REVISITING DATA MIXING THROUGH THE LENS OF MULTI-OBJECTIVE OPTIMIZATION

Hoang Phan
phanviethoang1512@gmail.com

ABSTRACT

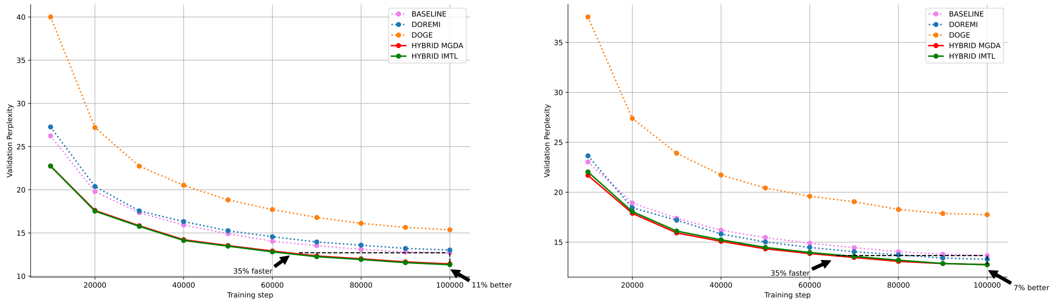
Effective pretraining of large language models (LLMs) relies significantly on the strategic composition of training data from various sources. Traditional domain weighting approaches often focus on minimizing either average empirical loss or worst-case domain loss, which can lead to overfitting to either simple or complex domains. To address these limitations, we recast data mixing as a multi-objective optimization problem, enabling the application of multi-objective optimization theory. Furthermore, we propose a hybrid method that leverages both data resampling and domain loss reweighting to directly address the mismatch between the training of proxy models and their base counterparts. Empirically, we evaluate our methodology against established baselines on The Pile, SlimPajama, and Wiki40b datasets, demonstrating its superiority in enhancing performance across diverse domains by speeding up the convergence of the 1B model to 40% compared to traditional training. Our extensive experiments show that our approach not only improves modeling ability across training domains but also surpasses prior methods on downstream tasks.

1 INTRODUCTION

Optimal data representation in the pretraining of large language models (LLMs) remains a pivotal challenge that directly impacts model utility and efficiency (Chowdhery et al., 2023; Touvron et al., 2023). It is often the case that an LLM pre-training project will have a limited token budget for training and available data sources with a much larger combined token count. Decisions will need to be made about how much data to include from each source.

Learning multiple tasks simultaneously can be a challenging optimization problem because it involves multiple objectives (Vandenhende et al., 2021). The most popular multi-task learning (MTL) objective in practice is the average loss over all tasks. Even when this average loss is exactly the true objective (as opposed to only caring about some specific tasks), directly optimizing the average loss could lead to undesirable performance, e.g. the optimizer struggles to make progress so the learning performance significantly deteriorates. A known cause of this phenomenon is the conflicting gradients (Yu et al., 2020): gradients from different tasks 1) may have varying scales with the largest gradient dominating the update, and 2) may point in different directions so that directly optimizing the average loss can be quite detrimental to a specific task's performance. Base on these findings, we show how current optimization-base methods for data mixture problem are sub-optimal and better training methods should be adopted.

While the question of data mixtures has gained interest in the recent years, the prevalent strategy of using heuristically determined domain weights or optimizing weights based on limited downstream tasks is increasingly recognized as inadequate for dealing with the complex dynamics of domain interactions (Gao et al., 2020; Du et al., 2022). Driven by the need for a principled optimizer, we introduce a multi-objective optimization technique to determine the most effective domain mixing ratios. This strategy aims to maximize the general learning capabilities of LLMs by deriving weights that can benefit every domain without overfitting to particular ones. Implemented on the diverse domains of The Pile (Gao et al., 2020) and SlimPajama (Soboleva et al., 2023) datasets, our approach offers a promising approach toward more effective and generalizable language models, demonstrating significant improvements in learning efficiency and domain robustness. Figure 1 shows that our proposed methods require 65% of the tokens (35% speed up) versus baseline on both of the datasets.



(a) The Pile

(b) SlimPajama

Figure 1: **Validation perplexity**, unweighted average over 22 domains from The Pile and 7 domains from SlimPajama. HYBRID multi-objective optimization methods require 65% of the tokens (35% speed up) versus baseline on both of the datasets. At the end of the training, they obtained 11% and 7% on The Pile and SlimPajama, respectively

At the end of the training, they obtained 11% and 7% on The Pile and SlimPajama, respectively. In summary, our contributions could be summarized as follows:

- We first reformulate the data mixture problem as a multi-objective optimization problems, from which we can leverage theoretical-grounded methods for optimizing different training losses.
- We point out the mismatch between the training of the proxy model and the base model. Based on this observation, we propose hybrid approaches to the data mixing problem that could give us additional degrees of freedom to either learning domains without biases or incorporate preference during training, targetting at improving downstream performance.
- On well-established benchmarks, we conduct extensive experiments to validate our proposed method. Empirically, our methods are not only effective on different datasets but also robust to training conditions and allow practitioners to transfer the obtained mixing ratios to different model sizes or tokenizers.

2 Background

In this section, we first present the concept of multi-objective optimization (MOO). Then we introduce data mixture problem set up and discuss how finding the optimal data mixing ratio is inherently a multi-objective optimization.

2.1 Multi-objective optimization

Given m objective functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}, i \in [m]$ parameterized by $\mathbf{x} \in \mathbb{R}^n$, the multi-objective optimization problem could be formulated as follows:

$$\min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) := [f_1(\mathbf{x}), \cdots, f_S(\mathbf{x})] \tag{1}$$

A key difference between sing-objective and multi-objective optimization is that there is often no optimal solution for a MOO problem that is better than all other solutions in every individual objective. Due to this conflict nature among objectives, one is often interested in (i) finding Pareto solutions, from which we can not improve all objectives simultaneously or searching for an updating direction that can optimize all objectives without biases.

Pareto dominance: A solution $x_1 \in \mathbb{R}^n$ is said to be dominated by solution $x_2 \in \mathbb{R}^n$ if and only if: $\forall i \in [m], f_i(x_2) \leq f_i(x_1)$, and $\exists\ j \in [m]$ such that $f_j(x_2) < f_j(x_1)$, denoted by $\mathbf{F}(x_2) \succ \mathbf{F}(x_1)$.

Pareto optimality A solution x^* is Pareto optimal if and only if it is not dominated by any point in $\mathbb{R}^n$. The set of all Pareto solutions is called the Pareto set and the image of those Pareto optimal solutions is called the Pareto front.

In the context of deep learning, x often represents model parameter while $\{f_i(\cdot)\}_{i=1}^m$ is the list of objectives of interest (e.g. classification, regression losses...). In practice, finding a Pareto solution is intractable due to non-convex objectives and gradient-based optimizers. Instead, prior work often search for the Pareto local optimal set containing solutions that are Pareto optimal on a neighborhood of itself.

2.2 Data mixture

We start by describing the data mixing setup: Given K training domains (e.g. Arxiv, Book, ...): $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K$. A language model θ is trained on this data composition , our goal is to find model parameters θ^* that minimize the domains' respective losses: $\ell_1(\theta) = \mathbb{E}_{\mathcal{D}_i}\ell(\theta)$:

$$min_\theta[\ell_1(\theta), \ell_2(\theta), ..., \ell_K(\theta)] \tag{2}$$

The overall objective is in fact a vector-value next-token prediction loss function over training domains. The data mixing problem aims to find the α to sample domains' data. The final data mixture used to train the full-size language model is constructed by first sampling a domain according to the domain-wise distribution α, followed by uniformly sampling a batch B from that domain.

The formula 2 allows us to leverage existing results from multi-objective optimization theory. From which we found that, simply minimizing the average/worst case domain loss may result in putting more weights on easy domains, which potentially cause the overfitting to those domains. Thus, in this paper, we propose to either adopt impartial multi-objective optimization solvers that aim to universally minimize domain losses without bias toward any specific domain or incorporate user preference during the training, targeting at improving downstream performance.

3 Proposed method

In the following, we will describe how to optimize α guided by training a proxy model of parameters θ on $\mathcal{D}_{train}$. We denote by $\ell_i(\theta)$ the expected next token prediction loss of the proxy model on domain $\mathcal{D}_i$.

In this paper, we propose to adopt multi-objective optimizers for minimizing multiple domain losses, namely MGDA (Sener & Koltun, 2018) and IMTL (Liu et al.). In short, MGDA finds the minimum-norm point in the convex hull composed by the gradients of multiple objectives while IMTL searches for the updating direction that has equal projections on per-objective gradients $\{g_i\}$ (scale-invariant).

$$\boldsymbol{g}_{MGDA} = \text{argmin}_{g \in \mathcal{CH}} ||g||_2$$

where

$$\mathcal{CH} = \left\{ \boldsymbol{G}^\top \boldsymbol{\alpha} = \sum_{i=1}^{k} \alpha_i \boldsymbol{g}_i \mid \alpha \in \Delta_k \right\}$$

MGDA theoretically guarantees that the obtained solution is a Pareto stationary point, from which we can not simultaneously improve an specific object without hurting another while IMTL presents a closed-form solution, which learns shared parameters without any bias.

Formally, let $\{\boldsymbol{u}_t = \boldsymbol{g}_t / \|\boldsymbol{g}_t\|\}$, IMTL searches for the updating direction g s.t. $\boldsymbol{g}\boldsymbol{u}_1^\top = \boldsymbol{g}\boldsymbol{u}_t^\top \Leftrightarrow \boldsymbol{g}\left(\boldsymbol{u}_1 - \boldsymbol{u}_i\right)^\top = 0, \forall 2 \leqslant i \leqslant k$, which yields the following solution $\sum_{i=1}^k \alpha_i \boldsymbol{g}_i$ where $\boldsymbol{\alpha} = \boldsymbol{g}_1 \boldsymbol{U}^\top \left(\boldsymbol{D}\boldsymbol{U}^\top\right)^{-1}$.

Overall, our proposed method is demonstrated in Figure 2, which consists of two separate stages. In the first stage, we train a proxy model using multi-objective optimizers (e.g. MGDA, IMTL or EPO). We then take the average mixing ratio during the training of the proxy model to either resample the data or reweight the domain losses. In practice, we find that using hybrid methods, uniformly sample data from every domains then reweighting domain losses based on obtained mixing ratio, often yield best results.

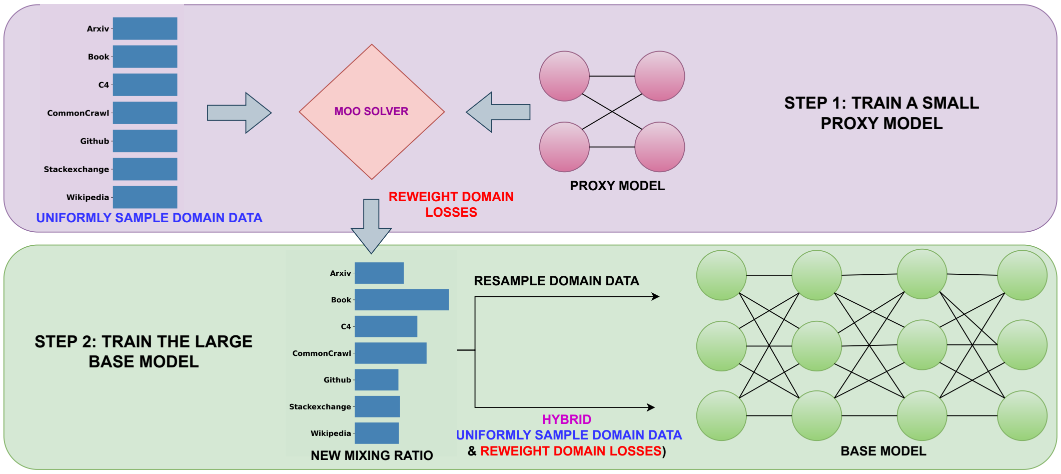


Figure 2: Overview of our proposed method.

4 Related work

Conceptually, approaches to the optimal data mixture problem can be divided into the following categories:

Optimization-based approaches Prior optimization-based methods for mixing ratio identification: DoReMi (Xie et al., 2024) employs GroupDRO (Sagawa et al., 2021), which was proposed for the problem of "learning from multiple groups", where the training data is dominated by one of the group. The updating formula of DoReMi is:

$$\min_{\theta} \max_{\alpha \in \Delta^k} L(\theta, \alpha) := \sum_{i=1}^{k} \alpha_i \cdot \left[\frac{1}{\sum_{x \in D_i} |x|} \sum_{x \in D_i} \ell_\theta(x) - \ell_{\text{ref}}(x) \right]$$

However, DoReMi/GroupDRO focus solely on worst group and tend to neglect the knowledge transfer between groups. DOGE (Fan et al., 2024) mitigates this drawback to some extent by minimizing the average validation loss across domain using bi-level optimization.

$$\begin{aligned} \boldsymbol{\alpha} \in \arg\min_{\boldsymbol{\alpha} \in \Delta^k} \textstyle\sum_{i \in [k]} \ell_i \left(\boldsymbol{\theta}^\star(\boldsymbol{\alpha}) \right) \\ \text{s.t. } \boldsymbol{\theta}^\star(\boldsymbol{\alpha}) \in \arg\min_{\boldsymbol{\theta}} \textstyle\sum_{i \in [k]} \alpha_i \ell_i(\boldsymbol{\theta}) \end{aligned}$$

However, the average loss among domains might not be a good proxy for indicating LM performance due to the large variation in loss magnitude between domains. In practice, the perplexities vary from 4.x to 44.x across training domains (Fan et al., 2024) and a low average training loss does not guarantee good performance on held-out datasets, which we will empirically verify in the experiment section.

Inspired by the use of multi-armed bandits (MAB) for auxiliary data selection in few-shot LLM fine-tuning, Online Data Mixing (ODM) (Albalak et al., 2023) treats each data domain as an MAB arm. They develop an algorithm to optimize the data mixing distribution dynamically, adapting to training changes. By leveraging information theory, specifically using perplexity as a measure of model uncertainty and expected information gain, ODM aims to increase the mixing ratio for the most informative domains. The training loss per domain is used as a reward to guide this process.

Data scaling laws approaches

Data scaling laws explore interactions of data quantity, quality, and mixing proportions, as LLMs are scaled up. Muennighoff et al. (2024) introduce scaling laws for data-constrained scenarios and Goyal et al. (2024) try to extend this approach to deal with multiple data pools. Prior research has confirmed that different datasets require different scaling (Hoffmann et al., 2022; Pandey, 2024), thus Ye et al.

(2024) and Ge et al. (2024) propose functional relationships to predict the impact of mixtures on language modeling loss.

5 Experimental results

We first analyze the convergence behaviors of different multi-objective optimizers on a toy example. We illustrate how well-developed methods for multi-objective optimization can surpass prior optimizers for the data mixture problem. Then, we empirically demonstrate the effectiveness of the proposed approach on various data mixture benchmarks. Due to space limit constraints, detailed training configurations and additional results are deferred to the Appendix.

5.1 Toy example

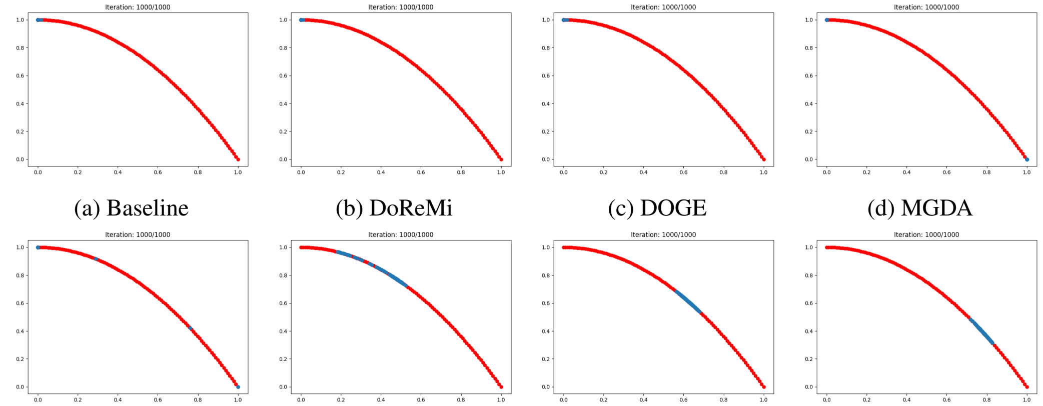
For the toy optimization example, we examine the behavior of Baseline, DoReMi, DOGE, MGDA, IMTL and EPO on the ZDT-2 problem (Zitzler et al., 2000). The obtained solution by each method is indicated in blue points in Figure 3 while the Pareto front is represented by the red curve.

$$\min \big(f_1(x), f_2(x)\big) = \big(x_1, g(x) h\left(x_1, g(x)\right)\big) \tag{3}$$

where $0 \leq x_i \leq 1 \; \forall i \in [30]$ and $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i$; $h\left(f_1, g\right) = 1 - \left(f_1/g\right)^2$.

The Pareto solution of ZDT-2 is given by $0 \leq x_1^* \leq 1$ and $x_i^* = 0$ for $i = 2, \ldots, 30$.

Overall, those methods that are currently used for the data mixture problem are biased toward $f_1(x)$, which is a linear objective and easy to optimize. In contrast, MGDA and IMTL can obtain other solutions that achieve low values for $f_2(x)$ while EPO can find x based on our desired preferences (e.g. such that $f_1(x) \approx 2f_2(x)$ or $f_1(x) \approx f_2(x)$).



(a) Baseline (b) DoReMi (c) DOGE (d) MGDA

(e) IMTL (f) EPO (2,1) (g) EPO (1,1) (h) EPO (1,2)

Figure 3: While Baseline, DoReMi, and DOGE primarily focus on the first (easier) objective, MGDA and IMTL are capable of obtaining solutions that achieve low loss on the second objective as well. Moreover, EPO enables the adjustment of preferences to focus more or less on individual objectives .

5.2 Training and evaluation

Datasets For our experiments, we use The Pile (Gao et al., 2020), an 825Gb open-sourced language modeling dataset comprising 22 smaller datasets from various domains including Wikipedia, Github, and PubMed Central. Similarly, we include SlimPajama (Soboleva et al., 2023) which is a deduplicated version of RedPajama (Computer, 2023) consisting of data from 7 domains (Arxiv, Book, C4, CommonCrawl, Github, Stackexchange, Wikipedia) and Wiki40B (Guo et al., 2020) for a multilingual setting, where we train the model on 13 different languages.

Training setup For experiments on The Pile and SlimPajama, we train a small 82M decoder-only transformer (Vaswani, 2017) as the proxy model for domain reweighting for 10k iterations then train a 1 billion parameter model as the base model. We train both the proxy and base models using a batch size of 16 sequences per GPU, and accumulate gradients across 8 GPUs in parallel (G = 8) to reach a total batch size of 128 samples. We train for a total of 100k steps with a maximum token length of 512, reaching 6.5 billion tokens. Similarly, for the multi-lingual experiment, we train 155M base models with the global batch size of 64 using a smaller-scale hardware (A100 40GB versus A100 80GB).

Evaluation protocol To validate the performance of our approach and the baselines, we compute perplexity on held-out validation and test data from each domain. We measure the average perplexity across all domains and 5-shot reasoning ability across a series of reasoning tasks, covering diverse knowledge fields including physics, social science, logic inference etc.: ARC easy (Clark et al., 2018), BLiMP (Warstadt et al., 2020), Copa, RTE, WiC (Wang et al., 2019), LAMBADA standard and OpenAI Paperno et al. (2016), LogiQA (Liu et al., 2021), MC Taco (Zhou et al., 2019), MuTual (Cui et al., 2020), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), QQP, SST-2 (Wang et al.), Social IQA (Sap et al., 2019), and WinoGrande (Sakaguchi et al., 2020). We use LM-eval Harness (Gao et al., 2024) to assess the few-shot reasoning performance.

Baselines We consider the following data mixture approaches: (i) optimization based methods: DoReMi (Xie et al., 2024), DOGE (Fan et al., 2024) and (ii) scaling-law methods: REGMIX (Liu et al., 2024) and BIMIX (Ge et al., 2024) apart from the original mixture baseline. While DOGE and our methods train a single 82M proxy model, DoReMi trains a reference model and a proxy model with the same size of 82M. We directly take the reported mixing ratio of BIMIX, which is produced by fitting $5K$ scaling-law coefficients (i.e. 110 for The Pile and 35 for SlimPajama). For REGMIX, since they did not experiment with the same datasets, we train 64 proxy models of size 27M to produce the mixing ratio using their released codebase for each dataset.

5.3 Experimental results

The average validation across domains on The Pile and SlimPajama after 100k iterations are reported in Table 1. It is interesting that optimization-based methods like DoReMi and DOGE are worse than the original baseline. By contrast, scaling-law methods REGMIX and BIMIX perform better but at the cost of training multiple proxy models. While using a single proxy model, multi-objective optimization methods illustrate superior performance, and enhance the pretraining over baseline on all datasets. Furthermore, our hybrid methods can improve themselves by correcting the mismatch between the training of the base and the proxy models, i.e. both employing uniform sampling and reweighting the loss functions.

Table 1: **Average validation perplexity across domains**: We calculate the unweighed average validation perplexity of 1B models across 22 domains of The Pile and 210M models 7 domains of SlimPajama, respectively.

								HYBRID	HYBRID
Dataset	**BASELINE**	**DOREMI**	**DOGE**	**REGMIX**	**BIMIX**	**MGDA**	**IMTL**	**MGDA**	**IMTL**
The Pile	12.71	13.03	15.36	13.00	11.69	11.75	11.66	11.42	11.32
SlimPajama	13.66	13.29	17.75	12.94	12.92	12.81	12.81	12.75	12.74

The performance of baselines on few-shot reasoning tasks is presented in Table 2, from which we can observe the impacts of different data mixtures on downstream tasks. In particular, BIMIX is the only method among baselines that can surpass baseline in terms of average score. Multi-objective optimizers still exhibit superior performance, with up to 1 point of improvement over the original baseline.

Table 3 shows per-domain validation perplexity on 13 different languages from Wiki40B. In this experiment, DOGE and DoReMi achieve 3.9% and 1.3% improvements over the original baseline, respectively. While this task shows such small improvements, our hybrid methods still achieve the best performance among comparative methods, with approximately 4.4% score gain compared to naive training.

Table 2: **Downstream performance:** Average 5-shot performance of 1B models on a wide-range of reasoning tasks. Models trained with hybrid methods achieve the best overall performance.

Method	arc easy	blimp	copa	lambada openai	lambada standard	logiqa	mc taco	mutual	open bookqa	piqa	qqp	rte	social iqa	sst2	wic	wino grande	avg —
BASELINE	43.7	82.6	64.0	25.5	22.5	22.1	63.3	61.1	16.4	62.4	57.3	51.3	36.8	50.9	48.6	51.4	47.3
DOREMI	42.4	78.9	61.0	18.1	15.0	21.8	65.8	61.2	14.2	58.3	48.6	51.3	35.8	51.0	48.8	53.2	45.3
DOGE	37.5	78.8	59.0	16.9	13.7	23.0	61.6	61.2	14.0	57.8	44.2	50.5	35.1	52.1	49.2	52.9	44.2
BIMIX	43.3	81.2	65.0	23.3	20.3	24.9	61.1	61.8	14.6	60.8	59.0	53.1	36.6	54.8	49.1	52.8	47.6
REGMIX	43.6	82.8	60.0	25.8	18.2	24.3	53.4	61.2	13.0	60.4	47.6	49.1	36.1	50.9	47.8	52.6	45.4
IMTL	43.9	83.1	64.0	26.9	22.1	24.6	64.2	62.0	16.2	60.8	52.0	53.4	37.1	51.0	49.7	51.9	47.7
MGDA	43.1	82.2	61.0	28.0	23.4	24.0	61.7	61.9	16.4	60.9	61.4	52.0	37.4	51.0	50.9	53.4	48.0
HYBRID IMTL	43.7	82.2	65.0	27.8	22.5	24.1	64.6	61.5	16.4	60.0	59.3	53.8	35.9	51.2	51.7	52.3	48.3
HYBRID MGDA	43.9	82.6	64.0	29.6	25.0	23.8	64.7	61.6	16.6	60.2	57.6	49.5	36.5	55.6	50.9	51.3	48.3
PILE-CC ONLY	46.8	83.2	71.0	27.9	24.3	23.7	63.6	61.9	16.8	65.8	48.5	53.4	37.3	52.3	49.7	51.5	48.6

Table 3: **Multi-lingual performance**: We train 155M base model on ca-Catalan, de-German, en-English, es-Spanish, fr-French, it-Italian, ja-Japanese, ko-Korean, nl-Dutch, pt-Portuguese, ru-Russian, vi-Vietnamese, zh-cn-Simplified Chinese languages.

Model	ca	de	en	es	fr	it	ja	ko	nl	pt	ru	vi	zh-cn	avg
BASELINE	3.806	5.958	6.654	5.496	5.086	6.219	8.274	5.353	5.819	6.299	5.721	5.455	8.772	5.948
DOGE	4.583	7.381	7.180	6.123	5.801	6.365	6.775	4.994	5.956	5.618	5.840	4.668	5.810	5.871
DOREMI	4.275	6.636	7.203	5.853	5.345	5.808	6.733	5.266	5.411	5.522	5.264	5.029	6.663	5.715
IMTL	4.312	6.636	6.972	5.716	5.235	5.768	6.759	5.140	5.382	5.518	5.593	5.044	6.809	5.708
MGDA	4.289	6.670	7.007	5.711	5.217	5.691	6.852	5.021	5.373	5.512	5.631	5.082	6.930	5.711
HYBRID IMTL	4.281	6.595	6.955	5.687	5.213	5.702	6.758	5.159	5.354	5.487	5.634	5.008	6.818	5.689
HYBRID MGDA	4.265	6.597	6.986	5.685	5.196	5.670	6.821	5.064	5.338	5.484	5.691	5.052	6.904	5.694

5.4 Cross-tokenizer generalization

As hypothesized by Xie et al. (2024); Albalak et al. (2023), different tokenizers may lead to different domain weights, which explains the gap between reported DoReMi results and its reimplementation score. We here conduct an experiment to quantify the robustness of the obtained mixing ratio found by 82M proxy model on The Pile dataset tokenized by GPT-2 (Radford et al., 2019) tokenizer. We retrain 1B base models from scratch on The Pile dataset but tokenized using GPT-NeoX (Black et al., 2022) tokenizer. Please note that the mixture obtained by BIMIX is from proxy models on tokenized data by GPT-NeoX, so we do not consider its performance as cross-tokenizer generalization but still include it here for benchmarking.

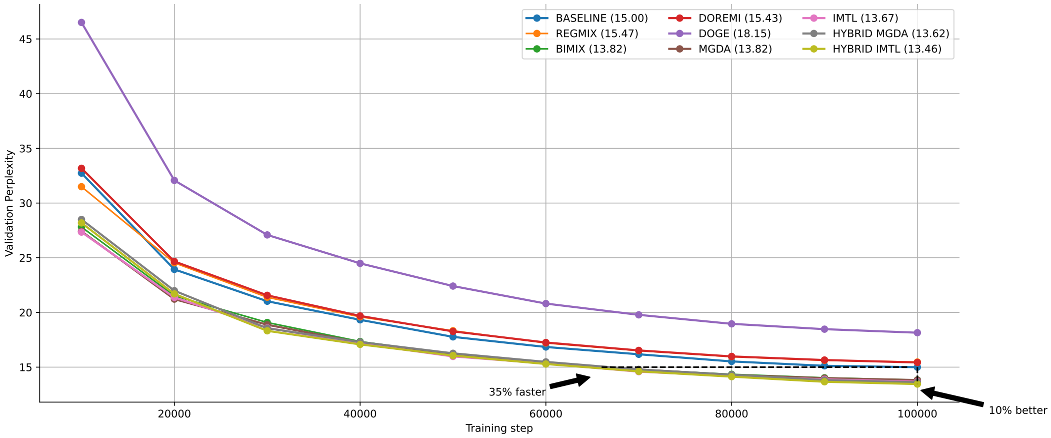


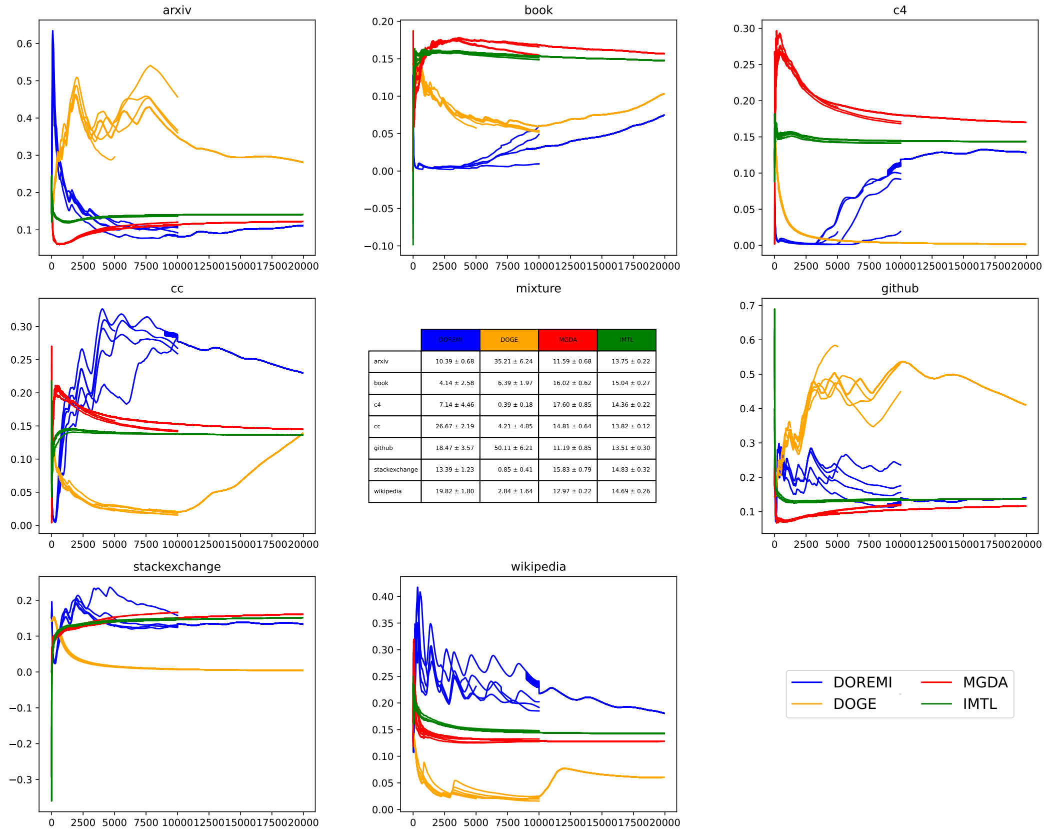
Figure 4: **Evolution of validation perplexity on The Pile in cross-tokenizer generalization**: We obtain data mixtures using 82M GPT-2 proxy models and use them to train a 1B model with GPT-NeoX tokenizer.

From Figure 4, we can observe that our methods obtain the lowest average validation perplexities among all baselines. Interestingly, while BIMIX is originally obtained from GPT-NeoX tokenizer, its performance lags behind multi-objective optimization methods. Those results show the effectiveness and robustness of the obtained mixing ratio beyond using the same tokenizer for both proxy and base models and we expect that we can train different base models using different tokenizers without retraining proxy models accordingly.

5.5 Ablation studies

Until now, we have presented the effectiveness of the proposed method on different datasets and benchmarks on different setups. In this section, we conduct further ablation studies to showcase the robustness of proposed methods and behaviors of comparative methods.

Mixing ratio across proxy training configurations We first present the evolution of the mixing ratio using different optimizers in Figure 5 where we vary proxy model size (60M, 82M and 124M) and number of training steps (5k, 10k and 20k). We choose the SlimPajama dataset in this experiment, which has 7 domains, for convenient visualization.

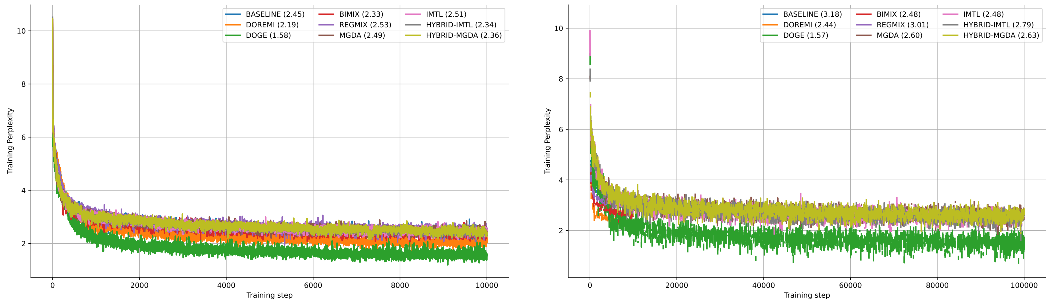


	DOREMI	DOGE	MGDA	IMTL
arxiv	10.39 ± 0.68	35.21 ± 6.24	11.59 ± 0.68	13.75 ± 0.22
book	4.14 ± 2.58	6.39 ± 1.97	16.02 ± 0.62	15.04 ± 0.27
c4	7.14 ± 4.46	0.39 ± 0.18	17.60 ± 0.85	14.36 ± 0.22
cc	26.67 ± 2.19	4.21 ± 4.85	14.81 ± 0.64	13.82 ± 0.12
github	18.47 ± 3.57	50.11 ± 6.21	11.19 ± 0.85	13.51 ± 0.30
stackexchange	13.39 ± 1.23	0.85 ± 0.41	15.83 ± 0.79	14.83 ± 0.32
wikipedia	19.82 ± 1.80	2.84 ± 1.64	12.97 ± 0.22	14.69 ± 0.26

Figure 5: **Ablation on different configurations for training proxy models**: SlimPajama mixture obtained by baselines on different hyperparameter configurations. Mixing ratios are consistent across model sizes and vary across different methods, which motivate us to use the loss reweighting ratio of small models to guide the training of large models.

Overall, the mixing ratios obtained by the same optimizer are relatively consistent across training configurations. Even though, DOGE and DoReMi show the fluctuating trend during training while MGDA and IMTL show their great stability. Particularly, after the first 1k iterations, the obtained mixing ratios almost remain till the end of training.

Training loss In the toy example, we show that Baseline, DoReMi and DOGE are easy to overfit to the easy objective. In Figure 6, those two methods also obtain the lowest training losses as they sample a lot of examples from easy domains, e.g. Github in this case. However, these low training loss values do not translate well to validation performance and their performance, in fact, lags behind other baselines.



(a) The Pile

(b) SlimPajama

Figure 6: **Training loss**, While the original mixture often obtains highest training loss among baselines, DoReMi and DOGE both put a lot of their mixture proportion on easy domains and obtain low training loss.

Performance of proxy models While correcting the mismatch between the training of proxy and base models could help improve the performance of base models, those multi-objective optimizers are better than conventional training methods.

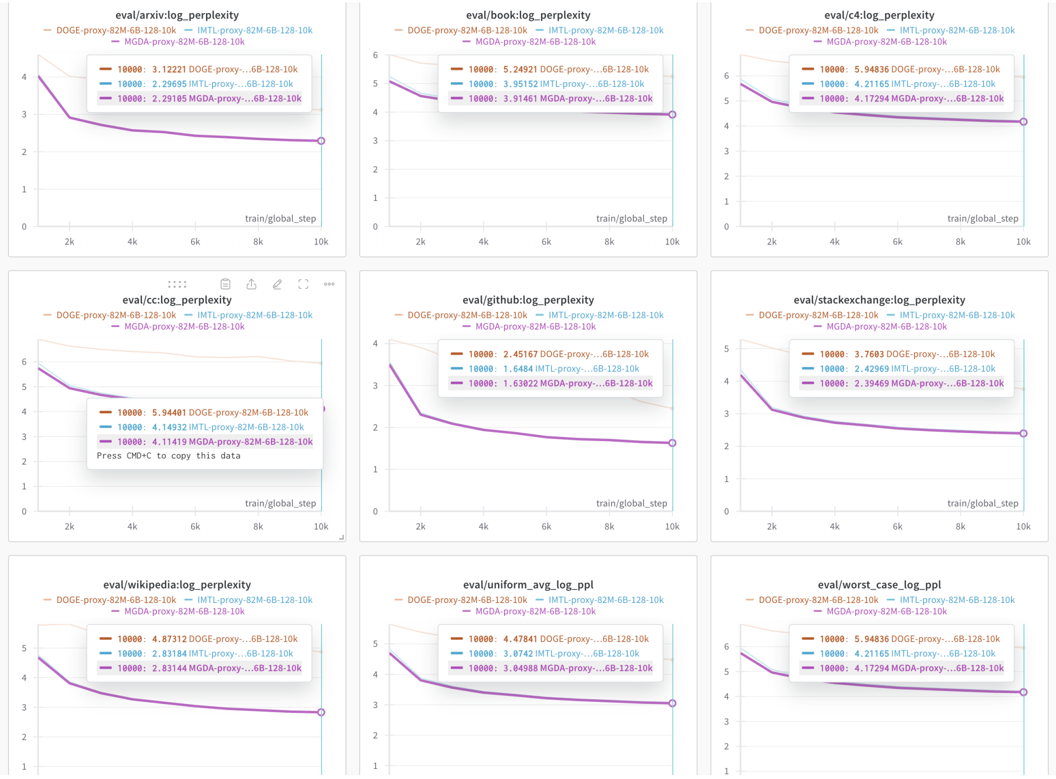

eval/arxiv:log_perplexity
DOGE-proxy-82M-6B-128-10k
IMTL-proxy-82M-6B-128-10k
MGDA-proxy-82M-6B-128-10k
10000: 3.12221 DOGE-proxy-...6B-128-10k
10000: 2.29695 IMTL-proxy-...6B-128-10k
10000: 2.29105 MGDA-proxy-...6B-128-10k
train/global_step
2k
4k
6k
8k
10k
eval/book:log_perplexity
DOGE-proxy-82M-6B-128-10k
IMTL-proxy-82M-6B-128-10k
MGDA-proxy-82M-6B-128-10k
10000: 5.24921 DOGE-proxy-...6B-128-10k
10000: 3.95152 IMTL-proxy-...6B-128-10k
10000: 3.91461 MGDA-proxy-...6B-128-10k
train/global_step
eval/c4:log_perplexity
DOGE-proxy-82M-6B-128-10k
IMTL-proxy-82M-6B-128-10k
MGDA-proxy-82M-6B-128-10k
10000: 5.94836 DOGE-proxy-...6B-128-10k
10000: 4.21165 IMTL-proxy-...6B-128-10k
10000: 4.17294 MGDA-proxy-...6B-128-10k
train/global_step
eval/cc:log_perplexity
DOGE-proxy-82M-6B-128-10k
IMTL-proxy-82M-6B-128-10k
MGDA-proxy-82M-6B-128-10k
10000: 5.94401 DOGE-proxy-82M-6B-128-10k
10000: 4.14932 IMTL-proxy-82M-6B-128-10k
10000: 4.11419 MGDA-proxy-82M-6B-128-10k
Press CMD+C to copy this data
train/global_step
eval/github:log_perplexity
DOGE-proxy-82M-6B-128-10k
IMTL-proxy-82M-6B-128-10k
MGDA-proxy-82M-6B-128-10k
10000: 2.45167 DOGE-proxy-...6B-128-10k
10000: 1.6484 IMTL-proxy-...6B-128-10k
10000: 1.63022 MGDA-proxy-...6B-128-10k
train/global_step
eval/stackexchange:log_perplexity
DOGE-proxy-82M-6B-128-10k
IMTL-proxy-82M-6B-128-10k
MGDA-proxy-82M-6B-128-10k
10000: 3.7603 DOGE-proxy-...6B-128-10k
10000: 2.42969 IMTL-proxy-...6B-128-10k
10000: 2.39469 MGDA-proxy-...6B-128-10k
train/global_step
eval/wikipedia:log_perplexity
DOGE-proxy-82M-6B-128-10k
IMTL-proxy-82M-6B-128-10k
MGDA-proxy-82M-6B-128-10k
10000: 4.87312 DOGE-proxy-...6B-128-10k
10000: 2.83184 IMTL-proxy-...6B-128-10k
10000: 2.83144 MGDA-proxy-...6B-128-10k
eval/uniform_avg_log_ppl
DOGE-proxy-82M-6B-128-10k
IMTL-proxy-82M-6B-128-10k
MGDA-proxy-82M-6B-128-10k
10000: 4.47841 DOGE-proxy-...6B-128-10k
10000: 3.0742 IMTL-proxy-...6B-128-10k
10000: 3.04988 MGDA-proxy-...6B-128-10k
eval/worst_case_log_ppl
DOGE-proxy-82M-6B-128-10k
IMTL-proxy-82M-6B-128-10k
MGDA-proxy-82M-6B-128-10k
10000: 5.94836 DOGE-proxy-...6B-128-10k
10000: 4.21165 IMTL-proxy-...6B-128-10k
10000: 4.17294 MGDA-proxy-...6B-128-10k

Figure 7: **Proxy validation perplexity**: Multi-objective optimization methods could help optimize domain losses better consistently across domains.

Figure 7 displays the validation perplexities across 7 domains of the SlimPajama dataset, where MGDA and IMTL obtain low validation loss while DOGE fails to achieve the level of aforementioned optimizers. We conjecture this is due to the fact that DOGE uses a regularization term via Bregman divergence (Fan et al., 2024) to promote the stability of the mixing ratio by penalizing the change between two consecutive steps. However, as shown in Figure 2, its obtained mixing ratio still fluctuates through time.

6 Conclusion

In this paper, we propose to formulate the data mixture problem as a multi-objective optimization, which allow us to leverage existing theoretical grounded results from multi-objective optimization theory. Base on this new formulation, we propose to correct the mismatch between the training of the base and proxy models of prior methods. Empirically, we showcase the effectiveness of proposed methods on various benchmarks and setups.

References

Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. Efficient online data mixing for language model pre-training. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. Gpt-neox-20b: An open-source autoregressive language model. *Challenges & Perspectives in Creating Large Language Models*, pp. 95, 2022.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

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

Together Computer. Redpajama: An open source recipe to reproduce llama training dataset, April 2023. URL `https://github.com/togethercomputer/RedPajama-Data`.

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. Mutual: A dataset for multi-turn dialogue reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1406–1416, 2020.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pp. 5547–5569. PMLR, 2022.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. Doge: Domain reweighting with generalization estimation. In *Forty-first International Conference on Machine Learning*, 2024.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024. URL `https://zenodo.org/records/12608602`.

Ce Ge, Zhijian Ma, Daoyuan Chen, Yaliang Li, and Bolin Ding. Data mixing made efficient: A bivariate scaling law for language model pretraining. *arXiv preprint arXiv:2405.14908*, 2024.

Sachin Goyal, Pratyush Maini, Zachary C Lipton, Aditi Raghunathan, and J Zico Kolter. Scaling laws for data filtering–data curation cannot be compute agnostic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22702–22711, 2024.

Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. Wiki-40b: Multilingual language model dataset. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 2440–2452, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 30016–30030, 2022.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: a challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 3622–3628, 2021.

Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*.

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, 2018.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Rohan Pandey. gzip predicts data-dependent scaling laws. *arXiv preprint arXiv:2405.16684*, 2024.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, 2016.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. 2019.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2021.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, June 2023. URL https://huggingface.co/datasets/cerebras/SlimPajama-627B.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3614–3633, 2021.

Ashish Vaswani. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36, 2024.

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

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33: 5824–5836, 2020.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3363–3369, 2019.

Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.

A Appendix

Due to space constraints, some details were omitted from the main paper. We therefore include the detailed experimental setup (Appendix A.1) and additional results (Appendix A.2) in this section.

A.1 Training details

A.1.1 Toy example

For the ZDT-2 problem, we randomly sample 100 samples $\mathbf{x}_i \sim \text{Uniform}(0,1)^{30}$ for $i = 1, \ldots, 100$. And update them using Adam (Kingma, 2014) optimizer with the learning rate of 0.001 for 1000 iterations without any scheduler.

A.1.2 Language model training

Following (Fan et al., 2024), we train all models, except 1B ones, with the maximal and minimum learning rates are 5×10^{-4} and 1×10^{-4}, respectively. These numbers for 1B model are 1.5×10^{-4} and 2×10^{-5}. The weight decay for all models is set as 0.01, the gradient clip is set as 1.0. All models are trained with AdamW (Loshchilov & Hutter, 2017) with the default number of training iterations is 100k.

Table 4: The detailed model configuration for different model sizes.

Experiment	**The Pile & SlimPajama**							**Multi-lingual**		**NeoX**
Vocabulary Size	50304	50304	50304	50304	50304	50304	50304	100000	100000	50277
n_{layers}	2	3	6	12	24	16	16	4	16	16
n_{heads}	8	6	12	12	16	16	32	4	8	32
$d_{\text{embedding}}$	256	768	768	768	768	1600	2048	256	512	2048
Num Parameters	27M	99M	120M	163M	248M	653M	1B	55M	155M	1B

A.2 Additional results

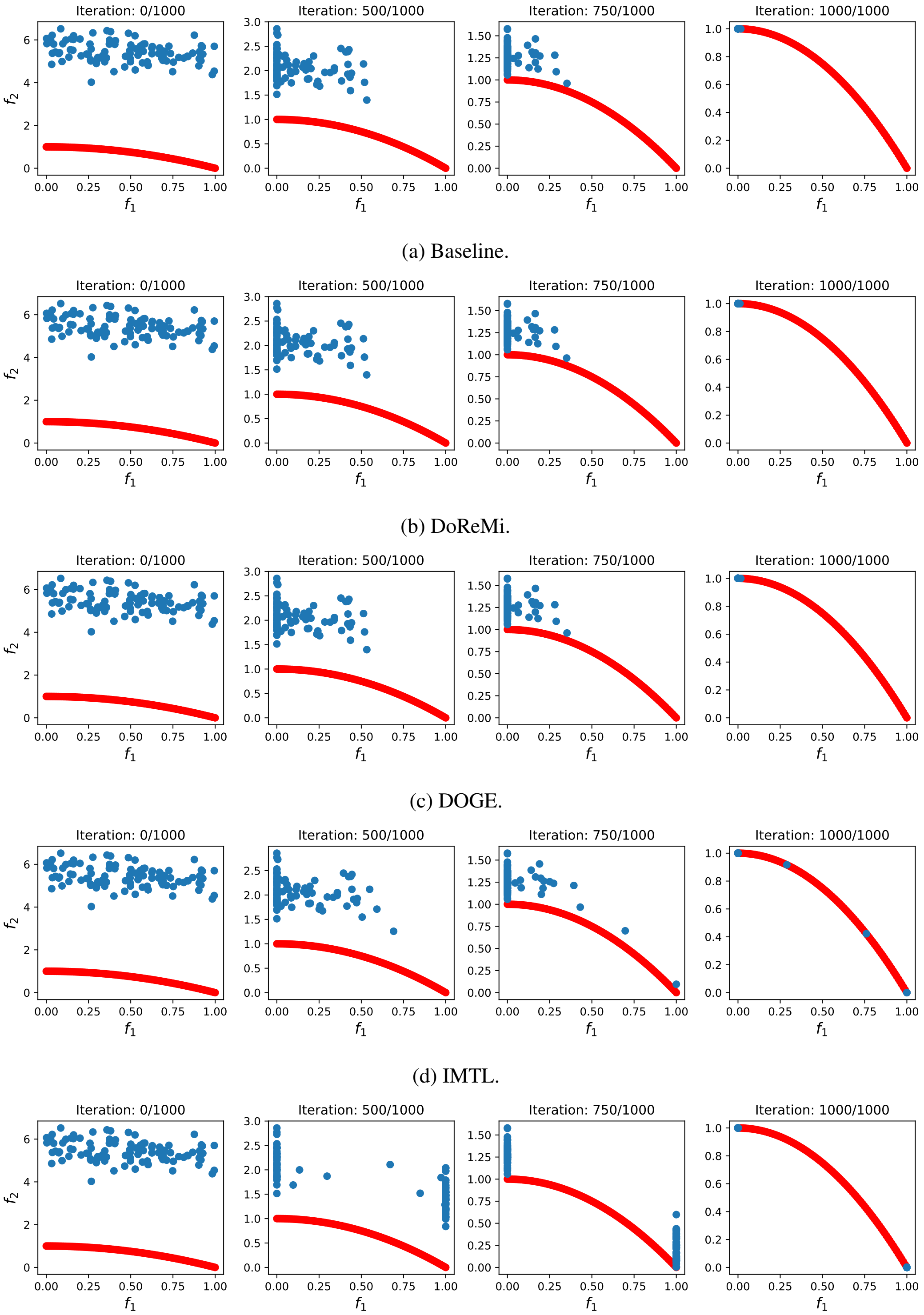
A.2.1 Toy example

A.2.2 Ablation on different base model scales

Table 5: **Average validation perplexity across domains**: We calculate the unweighed average validation perplexity of 1B, 653M and 248M models across 22 domains of The Pile.

								HYBRID	**HYBRID**
Model size	**BASELINE**	**DOREMI**	**DOGE**	**REGMIX**	**BIMIX**	**MGDA**	**IMTL**	**MGDA**	**IMTL**
1B	12.71	13.03	15.36	13.00	11.69	11.75	11.66	11.42	11.32
653M									
248M									

Here's the data converted to a LaTeX table:



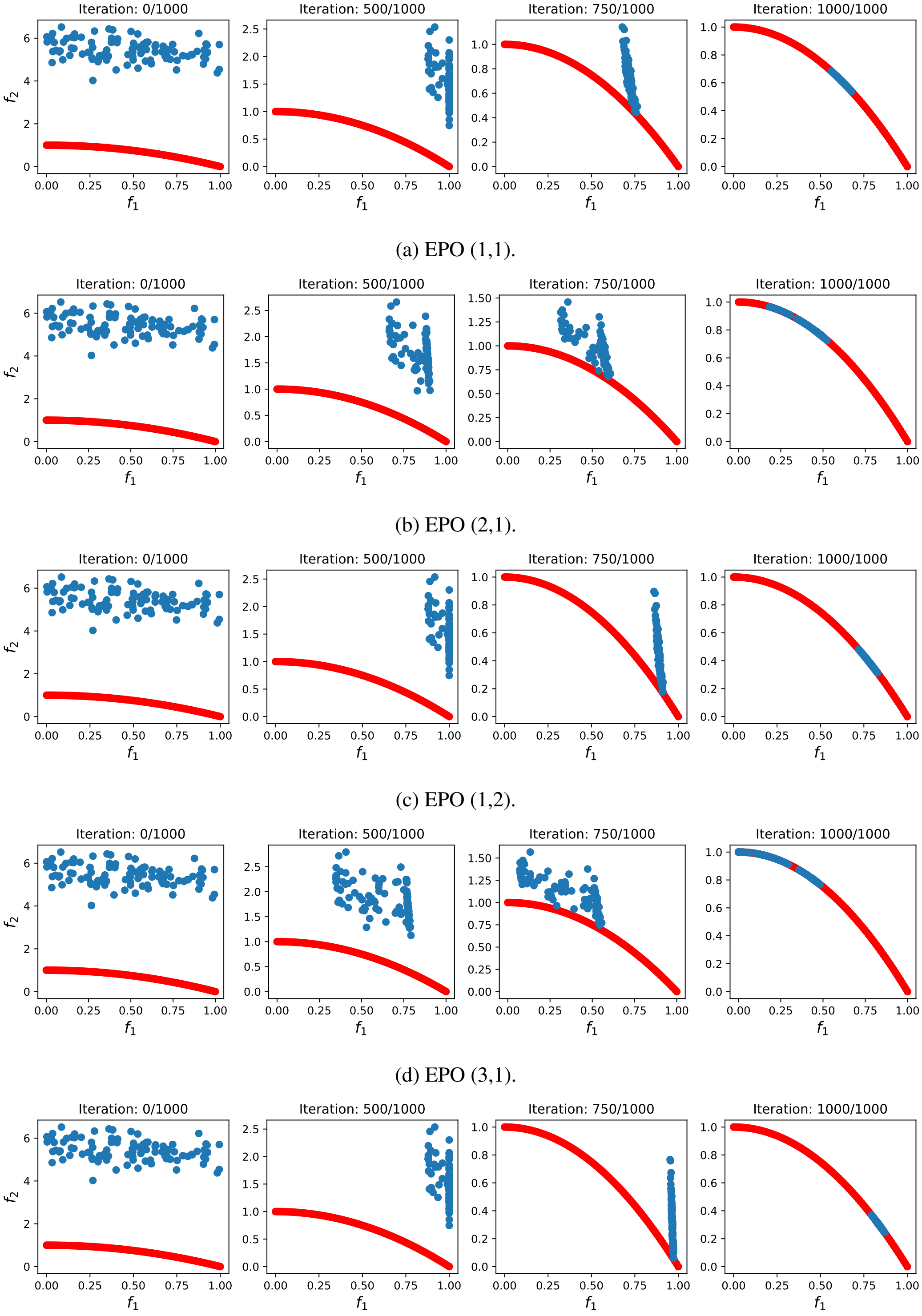
(a) Baseline.

(b) DoReMi.

(c) DOGE.

(d) IMTL.

(e) MGDA.



(a) EPO (1,1).

(b) EPO (2,1).

(c) EPO (1,2).

(d) EPO (3,1).

(e) EPO (1,3).

Table 6: Domain weights on The Pile

Dataset	Baseline	REGMIX	BIMIX	DoReMi	DOGE	IMTL	MGDA
Pile-CC	0.1121	0.03805	0.0507	0.05242	0.00619	0.0404	0.04245
PubMed Central	0.1071	0.00355	0.0088	0.00056	0.02525	0.05073	0.05923
Books3	0.0676	0.02321	0.0392	0.00285	0.01189	0.04591	0.04911
OpenWebText2	0.1247	0.01065	0.024	0.01156	0.00759	0.04034	0.04124
ArXiv	0.1052	0.00034	0.0786	0.06881	0.09824	0.04351	0.03291
Github	0.0427	0.00648	0.0648	0.06346	0.60794	0.03748	0.02414
FreeLaw	0.0386	0.13647	0.0441	0.0186	0.01151	0.0444	0.04166
StackExchange	0.0929	0.062	0.0812	0.27021	0.00559	0.04144	0.03504
USPTO Backgrounds	0.042	0.00995	0.0373	0.00175	0.0148	0.03923	0.0332
PubMed Abstracts	0.0845	0.01019	0.0228	0.00674	0.00117	0.04154	0.04182
Gutenberg (PG-19)	0.0199	0.00032	0.0155	0.02045	0.01253	0.04508	0.04657
OpenSubtitles	0.0124	0.00413	0.0187	0.00112	0.0028	0.05681	0.06785
Wikipedia (en)	0.0919	0.22724	0.0443	0.00287	0.01833	0.03949	0.0351
DM Mathematics	0.0198	0.00312	0.0372	0.03637	0.04542	0.05644	0.0617
Ubuntu IRC	0.0074	0.18762	0.0423	3.83E-05	0.02128	0.06164	0.06164
BookCorpus2	0.0044	0.05623	0.0727	0.16007	0.00587	0.05047	0.05785
EuroParl	0.0043	0.17261	0.0464	0.01647	0.00488	0.04562	0.03424
HackerNews	0.0075	0.01685	0.0616	0.04021	0.00146	0.04433	0.05238
YoutubeSubtitles	0.0042	0.00722	0.0387	0.04592	0.03757	0.0418	0.03108
PhilPapers	0.0027	0.00038	0.0435	0.0474	0.01856	0.04211	0.0331
NIH ExPorter	0.0052	0.02194	0.0789	0.11216	0.00181	0.04286	0.04381
Enron Emails	0.003	0.00143	0.0486	0.01995	0.05443	0.05443	0.07382