYNAMIXSFT: Dynamic Mixture Optimization of Instruction Tuning Collections

Haebin Shin[1,2*] **Lei Ji**[1] **Xiao Liu**[1] **Zhiwei Yu**[1] **Qi Chen**[1] **Yeyun Gong**[1]

[1]Microsoft Research [2]KAIST AI

haebin.shin@kaist.ac.kr {leiji, xiaoliu2, zhiwyu, cheqi, yegong}@microsoft.com

arXiv:2508.12116v1 [cs.LG] 16 Aug 2025

Abstract

As numerous instruction-tuning datasets continue to emerge during the post-training stage, dynamically balancing and optimizing their mixtures has become a critical challenge. To address this, we propose DYNAMIXSFT, a dynamic and automated method for instruction-tuning dataset mixture optimization. We formulate the problem as a multi-armed bandit setup and introduce a *Prior-scaled Boltzmann Exploration* that softly anchors the updated sampling distribution to the original dataset proportions, thereby preserving the inherent diversity and coverage of the collection. Sampling probabilities are updated using a lightweight *1-Step Look-ahead Reward*, reflecting how much the dataset contributes to improving the model's performance at its current state. When applied to the TÜLU-v2-mixture collection comprising 16 instruction-tuning datasets, DYNAMIXSFT achieves up to a 2.2% performance improvement across 10 benchmarks. Furthermore, we provide a comprehensive analysis and visualizations to offer deeper insights into the adaptive dynamics of our method.

Introduction

The mixture of diverse datasets has emerged as a pivotal factor in the post-training stage of large language models (LLMs), significantly enhancing their ability to generalize across a broad spectrum of domains—including instruction following, reasoning, mathematics, coding, and knowledge-intensive tasks (Wei et al. 2022; Longpre et al. 2023). A well-composed dataset mixture allows LLMs to acquire a broad set of capabilities from heterogeneous sources while preserving robustness and generality. This has led to the development of large-scale mixtures (Wang et al. 2023; Ivison et al. 2023; Longpre et al. 2023; Lambert et al. 2025), which offer standardized recipes for instruction tuning with a diverse set of supervision signals.

However, most curated mixtures are statically defined through manual heuristics or fixed rules (Wang et al. 2023; Ivison et al. 2023; Lambert et al. 2025), requiring extensive human effort and domain expertise to design effective sampling strategies. To address this limitation, several recent studies have explored dynamic mixture approaches, which often rely on proxy models (Xie et al. 2023; Fan, Pagliardini, and Jaggi 2024a; Liu et al. 2025), scorer networks (Wu et al. 2024; Wang et al. 2025) or performance on a well-defined validation set (Wang, Tsvetkov, and Neubig 2020; Wu et al. 2021) to guide dataset mixture. This highlights the need for a lightweight and self-evolving method that can dynamically adapt the dataset mixture during training without requiring additional overhead.

To address this challenge, we propose DYNAMIXSFT, a dynamic framework that formulates the allocation of training samples across k heterogeneous data sources as a multi-armed bandit (MAB) problem, where each dataset is treated as an arm. To support adaptive yet balanced mixture optimization, we further introduce a **Prior-scaled Boltzmann Exploration**, a strategy that softly anchors the updated sampling distribution to the original dataset proportions. This encourages adaptive sampling while preserving the inherent diversity and coverage of the dataset collection. Additionally, we introduce a lightweight **1-Step Look-ahead Reward** to estimate each dataset's contribution to current training dynamics, enabling the model to prioritize more beneficial data sources based on its own learning progress.

We apply DYNAMIXSFT to the TÜLU-v2-mixture collection (Ivison et al. 2023), which comprises 16 instruction-tuning datasets, and evaluate the model on 10 benchmarks covering knowledge, reasoning, mathematics, coding, and instruction following. Experiments conducted using LLaMA3.2 1B (Meta 2024) and Qwen2.5 3B (Qwen et al. 2025) demonstrate that DYNAMIXSFT achieves up to a +2.2% relative gain in average performance compared to the original static mixtures.

To better understand the benefits of DYNAMIXSFT, we conduct a comprehensive analysis, including visualizations of the evolving mixture proportions during training. This analysis provides practical insights into how the dataset mixture adapts over time in response to the model's training dynamics, and highlights the role of various factors from the perspective of the bandit algorithm. Ultimately, our approach enables LLMs to optimize their own dataset mixtures during post-training without the need for hand-crafted tuning, making it feasible to leverage large-scale dataset collections in a more principled and automated manner.

Our contributions are summarized in three folds:

- We propose DYNAMIXSFT, a dynamic dataset mixture optimization method for LLM post-training. DYNAMIXSFT formulates the dataset mixture problem as

[*]Work done during internship at Microsoft Research

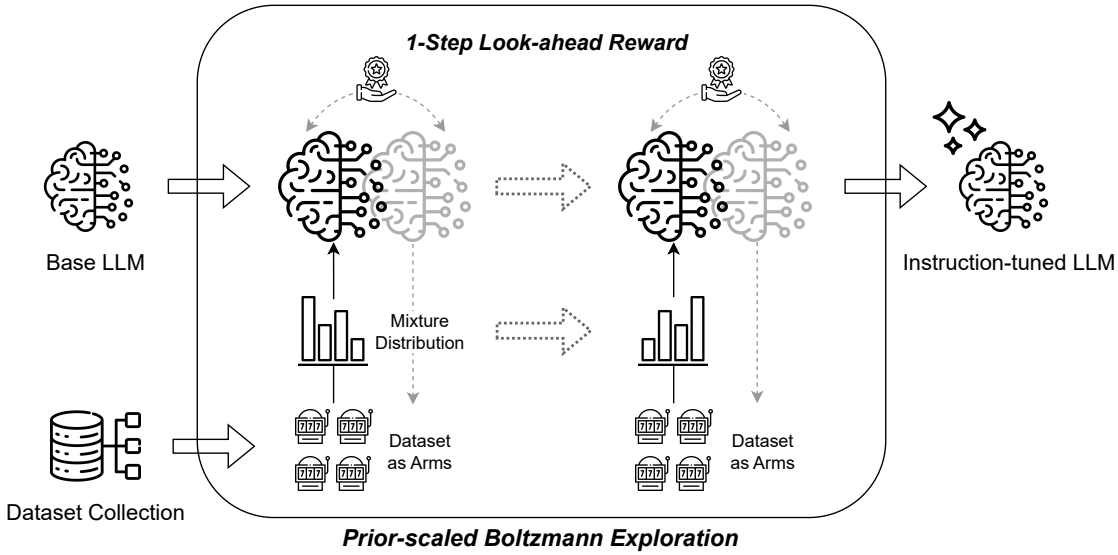


Figure 1: Overview of DYNAMIXSFT. Given a large collection of instruction-tuning datasets, DYNAMIXSFT treats each dataset as an arm in a *Multi-Armed Bandit* setup. The sampling policy dynamically evolves through *Prior-scaled Boltzmann Exploration*, periodically updated by a lightweight *1-Step Look-ahead Reward* that reflects the model's current training dynamics, enabling adaptive mixture optimization.

a multi-armed bandit setup and introduces a *Prior-scaled Boltzmann Exploration* policy that softly anchors the updated sampling distribution to the original dataset proportions. To update sampling probabilities, we further introduce a *1-Step Look-ahead Reward*, which reflects the model's current training dynamics and enables adaptive sampling over time.

- We apply DYNAMIXSFT to a large-scale collection of 16 instruction-tuning datasets and evaluate performance across 10 benchmarks spanning knowledge, reasoning, mathematics, coding, and general instruction-following. Our method outperforms static mixtures, achieving up to a +2.2% gain in average performance on 1B- and 3B-scaled models.
- We present detailed analysis and visualizations of how mixture dynamics change throughout training. This provides insight into the model's evolving needs and demonstrates how the bandit-based adaptation contributes to effective and interpretable mixture scheduling.

Related Works

Dataset Mixture for LLMs

Recent studies have shown that the composition of instruction-tuning data mixtures is a crucial factor in the effectiveness and generalization of large language models during post-training. However, most current practices rely on static or manually curated sampling strategies. TÜLU-1 (Wang et al. 2023) introduces initial mixtures for open-ended alignment, while TÜLU-2 (Ivison et al. 2023) expands this by incorporating a broader set of diverse, high-quality datasets with unified formatting and filtering strategies. TÜLU-3 (Lambert et al. 2025) further refines this mixture with additional sources and improved normalization. In this work, we focus on optimizing the TÜLU-*v2-mixture* collection, which consists of 16 instruction-tuning datasets covering a wide range of capabilities.

Mixture Optimization

Several studies have explored dataset mixture optimization in the pretraining domain. DoReMi (Xie et al. 2023) optimizes pretraining mixtures by computing per-dataset gradients using a small proxy model to minimize loss on a held-out validation set. DoGE (Fan, Pagliardini, and Jaggi 2024b) proposes a curriculum-style mixture guided by a proxy model's generalization gap. RegMix (Liu et al. 2025) formulates mixture optimization as a regression task, where a proxy model is trained to estimate the mixture that minimizes targeted evaluation loss. Although such proxy-based methods are feasible in pretraining where large-scale training budgets can justify the cost of auxiliary models, they become prohibitively expensive in post-training scenarios. Albalak et al. (2023) formulates pre-training dataset mixture as a bandit problem, dynamically adjusting sampling over a small number of curated domains, though its scope remains limited to the vanilla pre-training phase. In this work, we aim to optimize the dataset mixture without relying on an additional proxy model, making our approach both lightweight and dynamically adaptable for instruction tuning.

In the post-training setting, several methods have explored how to better utilize heterogeneous instruction-tuning data. Wang et al. (2024) provides an empirical study of how three manually chosen dataset types interact, highlighting growing interest in mixture strategies. MultiDDS (Wang,

Tsvetkov, and Neubig 2020) and MultiUAT (Wu et al. 2021) dynamically re-weight a few clearly separated tasks using gradient similarity or uncertainty from a trusted validation set. Mixture-of-Skills (Wu et al. 2024) and HBO (Wang et al. 2025) introduce additional actor networks trained via reinforcement learning or bilevel optimization, enabling scalable mixture optimization but at the cost of increased overhead. Zhu et al. (2025) utilizes token-routing statistics for dataset reweighting but is tied to Mixture-of-Experts (MoE) architectures, while SFTMix (Xiao et al. 2025) focuses on confidence-based mixup at the regularization level, rather than modifying the sampling policy. Chen et al. (2025) utilizes a non-stationary bandit in the reasoning domain to select examples of appropriate difficulty, relying on well-defined rewards such as advantage estimates.

In contrast, DYNAMIXSFT targets a more challenging post-training setting over large-scale, uncurated instruction-tuning collections, where dataset boundaries are ambiguous, and predefined reward signals are unreliable. Moreover, DYNAMIXSFT treats boundary-less instruction-tuning collections—commonly found in real-world post-training—as a multi-armed bandit problem, enabling efficient data allocation without introducing additional parameters or architectural constraints.

Dynamic Mixture Optimization

We introduce DYNAMIXSFT, a method that dynamically adjusts the mixture of datasets as described in Algorithm 1.

Mixture as a Multi-Armed Bandit Problem

Considering the nature of large-scaled instruction-tuning dataset collections (Wang et al. 2023; Ivison et al. 2023; Lambert et al. 2025), we formulate the problem of sampling an optimal batch from K candidate datasets as a multi-armed bandit (MAB) problem. To quickly adapt to the non-stationary reward distribution that naturally arises during fine-tuning, we adopt a Boltzmann exploration strategy (Sutton, Barto et al. 1998; Cesa-Bianchi et al. 2017; Gupta et al. 2019). A typical Boltzmann exploration (Sutton, Barto et al. 1998) computes sampling probabilities using the following $\mathrm{softmax}(\beta \cdot Q)$:

$$\frac{\exp(\beta \cdot Q_k)}{\sum_j \exp(\beta \cdot Q_j)}, \tag{1}$$

which treats all reward values Q as equally interpretable across datasets.

However, such equally interpretable rewards can in fact skew the sampling distribution toward certain datasets and fail to account for the inherent characteristics of each dataset. In the context of language model post-training, both *diversity* and *coverage* of the training instances are critical. Notably, we assume that the original dataset proportions—often based on dataset size—implicitly reflect each dataset's inherent characteristics and its intended contribution to domain coverage. For example, LIMA (Zhou et al. 2023a) is small but well-curated, whereas FLAN (Longpre et al. 2023) is relatively large to cover a broader range of domains.

Algorithm 1: DYNAMIXSFT

Require: SFT Dataset Collection $\mathcal{C} = \{\mathcal{D}_1, \ldots, \mathcal{D}_K\}$; EMA values $Q \in \mathbb{R}^K$; Prior probabilities $p^{(0)} \in \Delta_K$; Sharpness factor β; Uniformity factor γ; EMA smoothing factor α; Total steps T; Update interval T_{update}; Batch size B

Initialize $Q_k \leftarrow 0$ for all $k \in \{1, \ldots, K\}$
for step $t = 1$ to T **do**
  $B_t \leftarrow \emptyset$
  while $|B_t| < B$ **do**
    Compute Multi-Armed Bandit probabilities:

$$p_k \leftarrow (1-\gamma) \cdot \frac{\exp(\beta \cdot Q_k) \cdot p_k^{(0)}}{\sum_j \exp(\beta \cdot Q_j) \cdot p_j^{(0)}} + \gamma/K$$

    Sample dataset $\mathcal{D} \sim p$
    Sample example $x \sim \mathcal{D}$ uniformly
    Add to batch: $B_t \leftarrow B_t \cup \{x\}$
  end while
  Train model with batch B_t
  if $t \mod T_{\text{update}} = 0$ **then**
    for each $\mathcal{D}_k \in \mathcal{C}$ **do**
      Sample next batch $B_{\mathcal{D}_k}$ from $\mathcal{D}_k$
      Compute reward:

$$r_k \leftarrow \frac{1}{|B_{\mathcal{D}_k}|} \sum_{x \in B_{\mathcal{D}_k}} \frac{\mathcal{L}_{\text{pre}}(x) - \mathcal{L}_{\text{post}}(x)}{\mathcal{L}_{\text{pre}}(x) + \epsilon}$$

      Update EMA: $Q_k \leftarrow \alpha \cdot Q_k + (1-\alpha) \cdot r_k$
    end for
  end if
end for

To account for this, we introduce a **Prior-scaled Boltzmann Exploration** that explicitly incorporates the underlying dataset distribution $p^{(0)}$ as a prior, extending Equation 1 as follows:

$$\frac{\exp(\beta \cdot Q_k) \cdot p_k^{(0)}}{\sum_j \exp(\beta \cdot Q_j) \cdot p_j^{(0)}}. \tag{2}$$

Our prior-scaled policy softly anchors the learned sampling distribution to the original proportions, allowing for adaptive reward-driven updates while preserving the broad coverage and diversity encoded in the original mixture.

Nonetheless, a never-sampling problem may still arise due to the non-stationary nature of reward dynamics. To mitigate this, we apply a **Minimum Floor Probability** (γ/K), ensuring that each dataset retains a non-zero probability of being sampled. This guarantees uniform exploration and facilitates adaptation to sudden shifts in reward dynamics.

As a result, we formulate the final bandit sampling probability as shown in Equation (3). Here, the sharpness parameter β controls the exploitation strength of the Prior-Scaled Boltzmann term, while the uniformity factor γ provides the exploration guarantee through the γ/K term, striking an effective balance between exploitation and exploration.

$$(1-\gamma)\cdot\frac{\exp(\beta\cdot Q_k)\cdot p_k^{(0)}}{\sum_j \exp(\beta\cdot Q_j)\cdot p_j^{(0)}}+\gamma/K \tag{3}$$

1-Step Look-ahead Reward

The goal of updating the bandit probabilities is, intuitively, to assign higher sampling probability to datasets that are expected to contribute more effectively to the model's training progress at its current state. Concretely, at each update interval T_{update}, we estimate this contribution by performing a **1-Step Look-ahead Δ-Loss**: for each dataset, we temporarily update the model by one gradient step on a mini-batch, and then measure how much the model loss decreases as a result. This provides an immediate signal of each dataset's utility without requiring a separate validation set or permanent parameter updates. The reward for dataset D_i is computed as follows:

$$r_{\mathcal{D}_i} \leftarrow \frac{1}{|B_{D_i}|}\sum_{x\in B_{D_i}}\frac{\mathcal{L}_{\text{pre}}(x)-\mathcal{L}_{\text{post}}(x)}{\mathcal{L}_{\text{pre}}(x)+\epsilon} \tag{4}$$

Here, $\mathcal{L}_{\text{pre}}(x)$ and $\mathcal{L}_{\text{post}}(x)$ denote the loss before and after the virtual one-step update, respectively, and ϵ is a small constant for numerical stability. This lightweight look-ahead reward encourages the bandit scheduler to dynamically favor data sources that yield higher immediate loss reduction per unit cost.

To mitigate noisy fluctuations and adapt to the inherently non-stationary nature of training rewards, we maintain an exponential moving average of the reward for each dataset:

$$Q_{\mathcal{D}_i} \leftarrow \alpha\cdot Q_{\mathcal{D}_i}+(1-\alpha)\cdot r_{\mathcal{D}_i}, \tag{5}$$

where $\alpha\in(0,1)$ is the smoothing factor.

This smoothing helps to prevent overreacting to sudden batch-level spikes and ensures that the bandit scheduler remains robust to short-term variance while still tracking long-term trends in dataset utility.

Experimental Setup

Dataset Collection

We utilize TÜLU-2 (Ivison et al. 2023) mixture collection [1], which consists of about 300K examples from 16 datasets: FLAN (Longpre et al. 2023), FLAN-CoT (Longpre et al. 2023), Open Assistant 1 (Köpf et al. 2023), ShareGPT, GPT4-Alpaca (Peng et al. 2023), Code Alpaca (Chaudhary 2023), LIMA (Zhou et al. 2023a), WizardLM (Xu et al. 2024), Open-Orca (Lian et al. 2023), Hardcoded (Ivison et al. 2023), Evidence Inference (Lehman et al. 2019), Qasper (Dasigi et al. 2021), SciERC (Named entity recognition, Relation extraction) (Luan et al. 2018), SciFact (Wadden et al. 2020), SciTLDR (Cachola et al. 2020). Details of each dataset are described in the Appendix.

[1] https://huggingface.co/datasets/allenai/tulu-v2-sft-mixture

Evaluation

Following the setup from Lambert et al. (2025), we assess the model on 10 benchmarks covering diverse domain; **Knowledge**: MMLU (Hendrycks et al. 2021), TruthfulQA (Lin, Hilton, and Evans 2022), PopQA (Mallen et al. 2023), **Reasoning**: BigBench-Hard (Suzgun et al. 2022), DROP (Dua et al. 2019), **Coding**: HumanEval (Chen et al. 2021), HumanEval+ (Liu et al. 2023), **Mathmetical**: GSM8K (Cobbe et al. 2021), MATH (Hendrycks et al. 2021), **Instruction Following**: IFEval (Zhou et al. 2023b). Further details of the evaluation setup are provided in the Appendix.

Baselines

We compare our method with two baseline groups, **Heuristic Methods**: *Full Coverage* (Proportional sampling) and *Uniform Sampling*; and **Dynamic Methods**: *MultiDDS* (Wang, Tsvetkov, and Neubig 2020) and *MultiUAT* (Wu et al. 2021), adaptively adjust dataset mixtures based on gradient cosine similarity and model uncertainty.

Implementation Details

We conduct experiments with LLaMA3.2 1B (Meta 2024) and Qwen2.5 3B (Qwen et al. 2025) models. Following the TÜLU training setup (Ivison et al. 2023; Lambert et al. 2025), we train the model for 2 epochs, using a batch size of 128 for both training and reward updates. All training is conducted on 8 $\times$ A100 40GB GPUs. In accordance with the TÜLU-2 collection's configuration, the number of arms K is set to 16. Further hyperparameter details are described in the Appendix.

Results

Overall Comparison

As shown in Table 1, DYNAMIXSFT consistently outperforms other baselines across both 1B and 3B models. Compared to training with naive proportional sampling (*Full Coverage*), DYNAMIXSFT achieves a relative improvement of +2.2% on the 1B model and +1.5% on the 3B model in terms of the average score across 10 benchmarks. Furthermore, DYNAMIXSFT demonstrates superior performance over existing dynamic dataset mixture methods such as *MultiDDS* (Wang, Tsvetkov, and Neubig 2020) and *MultiUAT* (Wu et al. 2021), confirming the effectiveness of our bandit-based approach using *1-Step Look-ahead Δ-Loss*.

Impact of Instance Coverage

Among the baselines, the *Full Coverage* (proportional sampling) approach shows relatively strong performance for both 1B and 3B models. In contrast, *Uniform Sampling* consistently underperforms. As illustrated in Figure 2(b), *Uniform Sampling* disregards the inherent dataset size differences, leading to over-sampling or under-sampling of datasets. This imbalance often distorts the natural characteristics of the original dataset collection, resulting in suboptimal model performance.

Method	General Knowledge			Reasoning		Coding		Mathematical		IFEval	AVG
	MMLU	TQA	PopQA	BBH	DROP	CHE	CHE+	GSM8K	MATH		
LLAMA3.2 1B											
Base	33.93	37.73	18.81	31.17	28.58	31.94	29.70	6.97	1.85	10.53	23.12
Full Coverage	32.03	41.39	16.62	33.37	28.12	37.02	35.27	15.79	2.97	25.50	26.81
Uniform Sampling	30.22	40.94	15.88	31.83	28.01	34.89	29.72	11.59	2.86	23.65	24.96
MultiDDS	31.55	41.49	16.52	32.95	28.20	38.01	35.71	15.54	2.97	25.32	26.83
MultiUAT	32.14	41.56	16.52	33.28	28.14	37.89	34.63	14.70	2.91	25.69	26.75
DYNAMIXSFT (W/O PRIOR)	30.31	40.95	16.82	31.56	28.28	37.42	32.19	12.88	2.63	26.06	25.91
DYNAMIXSFT	31.97	41.59	16.05	33.48	28.78	41.23	34.96	15.69	3.05	27.17	**27.40**
QWEN2.5 3B											
Base	64.40	48.95	18.89	54.78	46.82	85.13	80.85	63.76	25.71	24.02	51.33
Full Coverage	57.16	49.65	18.15	54.46	42.46	87.34	80.57	71.41	21.87	39.00	52.21
Uniform Sampling	56.25	44.95	17.50	55.26	45.65	86.21	80.33	67.47	9.26	40.29	50.32
MultiDDS	55.14	48.83	18.28	54.50	43.45	86.39	81.83	70.96	19.71	39.00	51.81
MultiUAT	56.82	49.22	17.92	54.87	43.12	85.46	80.66	71.11	22.17	40.29	52.16
DYNAMIXSFT (W/O PRIOR)	60.21	46.15	18.72	54.44	45.61	86.91	81.31	69.29	14.77	38.63	51.60
DYNAMIXSFT	58.30	49.60	18.11	55.49	45.90	88.17	81.32	71.79	20.13	40.85	**52.97**

Table 1: Performance comparison across 10 evaluation suites, including knowledge, reasoning, coding, mathematical, and instruction-following categories. **AVG** is the unweighted mean of all 10 scores. The best results are highlighted in **bold**.

Effect of Prior-scaled Boltzmann Exploration

The impact of coverage is further supported by the ablation study of DYNAMIXSFT (W/O PRIOR). When the prior distribution $p^{(0)}$ in Equation (2) is omitted, DYNAMIXSFT starts with a uniform distribution, which mirrors the same drawback observed with *Uniform Sampling*. Despite the mixture proportions being updated by rewards, the sampling distribution remains close to its initial uniform distribution throughout training (see Appendix for visualization). This highlights the importance of the initial distribution in DYNAMIXSFT, as a uniform initialization can bias the learning process and negatively affect the final mixture quality and model performance. Therefore, especially in large-scale post-training scenarios with diverse instruction-tuning datasets, it is critical to reflect dataset characteristics in the initial mixture to enable effective dynamic mixture learning.

Visualization of the Effect of Dynamic Mixture

Figure 2 provides an intuitive visualization of how different sampling strategies shape the dataset composition during training. Compared to baselines with static ratios such as *Full Coverage* and *Uniform Sampling*, DYNAMIXSFT dynamically adjusts the mixture weights over time. This dynamic behavior is driven by our *1-Step Look-ahead Reward*, which adapts to the current gradient signal and naturally follows the learning rate schedule. As a result, the sampling proportions become more responsive until mid-training steps, when the learning rate is higher. This adaptivity ultimately produces a more balanced final dataset coverage, as shown in Figure 2(c).

Analysis

Exploitation vs. Exploration

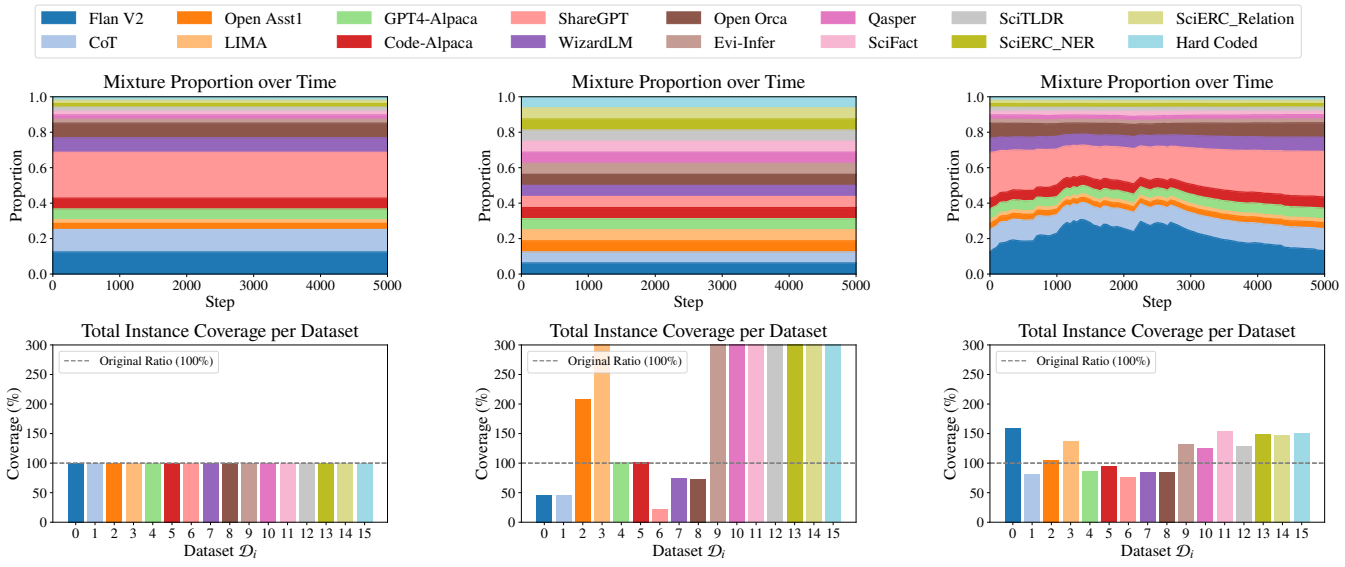
As described in Equation (3), we control the exploitation strength using the sharpness factor β and the exploration strength using the uniformity factor γ. As shown in Figure 5(a), we investigate the effect of varying these two parameters on the 1B model's performance.

Under strong exploration settings ($\gamma = 0.3, 0.25$; blue lines), performance peaks at lower exploitation sharpness values ($\beta = 4$). In contrast, under weaker exploration settings ($\gamma = 0.2, 0.15$; orange lines), higher sharpness values ($\beta = 7, 8$) lead to better performance. Notably, extreme exploration values ($\gamma = 0.3, 0.15$; solid lines) lead to greater performance variance across different β. This trends is further observed in the heatmap of Figure 5(b), where the color shift from red to blue indicates significant score variation. These results highlight the importance of carefully balancing exploitation sharpness and exploration uniformity to achieve optimal mixture performance.

Smoothing EMA Reward Strength

As described in Equation (5), we control the strength of the exponential moving average reward signal using the smoothing factor α, which determines how strongly recent rewards influence the sampling policy. We investigate the effect of different smoothing factor α on performance and mixture dynamics.

In Figure 6(a), moderate smoothing values ($\alpha = 0.8, 0.5$; solid lines) lead to the highest performance when the sharpness factor $\beta = 4$, indicating that a balance between responsiveness and stability leads to effective mixture up-



(a) *Full Coverage* (b) *Uniform Sampling* (c) DYNAMIXSFT

Figure 2: Comparison of "Mixture Proportions over Time" and "Total Instance Coverage per Dataset" under static and dynamic sampling strategies. In the *Full Coverage* setting, mixture proportions are skewed based on dataset sizes (*top left*), but all instances from each dataset are guaranteed to be used during training (*bottom left*). In *Uniform Sampling*, each dataset is sampled equally (*top center)*, which leads to over- and under-sampling of instances depending on dataset size (*bottom center*). In contrast, DYNAMIXSFT adaptively adjusts mixture proportions over time (*top right*), resulting in well-balanced instance coverage across datasets (*bottom right*). Best viewed in color.

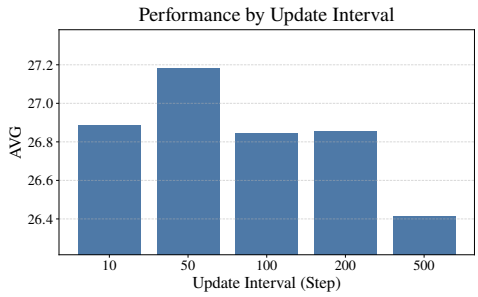


Figure 3: Comparison of Performance by Varying Reward Update Interval.

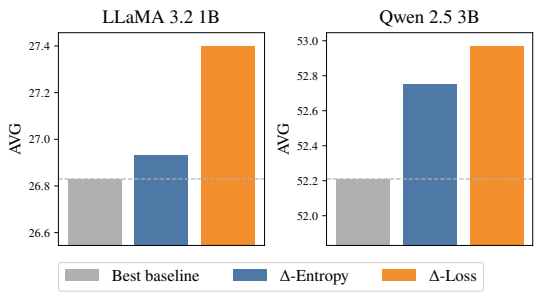


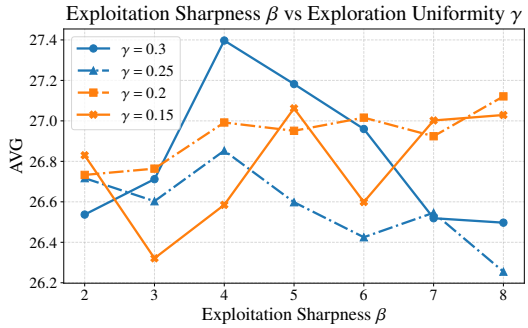
Figure 4: Comparison of Performance Using Δ-Loss vs. Δ-Entropy Rewards.

dates. Strong smoothing ($\alpha = 0.95$) maintains stable performance even as β increases, suggesting that higher exploitation sharpness can be tolerated when updates are sufficiently smoothed. In contrast, weak smoothing ($\alpha = 0.2$) causes performance to degrade rapidly as β increases, highlighting its inability to regulate over-sharp exploitation.

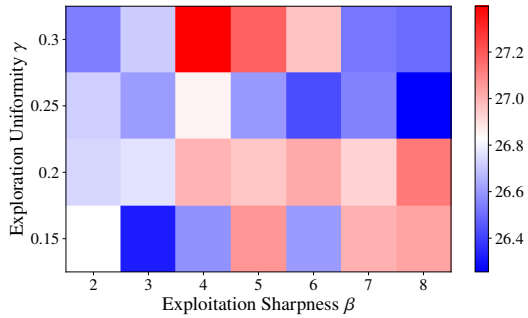
To better understand these behaviors, we visualize the dynamics of mixture proportions in Figure 6(b) under a fixed sharpness $\beta = 4$. Strong smoothing ($\alpha = 0.95$) produces smooth and gradually evolving proportions, while weaker smoothing ($\alpha = 0.2$) leads to highly volatile and unstable allocation, frequently switching focus among datasets. These results highlight the importance of EMA smoothing for stable and robust mixture learning.

Varying Update Interval

We evaluate how different update intervals affect overall performance. As shown in Figure 3, updating the sampling policy every 50 steps yields the best performance. While shorter intervals (e.g., 10 steps) enable quicker adaptation to recent rewards, they may introduce instability. Conversely, longer intervals (e.g., 500 steps) lead to slower adaptation and slightly degraded performance. These findings suggest that moderately frequent updates strike a good balance between responsiveness and stability in mixture optimization.

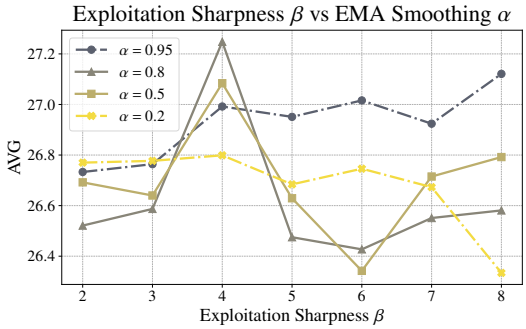


(a) As γ decreases (blue $\rightarrow$ orange), the best performance shifts toward higher β values, highlighting the trade-off between exploration and exploitation. Extremely large or small values of γ (solid lines) exhibit greater performance variance across different β values.

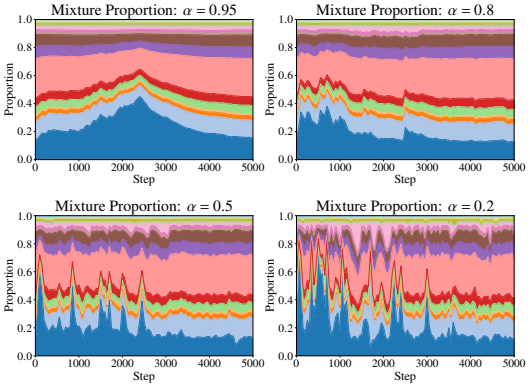


(b) Extremely large or small values of γ show a wide performance variance, as indicated by the spread from blue to red. This suggests that the effect of γ can be offset by selecting an appropriate β.

Figure 5: Performance Comparison varying Exploitation Sharpness β vs. Exploration Uniformity γ. Best viewed in color.



(a) Stronger smoothing (darker colors) tends to favor sharper β values, while weaker smoothing shifts the optimal β to lower values (lighter colors). This suggests that moderate smoothing factor (solid lines) effectively modulates reward sharpness, enabling more stable exploitation behavior.



(b) As α decreases, the mixture becomes more sensitive to reward signals, resulting in unstable and noisy dynamics.

Figure 6: Effect of EMA Smoothing Strength α on Performance Trends. Best viewed in color.

Reward Comparison with Δ-Loss vs. Δ-Entropy

To further explore the potential of the *1-Step Look-ahead Reward*, we investigate an alternative reward formulation based on entropy difference. As shown in Figure 4, using the entropy-based reward (Δ-Entropy) also yields performance gains over the baselines in both 1B and 3B. Notably, for the 3B model, Δ-Entropy achieves performance comparable to the Δ-Loss, suggesting that uncertainty reduction can serve as an effective signal for guiding mixture optimization. These results demonstrate the flexibility of our framework in accommodating different reward definitions while still improving overall performance.

Limitations

In this work, we focus dataset mixture optimization in the context of LLM post-training, where the dataset collection consists of large-scale, heterogeneous datasets aggregated without explicit domain structuring. Instead of conducting fine-grained analysis on individual datasets or domain-specific effects, our focus lies in optimizing mixture quality at scale across diverse sources. Moreover, our mixture policy operates at the dataset level, assigning sampling weights to entire datasets rather than individual instances. While this design enables efficient optimization and interpretable mixture control, instance-level mixture strategies remain an open direction for future work. Finally, due to computational constraints, our experimental evaluation is limited to relatively small-scale models (1B and 3B parameters). While these settings allow for controlled comparison, further studies on larger-scale LLMs or more extensive dataset collections are needed to fully assess the scalability of the proposed method.

Conclusion

We propose DYNAMIXSFT, a method for dynamic mixture optimization of instruction-tuning collections. We formulate the dataset mixture problem as a multi-armed bandit setup and introduce a *Prior-scaled Boltzmann Exploration*

to softly anchor the sampling distribution. Each dataset is treated as an arm and updated using a lightweight *1-Step Look-ahead Reward*, without requiring additional parameters or a validation set. We demonstrate the effectiveness of DYNAMIXSFT on the TÜLU-v2-mixture collection, without relying on human-crafted criteria. Furthermore, we provide extensive analyses and visualizations that illustrate how DYNAMIXSFT adapts mixture dynamics over time and highlights the contribution of each dataset during training.

Acknowledgements

This research was supported by the MSIT (Ministry of Science, ICT), Korea, under the Global Research Support Program in the Digital Field program (RS-2024-00436680) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation). And, this project was also supported by Microsoft Research Asia.

References

Albalak, A.; Pan, L.; Raffel, C.; and Wang, W. Y. 2023. Efficient Online Data Mixing For Language Model Pre-Training. arXiv:2312.02406.

Cachola, I.; Lo, K.; Cohan, A.; and Weld, D. 2020. TLDR: Extreme Summarization of Scientific Documents. In *Findings of EMNLP*.

Cesa-Bianchi, N.; Gentile, C.; Lugosi, G.; and Neu, G. 2017. Boltzmann Exploration Done Right. In *NeurIPS*.

Chaudhary, S. 2023. Code Alpaca: An Instruction-following LLaMA model for code generation. https://github.com/sahil280114/codealpaca.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; de Oliveira Pinto, H. P.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; Ray, A.; Puri, R.; Krueger, G.; Petrov, M.; Khlaaf, H.; Sastry, G.; Mishkin, P.; Chan, B.; Gray, S.; Ryder, N.; Pavlov, M.; Power, A.; Kaiser, L.; Bavarian, M.; Winter, C.; Tillet, P.; Such, F. P.; Cummings, D.; Plappert, M.; Chantzis, F.; Barnes, E.; Herbert-Voss, A.; Guss, W. H.; Nichol, A.; Paino, A.; Tezak, N.; Tang, J.; Babuschkin, I.; Balaji, S.; Jain, S.; Saunders, W.; Hesse, C.; Carr, A. N.; Leike, J.; Achiam, J.; Misra, V.; Morikawa, E.; Radford, A.; Knight, M.; Brundage, M.; Murati, M.; Mayer, K.; Welinder, P.; McGrew, B.; Amodei, D.; McCandlish, S.; Sutskever, I.; and Zaremba, W. 2021. Evaluating Large Language Models Trained on Code. arXiv:2107.03374.

Chen, X.; Lu, J.; Kim, M.; Zhang, D.; Tang, J.; Piché, A.; Gontier, N.; Bengio, Y.; and Kamalloo, E. 2025. Self-Evolving Curriculum for LLM Reasoning. arXiv:2505.14970.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

Dasigi, P.; Lo, K.; Beltagy, I.; Cohan, A.; Smith, N. A.; and Gardner, M. 2021. A Dataset of Information-Seeking Questions and Answers Anchored in Research Papers. In *NAACL*.

Dua, D.; Wang, Y.; Dasigi, P.; Stanovsky, G.; Singh, S.; and Gardner, M. 2019. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. In *NAACL*.

Fan, S.; Pagliardini, M.; and Jaggi, M. 2024a. DoGE: Domain Reweighting with Generalization Estimation. arXiv:2310.15393.

Fan, S.; Pagliardini, M.; and Jaggi, M. 2024b. DOGE: Domain Reweighting with Generalization Estimation. In *ICML*.

Gupta, H.; Kong, S. T.; Srikant, R.; and Wang, W. 2019. Almost Boltzmann Exploration. arXiv:1901.08708.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. In *ICLR*.

Ivison, H.; Wang, Y.; Pyatkin, V.; Lambert, N.; Peters, M.; Dasigi, P.; Jang, J.; Wadden, D.; Smith, N. A.; Beltagy, I.; and Hajishirzi, H. 2023. Camels in a Changing Climate: Enhancing LM Adaptation with Tulu 2. arXiv:2311.10702.

Köpf, A.; Kilcher, Y.; von Rütte, D.; Anagnostidis, S.; Tam, Z. R.; Stevens, K.; Barhoum, A.; Nguyen, D.; Stanley, O.; Nagyfi, R.; ES, S.; Suri, S.; Glushkov, D.; Dantuluri, A.; Maguire, A.; Schuhmann, C.; Nguyen, H.; and Mattick, A. 2023. OpenAssistant Conversations - Democratizing Large Language Model Alignment. In *NeurIPS*.

Lambert, N.; Morrison, J.; Pyatkin, V.; Huang, S.; Ivison, H.; Brahman, F.; Miranda, L. J. V.; Liu, A.; Dziri, N.; Lyu, S.; Gu, Y.; Malik, S.; Graf, V.; Hwang, J. D.; Yang, J.; Bras, R. L.; Tafjord, O.; Wilhelm, C.; Soldaini, L.; Smith, N. A.; Wang, Y.; Dasigi, P.; and Hajishirzi, H. 2025. Tulu 3: Pushing Frontiers in Open Language Model Post-Training. arXiv:2411.15124.

Lehman, E.; DeYoung, J.; Barzilay, R.; and Wallace, B. C. 2019. Inferring Which Medical Treatments Work from Reports of Clinical Trials. In *NAACL*.

Lian, W.; Goodson, B.; Pentland, E.; Cook, A.; Vong, C.; and "Teknium". 2023. OpenOrca: An Open Dataset of GPT Augmented FLAN Reasoning Traces. https://huggingface.co/datasets/Open-Orca/OpenOrca.

Lin, S.; Hilton, J.; and Evans, O. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *ACL*.

Liu, J.; Xia, C. S.; Wang, Y.; and ZHANG, L. 2023. Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. In *NeurIPS*.

Liu, Q.; Zheng, X.; Muennighoff, N.; Zeng, G.; Dou, L.; Pang, T.; Jiang, J.; and Lin, M. 2025. RegMix: Data Mixture as Regression for Language Model Pre-training. arXiv:2407.01492.

Longpre, S.; Hou, L.; Vu, T.; Webson, A.; Chung, H. W.; Tay, Y.; Zhou, D.; Le, Q. V.; Zoph, B.; Wei, J.; and Roberts, A. 2023. The Flan Collection: Designing Data and Methods for Effective Instruction Tuning. arXiv:2301.13688.

Luan, Y.; He, L.; Ostendorf, M.; and Hajishirzi, H. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In *EMNLP*.

Mallen, A.; Asai, A.; Zhong, V.; Das, R.; Khashabi, D.; and Hajishirzi, H. 2023. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In *ACL*.

Meta. 2024. Llama 3.2 Model Card. https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md.

Peng, B.; Li, C.; He, P.; Galley, M.; and Gao, J. 2023. Instruction Tuning with GPT-4. arXiv:2304.03277.

Qwen; :; Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Tang, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2025. Qwen2.5 Technical Report. arXiv:2412.15115.

Sutton, R. S.; Barto, A. G.; et al. 1998. *Reinforcement learning: An introduction*. MIT press Cambridge.

Suzgun, M.; Scales, N.; Schärli, N.; Gehrmann, S.; Tay, Y.; Chung, H. W.; Chowdhery, A.; Le, Q. V.; Chi, E. H.; Zhou, D.; and Wei, J. 2022. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them. arXiv:2210.09261.

Wadden, D.; Lin, S.; Lo, K.; Wang, L. L.; van Zuylen, M.; Cohan, A.; and Hajishirzi, H. 2020. Fact or Fiction: Verifying Scientific Claims. In *EMNLP*.

Wang, R.; Li, H.; Wu, M.; Wang, Y.; Han, X.; Zhang, C.; and Baldwin, T. 2024. Demystifying Instruction Mixing for Fine-tuning Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*.

Wang, W.; Wu, M.; Haddow, B.; and Birch, A. 2025. HBO: Hierarchical Balancing Optimization for Fine-Tuning Large Language Models. arXiv:2505.12300.

Wang, X.; Tsvetkov, Y.; and Neubig, G. 2020. Balancing Training for Multilingual Neural Machine Translation. In *ACL*.

Wang, Y.; Ivison, H.; Dasigi, P.; Hessel, J.; Khot, T.; Chandu, K.; Wadden, D.; MacMillan, K.; Smith, N. A.; Beltagy, I.; and Hajishirzi, H. 2023. How Far Can Camels Go? Exploring the State of Instruction Tuning on Open Resources. In *NeurIPS*.

Wei, J.; Bosma, M.; Zhao, V.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022. Finetuned Language Models are Zero-Shot Learners. In *ICLR*.

Wu, M.; Li, Y.; Zhang, M.; Li, L.; Haffari, G.; and Liu, Q. 2021. Uncertainty-Aware Balancing for Multilingual and Multi-Domain Neural Machine Translation Training. In *EMNLP*.

Wu, M.; Vu, T.-T.; Qu, L.; and Haf, R. 2024. Mixture-of-Skills: Learning to Optimize Data Usage for Fine-Tuning Large Language Models. In *EMNLP*.

Xiao, Y.; Zhang, S.; Zhou, W.; Ghassemi, M.; and Zhao, S. 2025. SFTMix: Elevating Language Model Instruction Tuning with Mixup Recipe. arXiv:2410.05248.

Xie, S. M.; Pham, H.; Dong, X.; Du, N.; Liu, H.; Lu, Y.; Liang, P.; Le, Q. V.; Ma, T.; and Yu, A. W. 2023. DoReMi: Optimizing Data Mixtures Speeds Up Language Model Pretraining. arXiv:2305.10429.

Xu, C.; Sun, Q.; Zheng, K.; Geng, X.; Zhao, P.; Feng, J.; Tao, C.; Lin, Q.; and Jiang, D. 2024. WizardLM: Empowering Large Pre-Trained Language Models to Follow Complex Instructions. In *ICLR*.

Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P.; YU, L.; Zhang, S.; Ghosh, G.; Lewis, M.; Zettlemoyer, L.; and Levy, O. 2023a. LIMA: Less Is More for Alignment. In *NeurIPS*.

Zhou, J.; Lu, T.; Mishra, S.; Brahma, S.; Basu, S.; Luan, Y.; Zhou, D.; and Hou, L. 2023b. Instruction-Following Evaluation for Large Language Models. arXiv:2311.07911.

Zhu, T.; Dong, D.; Qu, X.; Ruan, J.; Chen, W.; and Cheng, Y. 2025. Dynamic Data Mixing Maximizes Instruction Tuning for Mixture-of-Experts. In *NAACL*.

Appendices

Implementation Details

We report results using $\gamma = 0.3$, $\alpha = 0.95$, and $\beta = 4$ for the 1B model, and $\beta = 5$ for the 3B model. The learning rate is set to 1e-5, with a linear decay scheduler and a warmup ratio of 0.03.

Details of Dataset Collection

We utilize the TÜLU-v2-mixture collection, comprising ∼320K instances from 16 instruction-tuning datasets: FLAN (50K), FLAN-CoT (50K), Open Assistant 1 (7K), ShareGPT (114K), GPT4-Alpaca (20K), Code Alpaca (20K), LIMA (1K), WizardLM (30K), Open-Orca (30K), Hardcoded (140), and a science-related dataset (7K) combining Evidence Inference, Qasper, SciERC, SciFact, and SciTLDR.

Details of Evaluation Suite

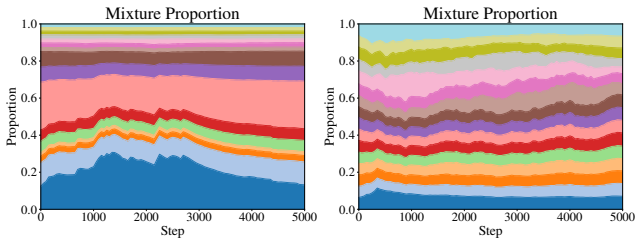
We evaluate our model on 10 benchmarks, following the recent TÜLU-3 evaluation setup, as detailed in Table 2.

Benchmark	CoT	Shots	Multiturn ICL	Metric
MMLU	✓	0	✗	EM
TQA	✗	6	✗	MC2
PopQA	✗	15	✓	EM
BBH	✓	3	✓	EM
DROP	✗	3	✗	F1
CHE	✗	0	✗	Pass@10
CHE+	✗	0	✗	Pass@10
GSM8K	✓	8	✓	EM
MATH	✓	4	✓	Flex EM
IFEval	✗	0	✗	Pass@1

Table 2: Evaluation Setup for DYNAMIXSFT.

Mixture Proportion without Prior Scaling

Figure 7 shows the effect of removing Prior-scaled Boltzmann Exploration. Without prior scaling, the sampling proportion starts from a uniform distribution and shows limited adaptability, in contrast to DYNAMIXSFT.



(a) DYNAMIXSFT (b) W/O PRIOR

Figure 7: Effect of removing the prior in DYNAMIXSFT.