DYNAMIXSFT: Dynamic Mixture Optimization of Instruction Tuning Collections

Haebin Shin[1,2,3*] **Lei Ji**[2] **Xiao Liu**[2] **Zhiwei Yu**[4]
Hyunwoo Yoo[5] **Qi Chen**[2] **Yeyun Gong**[2]

[1]University of Michigan [2]Microsoft Research [3]KAIST AI [4]BAAI [5]Drexel University
haebin@umich.edu

Abstract

As numerous instruction-tuning datasets continue to emerge, 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. We demonstrate that DYNAMIXSFT effectively optimizes the TÜLU-2-mixture and TÜLU-3-mixture collections across 10 benchmarks, while introducing minimal computational overhead over naive sampling. Furthermore, we provide a comprehensive analysis and visualizations to offer deeper insights into the adaptive dynamics of our method.

1 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.

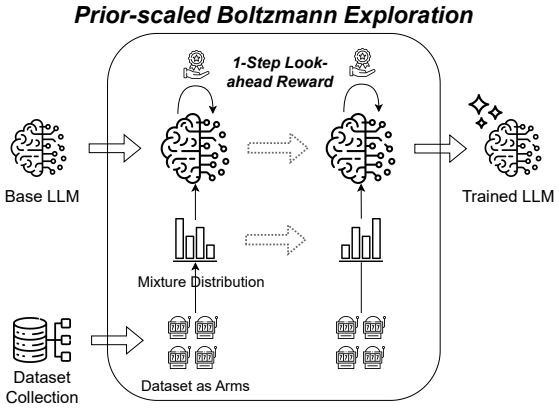


Figure 1: Overview of DYNAMIXSFT. Given a large collection of instruction-tuning datasets, DYNAMIXSFT treats each dataset as an arm in a *Multi-Armed Bandit* setup. The sampling policy dynamically evolves through *Prior-scaled Boltzmann Exploration*, periodically updated by a lightweight *1-Step Look-ahead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

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 to design effective mixing strategies. Several studies rely on additional models (Xie et al., 2023; Fan et al., 2024a; Liu et al., 2025; Wu et al., 2024; Wang et al., 2025) or resources (Wang et al., 2020; Wu et al., 2021) to guide dynamic dataset mixture optimization. This highlights the need for a lightweight, self-evolving method that can adapt the dataset mixture during training without additional computation.

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 Boltz-**

[*] Work done during internship at Microsoft Research

Findings of the Association for Computational Linguistics: ACL 2026, pages 39590–39603

mann 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-2-mixture (Ivison et al., 2023) and TÜLU-3-mixture (Lambert et al., 2025), comprising 16 and 19 instruction-tuning datasets, respectively. We observe consistent gains on LLaMA3.2 1B (Meta, 2024) and LLaMA3.1 8B (Grattafiori et al., 2024) across 10 benchmarks covering knowledge, reasoning, mathematics, coding, and instruction following.

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 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 large-scale instruction-tuning collections containing up to **19 datasets**, evaluating performance across **10 benchmarks** spanning knowledge, reasoning, mathematics, coding, and general instruction-following. Our method outperforms static and dynamic mixture baselines achieving consistent improvement on both 1B and 8B models with minimal computational overhead.

- 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.

2 Related Works

2.1 Dataset Mixture for LLMs

The composition of instruction-tuning data mixtures critically affects model effectiveness and generalization during post-training. The TÜLU collection (Wang et al., 2023; Ivison et al., 2023; Lambert et al., 2025) proposes this mixture by incorporating diverse high-quality datasets with unified formatting and expanded task coverage, but still relies on static or manually curated mixture strategies. We focus on dynamically optimizing the TÜLU-2-mixture (Ivison et al., 2023) and TÜLU-3-mixture (Lambert et al., 2025).

2.2 Mixture Optimization

Despite its importance, dataset mixture optimization remains challenging. Existing approaches are often limited in scope or flexibility. Some are confined to a small number of manually chosen dataset types (Wang et al., 2024) or operate at the regularization level rather than directly modifying sampling policies (Xiao et al., 2025). Others require pre-defined supervision from validation sets (Wang et al., 2020; Wu et al., 2021), introduce additional parameters through proxy models (Xie et al., 2023; Fan et al., 2024b; Liu et al., 2025) or actor networks (Wu et al., 2024; Wang et al., 2025), rely on MoE architectures (Zhu et al., 2025), lack generality across diverse tasks (Chen et al., 2025; Albalak et al., 2023). In contrast, DYNAMIXSFT targets large-scale, uncurated instruction-tuning collections covering diverse tasks without external supervision.

Dataset	General Knowledge			Reasoning		Coding		Mathematical		IFEval	AVG		
	MMLU	TQA	PopQA	BBH	DROP	CHE	CHE+	GSM8K	MATH		Total	Know+Rea	Code+Math
G1	29.87	41.77	16.35	32.56	27.37	36.03	31.83	9.62	2.38	22.73	25.05	29.58	19.97
G2	29.58	40.61	16.07	31.85	27.57	38.68	30.77	6.82	3.44	25.34	25.07	29.14	19.93
G3	28.68	38.31	15.84	31.88	27.62	38.93	32.26	14.93	2.90	24.58	25.59	28.47	22.26
G1 + G2	31.54	41.73	16.70	32.71	28.00	36.84	31.71	8.18	2.75	25.32	25.55	30.14	19.87
G2 + G3	30.72	39.99	16.31	32.28	28.20	41.32	33.53	14.10	2.99	28.09	26.75	29.50	22.99
G3 → G2 → G1	30.03	40.40	16.40	33.26	28.34	38.34	31.66	12.13	2.28	25.69	25.85	29.69	21.10
G1 → G2 → G3	29.35	40.94	16.74	32.89	27.57	38.78	32.25	14.25	3.34	28.09	26.42	29.50	22.16

Table 1: Performance trends across different data mixtures and training curricula. Different mixture groups favor different capabilities: G1 improves knowledge and reasoning, while G3 strengthens coding and math.

3 Preliminary: Does Data Mixture Matter?

Our hypothesis is that even when training on the same amount of data, different combinations can lead to qualitatively different learning behaviors. To explore the effect of their mixture and ordering, we conduct a simple pilot study using the TÜLU-2-mixture on LLaMA3.2 1B (Meta, 2024). We split the given collection into three equal groups (each sub-dataset is independently divided into three groups and then merged), according to token-averaged negative log-likelihood (NLL) scored by Qwen3-30B-A3B (Yang et al., 2025): high-loss (G1), medium-loss (G2), and low-loss (G3). In Table 1, training on G1 tends to improve general knowledge and reasoning benchmarks, while G3 shows stronger gains on coding and math tasks. Moreover, simply changing the curriculum order across the same partitions leads to different trends in overall performance.

These results suggest that even within a heterogeneous collection, there may exist state-dependent optimal mixtures rather than a single static recipe. This motivates us to dynamically adjust sampling during training to optimize the data mixture based on the model's evolving needs.

4 Dynamic Mixture Optimization

We introduce DYNAMIXSFT, a method that dynamically adjusts dataset mixtures during training Algorithm 1.

4.1 Mixture as a Multi-Armed Bandit

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 et al., 1998; Cesa-Bianchi et al., 2017; Gupta et al., 2019). A typical Boltzmann exploration (Sutton 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, treating all rewards as equally comparable across datasets can skew the sampling distribution toward specific datasets, failing to account for their inherent characteristics. In the SFT phase, 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.

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

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 probs:

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

    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

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.

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

4.2 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 current training progress. 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 additional supervision or permanent parameter updates. This 1-step look-ahead mechanism not only captures immediate learning signals but also serves as a proxy for long-term convergence, enabling state-dependent sampling that aligns with the model's evolving needs. We further provide theoretical support for the long-term effects of immediate reward signals in Appendix D.

The reward for dataset D_i is computed as follows:

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

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. We denote by B_{D_i} a mini-batch sampled from dataset D_i for reward estimation. To mitigate scale differences and occasional negative rewards, we normalize reward across datasets using min–max normalization. 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:

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

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.

5 Experimental Setup

5.1 Training Setting

We conduct experiments with two base models: LLaMA3.2 1B (Meta, 2024) and LLaMA3.1 8B (Grattafiori et al., 2024). For training data, we use the TÜLU-2 (Ivison et al., 2023) and TÜLU-3 (Lambert et al., 2025) collections, which consist of ∼320K examples from 16 datasets and about ∼930K examples from 19 datasets, respectively. Further dataset compositions and implementation details are provided in Appendix C and A.

5.2 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 et al., 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 evaluation details are provided in the Appendix B.

5.3 Baselines

We compare our method with two baseline groups, **Heuristic Methods**: Full Coverage (Proportional sampling) and Uniform Sampling; and **Dynamic Methods**: *MultiDDS* (Wang et al., 2020), *MultiUAT* (Wu et al., 2021) and *HBO* (Wang et al., 2025), adaptively adjust dataset mixtures based on gradient cosine similarity, uncertainty and difficulties.

6 Results

6.1 Overall Comparison

In Table 2, DYNAMIXSFT consistently outperforms other baselines across both 1B and 8B models. Compared to naive proportional sampling (Full Coverage), DYNAMIXSFT achieves relative improvements of up to +5.1% on TÜLU-2 and +5.3% on TÜLU-3 when averaged across 10 benchmarks (see Appendix A for statistical details). Furthermore, DYNAMIXSFT outperforms existing dynamic mixture methods such as HBO (Wang et al., 2025), which requires an additional actor network, confirming the effectiveness of our lightweight bandit approach.

6.2 Impact of Instance Coverage

Among the baselines, the Full Coverage (proportional sampling) approach shows relatively strong performance for both 1B and 8B 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.

6.3 Visualization 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).

7 Analysis

7.1 Exploitation vs. Exploration

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 5a, 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. Figure 5b visualizes this interaction via a heatmap, where the color gradient (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.

Method	General Knowledge			Reasoning		Coding		Mathematical		IFEval	AVG
	MMLU	TQA	PopQA	BBH	DROP	CHE	CHE+	GSM8K	MATH		
TÜLU 2 MIXTURE											
LLAMA3.2 1B											
Uniform Sampling	30.28	40.78	15.68	31.53	27.88	34.81	29.42	11.69	2.74	23.55	24.84
Full Coverage	32.13	41.12	16.22	32.93	28.18	37.23	35.47	15.77	2.91	24.78	26.67
MultiDDS	31.85	41.63	16.25	32.79	28.36	38.13	35.74	15.55	2.87	25.32	26.85
MultiUAT	32.11	41.59	16.12	33.52	28.37	37.28	34.33	14.51	2.81	26.18	26.68
HBO	31.75	41.77	16.21	32.81	28.68	38.21	35.81	15.49	2.81	26.38	26.99
DYNAMIXSFT	31.91	42.02	16.18	33.73	28.75	41.69	36.43	15.89	3.12	27.23	**27.70**
LLAMA3.1 8B											
Uniform Sampling	58.78	46.71	19.33	53.21	59.83	60.51	59.55	55.31	13.1	40.44	46.68
Full Coverage	61.80	49.40	23.30	57.10	61.70	66.90	63.10	60.40	14.00	42.30	50.00
MultiDDS	59.81	51.67	22.82	57.11	63.53	68.84	63.76	63.25	14.84	44.29	50.99
MultiUAT	59.13	51.62	22.54	57.89	63.11	68.38	63.66	63.12	14.93	44.31	50.87
HBO	58.33	52.32	22.21	59.44	63.81	67.91	64.30	62.14	15.24	44.12	50.98
DYNAMIXSFT	59.78	53.30	22.97	59.33	63.91	71.27	66.49	65.93	16.84	45.92	**52.57**
TÜLU 3 MIXTURE											
LLAMA3.2 1B											
Uniform Sampling	30.45	36.58	17.45	30.73	27.84	39.32	35.12	21.12	7.11	40.98	28.67
Full Coverage	31.09	37.80	18.26	32.01	28.02	43.66	42.71	28.27	8.21	44.17	31.42
MultiDDS	32.78	37.21	17.98	31.73	27.84	43.75	42.43	28.43	7.94	45.63	31.57
MultiUAT	32.51	37.46	18.28	31.87	27.21	44.32	43.15	28.11	8.13	45.33	31.64
HBO	30.79	37.63	18.02	32.34	27.94	43.97	42.82	28.82	8.57	45.87	31.68
DYNAMIXSFT	32.31	38.53	18.23	33.21	27.92	46.21	45.81	31.44	9.64	47.41	**33.07**
LLAMA3.1 8B											
Uniform Sampling	60.31	45.24	28.48	61.31	60.82	78.32	76.93	70.67	24.70	60.46	56.72
Full Coverage	62.10	46.80	29.30	67.90	61.30	86.20	81.40	76.20	31.50	72.80	61.55
MultiDDS	61.21	45.87	29.33	68.21	61.82	87.14	82.41	78.21	30.86	71.55	61.66
MultiUAT	61.24	47.14	29.11	68.18	60.12	88.23	82.13	77.91	31.41	70.13	61.56
HBO	61.91	46.53	28.58	68.32	62.52	88.11	83.08	77.18	32.53	73.14	62.19
DYNAMIXSFT	61.87	47.12	29.51	69.39	62.47	89.14	84.12	79.46	33.21	74.81	**63.11**

Table 2: Performance comparison across 10 evaluation suites. Best results are shown in dark red and second-best results in light red.

Ablations	AVG
DYNAMIXSFT	27.7
w/o prior	25.9
w/o prior + prop. init	26.3

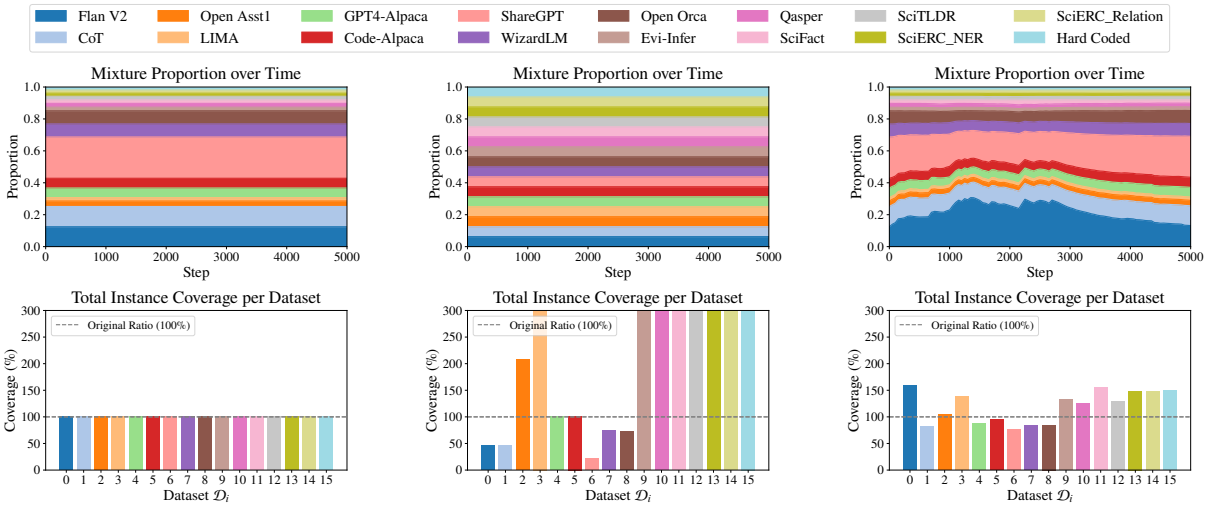
Table 3: Ablation study for the effect of prior distribution $p^{(0)}$ in Equation (2). W/O PRIOR + PROP. INIT denotes W/O PRIOR with proportional initialization.

7.2 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 6a, 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 updates. 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 6b 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



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

Figure 2: Visualization of Mixture Proportions and Instance Coverage under different sampling strategies. (a) *Full Coverage*: Mixture proportions are skewed based on dataset sizes (top), but all instances from each dataset are guaranteed to be used during training (bottom). (b) *Uniform Sampling*: Each dataset is sampled equally (top), leading to severe over- and under-sampling depending on dataset size (bottom). (c) DYNAMIXSFT: Our method adaptively adjusts mixture proportions over time (top), achieving well-balanced instance coverage across all datasets (bottom). All results are based on TÜLU-2-mixture.

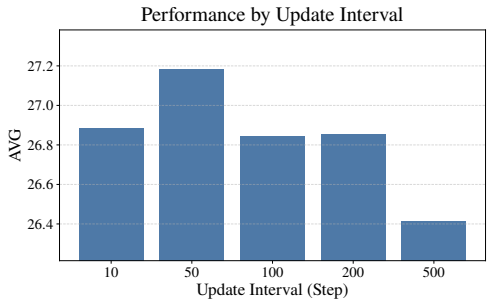


Figure 3: Comparison of Performance by Varying Reward Update Interval.

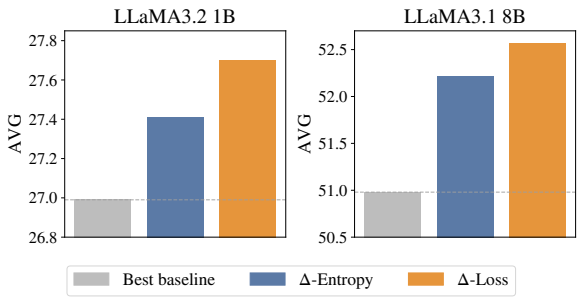


Figure 4: Comparison of Performance Using Δ-Loss vs. Δ-Entropy Rewards.

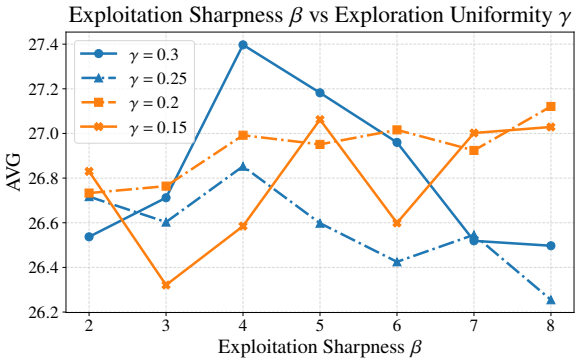
smoothing for stable and robust mixture learning.

7.3 Varying Update Interval

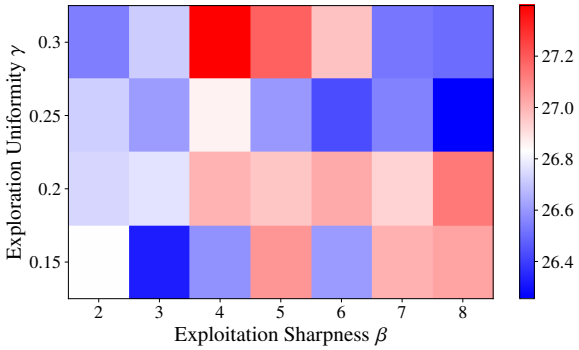
We evaluate how different update intervals affect overall performance. As shown in Figure 3, updating the sampling policy every 50 steps yields the best performance. While shorter intervals (e.g., 10 steps) enable quicker adaptation to recent rewards, they may introduce instability. Conversely, longer 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.

7.4 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 8B. Notably, for the 8B 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.

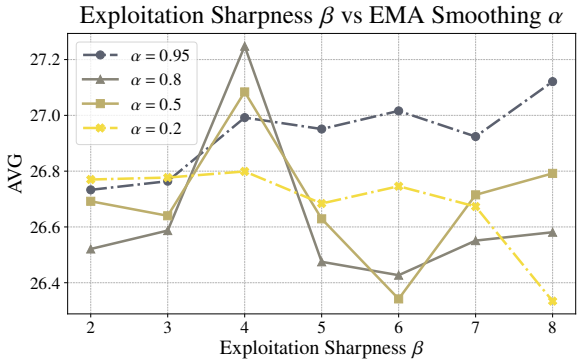


(a) As γ decreases (blue $\rightarrow$ orange), the best performance shifts toward higher β values, highlighting the trade-off between exploration and exploitation. Extremely large or small values of γ (solid lines) exhibit greater performance variance across different β values.

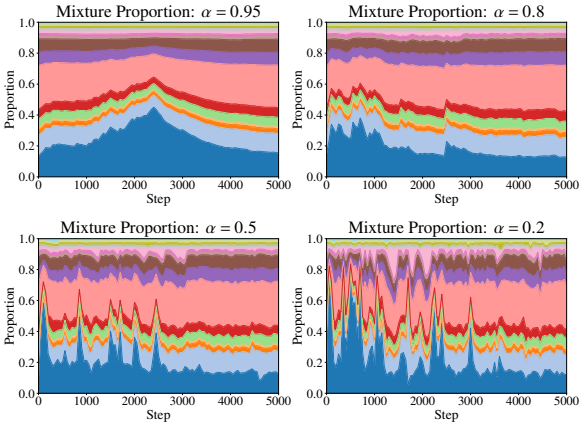


(b) Extremely large or small values of γ show a wide performance variance, as indicated by the spread from blue to red. This suggests that the effect of γ can be offset by selecting an appropriate β.

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.

7.5 Effect of Prior Scaling

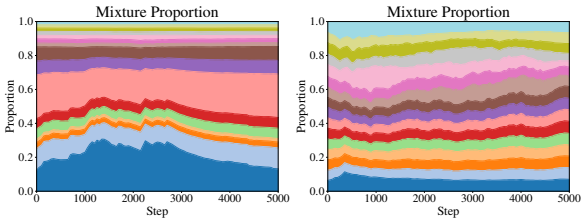
We analyze how the prior-scaled formulation $p^{(0)}$ in Equation (2) affects overall performance on the TÜLU-2-mixture with a 1B model. Table 3 shows that removing the $p^{(0)}$ (W/O PRIOR in Table 3) leads to a substantial performance drop. To better understand this effect, we compare the mixture dynamics with and without $p^{(0)}$ in Figure 7. Without prior-scaling, the sampling distribution remains close to its initial uniform distribution throughout training, showing that model fails to discover a good mixture.

Even when we manually set the initial distribution to be proportional (W/O PRIOR + PROP. INIT in Table 3), a performance gap still remains compared to the prior-scaled formulation. This demonstrates that prior-scaling not only naturally initializes the mixture near the data's intrinsic distribution, but also continuously anchors it throughout training, preventing drift during reward-based optimization.

8 Efficiency

We analyze the computational overhead of DYNAMIXSFT against existing dynamic data mixture approaches, under the TÜLU-3-mixture setup with $K = 19$ datasets and $T_{\text{update}} = 50$. In Table 4, DYNAMIXSFT introduces only $\sim$12.7% additional training cost over naive sampling, while prior methods incur substantially higher overhead (+139% to +760%).

This efficiency stems from our 1-Step Lookahead Reward, which relies solely on forward passes and requires no additional backward computation. Assuming the standard ratio where a forward pass accounts for approximately 1/3 of a full training step, the theoretical overhead is



(a) DYNAMIXSFT (b) W/O PRIOR

Figure 7: Effect of removing the prior in DYNAMIXSFT.

Method	Overhead vs. Naive
MultiDDS	+760% (8.6×)
MultiUAT	+380% (4.8×)
HBO	+139% (2.39×)
DYNAMIXSFT	+12.7% (1.13×)

Table 4: Computational overhead of dynamic data mixture methods relative to naive sampling.

$\frac{19}{50} \times \frac{1}{3} \approx 12.7\%$.

In contrast, existing approaches incur substantially higher overhead, as they rely on gradient computations across all datasets (Wang et al., 2020, 2025) or repeated stochastic forward passes for Monte Carlo estimation (Wu et al., 2021). Overall, DYNAMIXSFT achieves strong performance while maintaining minimal computational overhead, making it well-suited for large-scale data mixture training.

9 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 models or a validation set. We demonstrate the effectiveness of DYNAMIXSFT on the TÜLU-2-mixture and TÜLU-3-mixture collections, 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.

Limitations

In this work, we focus dataset mixture optimization in instruction-tuning stage, 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.

Ethical Considerations

This work employs the TÜLU collections, which is distributed under the ODC-BY license. The dataset combines multiple publicly available and web-sourced subsets, some of which are subject to non-commercial or mixed-use restrictions. All experiments in this study were conducted strictly for academic research purposes without any commercial intent. Given the inclusion of web-crawled data, a small portion of the corpus may inadvertently contain imperfect or biased content. We follow the data usage policies provided by the dataset maintainers and do not perform any additional data redistribution or manual modification. All experiments strictly comply with the dataset's research-only usage guidelines.

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

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

Faeze Brahman, Sachin Kumar, Vidhisha Balachandran, Pradeep Dasigi, Valentina Pyatkin, Abhilasha Ravichander, Sarah Wiegreffe, Nouha Dziri, Khyathi

Chandu, Jack Hessel, Yulia Tsvetkov, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. 2024. The art of saying no: Contextual noncompliance in language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 49706–49748. Curran Associates, Inc.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. 2020. TLDR: Extreme summarization of scientific documents. In *Findings of EMNLP*.

Nicolò Cesa-Bianchi, Claudio Gentile, Gabor Lugosi, and Gergely Neu. 2017. Boltzmann exploration done right. In *NeurIPS*.

Sahil Chaudhary. 2023. Code alpaca: An instruction-following llama model for code generation. https://github.com/sahil280114/codealpaca.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. 2025. Self-evolving curriculum for llm reasoning. *Preprint*, arXiv:2505.14970.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *NAACL*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *NAACL*.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. 2024a. Doge: Domain reweighting with generalization estimation. *Preprint*, arXiv:2310.15393.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. 2024b. DOGE: Domain reweighting with generalization estimation. In *ICML*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Harsh Gupta, Seo Taek Kong, R. Srikant, and Weina Wang. 2019. Almost boltzmann exploration. *Preprint*, arXiv:1901.08708.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *ICLR*.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *Preprint*, arXiv:2311.10702.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations - democratizing large language model alignment. In *NeurIPS*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2025. Tulu 3: Pushing frontiers in open language model post-training. *Preprint*, arXiv:2411.15124.

Eric Lehman, Jay DeYoung, Regina Barzilay, and Byron C. Wallace. 2019. Inferring which medical treatments work from reports of clinical trials. In *NAACL*.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024. Numinamath tir. https://huggingface.co/AI-MO/NuminaMath-TIR.

Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2024. Table-gpt: Table fine-tuned gpt for diverse table tasks. *Proceedings of the ACM on Management of Data*, 2(3).

Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Openorca: An open dataset of gpt augmented flan reasoning traces. https://huggingface.co/datasets/Open-Orca/OpenOrca.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *ACL*.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and LINGMING ZHANG. 2023. Is your code generated by chatGPT really correct? rigorous evaluation of large language models for code generation. In *NeurIPS*.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. 2025. Regmix: Data mixture as regression for language model pre-training. *Preprint*, arXiv:2407.01492.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. *Preprint*, arXiv:2301.13688.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *EMNLP*.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Wizardcoder: Empowering code large language models with evol-instruct. *Preprint*, arXiv:2306.08568.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 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`.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *Preprint*, arXiv:2304.03277.

Nazneen Rajani, Lewis Tunstall, Edward Beeching, Nathan Lambert, Alexander M. Rush, and Thomas Wolf. 2023. No robots. `https://huggingface.co/datasets/HuggingFaceH4/no_robots`.

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Souza Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergün, Ifeoma Okoh, and 14 others. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. *Preprint*, arXiv:2402.06619.

Richard S Sutton, Andrew G Barto, and 1 others. 1998. *Reinforcement learning: An introduction*. MIT press Cambridge.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *Preprint*, arXiv:2210.09261.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. 2024. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. *Preprint*, arXiv:2410.01560.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *EMNLP*.

David Wadden, Kejian Shi, Jacob Morrison, Alan Li, Aakanksha Naik, Shruti Singh, Nitzan Barzilay, Kyle Lo, Tom Hope, Luca Soldaini, Shannon Zejiang Shen, Doug Downey, Hannaneh Hajishirzi, and Arman Cohan. 2025. SciRIFF: A resource to enhance language model instruction-following over scientific literature. In *EMNLP*.

Renxi Wang, Haonan Li, Minghao Wu, Yuxia Wang, Xudong Han, Chiyu Zhang, and Timothy Baldwin. 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)*.

Weixuan Wang, Minghao Wu, Barry Haddow, and Alexandra Birch. 2025. Hbo: Hierarchical balancing optimization for fine-tuning large language models. *Preprint*, arXiv:2505.12300.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020. Balancing training for multilingual neural machine translation. In *ACL*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. How far can camels go? exploring the state of instruction tuning on open resources. In *NeurIPS*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *ICLR*.

Minghao Wu, Yitong Li, Meng Zhang, Liangyou Li, Gholamreza Haffari, and Qun Liu. 2021. Uncertainty-aware balancing for multilingual and multi-domain neural machine translation training. In *EMNLP*.

Minghao Wu, Thuy-Trang Vu, Lizhen Qu, and Reza Haf. 2024. Mixture-of-skills: Learning to optimize data usage for fine-tuning large language models. In *EMNLP*.

Yuxin Xiao, Shujian Zhang, Wenxuan Zhou, Marzyeh Ghassemi, and Sanqiang Zhao. 2025. Sftmix: Elevating language model instruction tuning with mixup recipe. *Preprint*, arXiv:2410.05248.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. 2023. Doremi: Optimizing data mixtures speeds up language model pretraining. *Preprint*, arXiv:2305.10429.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *ICLR*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. Lima: Less is more for alignment. In *NeurIPS*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023b. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

Tong Zhu, Daize Dong, Xiaoye Qu, Jiacheng Ruan, Wenliang Chen, and Yu Cheng. 2025. Dynamic data mixing maximizes instruction tuning for mixture-of-experts. In *NAACL*.

A Implementation Details

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 × A100 40GB GPUs. In accordance with the each dataset collection, the number of arms K is set to 16 for TÜLU-2 and 19 for TÜLU-3. We report results using $\gamma = 0.3$, $\alpha = 0.95$, and $\beta = 4$ for the 1B model, and $\beta = 5$ for the 8B model. The learning rate is set to 1e-5, with a linear decay scheduler and a warmup ratio of 0.03. Reported results are averaged over three runs. The standard deviations for DYNAMIXSFT were consistently small ($\sigma < 0.25$), and a t-test against the primary baseline yielded p-value < 0.05, confirming the statistical significance of the improvements.

B Details of Evaluation Suite

We evaluate our model on 10 benchmarks, following the TÜLU-3 evaluation setup (Lambert et al., 2025), as detailed in Table 5.

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 5: Evaluation Setup for DYNAMIXSFT.

C Details of Dataset Collection

We utilize the TÜLU-2-mixture and TÜLU-3-mixture collections. TÜLU-2-mixture comprises ∼320K instances from 16 instruction-tuning datasets: FLAN (50K) (Longpre et al., 2023), FLAN-CoT (50K) (Longpre et al., 2023), Open Assistant 1 (7K) (Köpf et al., 2023), ShareGPT (114K), GPT4-Alpaca (20K) (Peng et al., 2023), Code Alpaca (20K) (Chaudhary, 2023), LIMA (1K) (Zhou et al., 2023a), WizardLM (30K) (Xu et al., 2024), Open-Orca (30K) (Lian et al., 2023), Hardcoded (140) (Ivison et al., 2023), and a science-related dataset (7K) combining Evidence Inference (Lehman et al., 2019), Qasper (Dasigi et al., 2021), SciERC (Luan et al., 2018), SciFact (Wadden et al., 2020), and SciTLDR (Cachola et al., 2020).

TÜLU-3-mixture contains ∼930K instances from 19 datasets: CoCoNot (10K) (Brahman et al., 2024), FLAN v2 (89K) (Longpre et al., 2023), No Robots (9K) (Rajani et al., 2023), OpenAssistant Guanaco (7K) (Köpf et al., 2023), Tulu 3 Persona MATH (149K), Tulu 3 Persona GSM (49K), Tulu 3 Persona Python (34K), Tulu 3 Persona Algebra (20K), Tulu 3 Persona IF (29K), NuminaMath-TIR (64K) (LI et al., 2024), OpenMathInstruct 2 (50K) (Toshniwal et al., 2024), Tulu 3 WildGuardMix (50K), Tulu 3 WildJailbreak (50K), Tulu 3 Hardcoded (240), Aya (100K) (Singh et al., 2024), WildChat GPT-4 (100K) (Zhao et al., 2024), TableGPT (5K) (Li et al., 2024), SciRIFF (10K) (Wadden et al., 2025). Evol CodeAlpaca (107K) (Luo et al., 2023).

D Long-term Effects of Immediate Reward

We demonstrate that optimizing the 1-Step Look-ahead Reward (immediate objective) is theoretically equivalent to achieving optimal convergence in the long term.

Given a loss function $\mathcal{L}(\theta)$ and standard gradient update $\theta_{t+1} = \theta_t - \eta \nabla \mathcal{L}_k(\theta_t)$, let's consider the immediate reward:

$$r_{t,k} = \mathcal{L}(\theta_t) - \mathcal{L}(\theta_{t+1}). \tag{6}$$

Assuming $\mathcal{L}$ is smooth and the learning rate η is sufficiently small, we can approximate $\mathcal{L}(\theta_{t+1})$ using the first-order Taylor expansion around θ_t:

$$\mathcal{L}(\theta_{t+1}) \approx \mathcal{L}(\theta_t) - \eta ||\nabla \mathcal{L}(\theta_t)||^2, \tag{7}$$

and rearranging the terms to express the reward:

$$r_{t,k} = \mathcal{L}(\theta_t) - \mathcal{L}(\theta_{t+1}) \approx \eta ||\nabla \mathcal{L}(\theta_t)||^2. \tag{8}$$

Selecting dataset k to maximize the 1-step reward $r_{t,k}$ is effectively equivalent to maximizing the descent magnitude along the gradient direction. Therefore, our reward does not track raw loss value, it explicitly measures learning progress.

Why MAB is Appropriate? The Multi-Armed Bandit (MAB) framework is naturally suited for this task because it optimizes the *cumulative reward* $\sum_{t=0}^{T} r_t$ over the entire training horizon. In

our setting, the reward is defined as the immediate loss reduction: $r_t = \mathcal{L}(\theta_t) - \mathcal{L}(\theta_{t+1})$. By the telescoping property, the cumulative reward is:

$$\sum_{t=0}^{T} r_t = \sum_{t=0}^{T} [\mathcal{L}(\theta_t) - \mathcal{L}(\theta_{t+1})] = \mathcal{L}(\theta_0) - \mathcal{L}(\theta_{T+1}) \tag{9}$$

Consequently, using the Δ-Loss reward within a bandit policy is not just a heuristic trick, it is a principled policy that prioritizes datasets yielding greater long-term loss reduction while maintaining essential exploration.