DRPruning: Efficient Large Language Model Pruning through Distributionally Robust Optimization

Hexuan Deng[1] **Wenxiang Jiao** **Xuebo Liu**[1*] **Jing Li**[1] **Min Zhang**[1] **Zhaopeng Tu**
[1]Institute of Computing and Intelligence, Harbin Institute of Technology, Shenzhen, China
{hxuandeng,wenxiangjiaonju,tuzhaopeng}@gmail.com,
{liuxuebo,li.jing,zhangmin2021}@hit.edu.cn

Abstract

Large language models (LLMs) deliver impressive results but face challenges from increasing model sizes and computational costs. Structured pruning reduces model size and speeds up inference but often causes uneven degradation across domains, leading to biased performance. To address this, we propose *DRPruning*, a method that dynamically adjusts the data distribution during training to restore balanced performance across heterogeneous and multi-tasking data. Experiments in monolingual and multilingual settings show that DRPruning surpasses similarly sized models in both pruning and continued pretraining over perplexity, downstream tasks, and instruction tuning. Further analysis demonstrates the robustness of DRPruning towards various domains and distribution shifts. Furthermore, DRPruning can determine optimal reference losses and data ratios automatically, suggesting potential for broader applications. Code and scripts are available at https://github.com/hexuandeng/DRPruning.

1 Introduction

Large language models (LLMs) have advanced rapidly, achieving impressive results across a wide range of tasks (Jiao et al., 2023b; Frieder et al., 2023; Deng et al., 2023; Bian et al., 2024; Rao et al., 2024). However, this progress has come with increasing model sizes, significantly raising computational costs for both training and inference, which impacts their accessibility. Structured pruning is a promising approach to reduce model size (Han et al., 2015; Wen et al., 2016), but it often causes uneven performance degradation across domains, leading to biased capabilities and unfair downstream task performance (Xia et al., 2024).

Given that LLMs inherently handle heterogeneous, multi-domain data, distribution robustness becomes essential. A commonly used approach is distributionally robust optimization (DRO; Oren et al., 2019; Sagawa et al., 2019), which aims to optimize worst-case performance across distributions. A reference loss is defined for each domain as a target. Domains with larger deviations from this reference loss are assigned higher weights, while not straying too far from a predefined reference data ratio. However, setting these hyperparameters is challenging, and suboptimal configurations often result in poor outcomes (Zhou et al., 2021).

To address this, we propose DRPruning, a distributionally robust pruning method that incorporates DRO to dynamically adjust the data distribution during training. Further, using scaling laws (Kaplan et al., 2020; Ghorbani et al., 2022), we predict the loss after training as the reference loss, where larger deviations indicate poorer performance, thereby promoting capability recovery in these areas. Additionally, we gradually increase the reference data ratio for domains with greater deviations, ensuring robustness across a wider range of distributions, particularly more challenging ones.

DRPruning is validated through experiments in monolingual and multilingual settings, which represent varying degrees of distributional shift. DRPruning outperforms other data scheduling methods in both pruning and continued pretraining, as measured by perplexity (-5.59%), downstream tasks (+1.52%), and instruction tuning (55.4% win rate). Particularly in multilingual settings, DRPruning achieves +2.95% in downstream tasks. To further assess domain-specific performance, we develop a sentence continuation benchmark using existing unlabeled data, demonstrating our improved domain-level capabilities (+17.9%).

Our contributions are summarized as follows:

- DRPruning tackles domain imbalance in structured pruning by introducing a distributionally robust pruning method, that dynamically ad-

* Corresponding Author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 29152–29173
July 27 - August 1, 2025

justs data ratios during training to ensure robustness against distributional shifts.

- We validate DRPruning through extensive experiments in monolingual and multilingual settings. Further analysis confirms its advantages in handling data heterogeneity and distribution shifts.
- DRPruning offers refined reference losses and data ratios, which can be applied more broadly to enhance various model training processes and contribute to advancements for LLMs.

2 Background

2.1 Structured Pruning

To prune the model to any target configuration, we adopt structured pruning based on Sheared Llama (Xia et al., 2024). For each granularity i, pruning masks $Z = \{\mathbf{z}^i \mid \mathbf{z}^i \in \mathbb{R}^{D_i}\}$ are learned to determine whether substructures are pruned or retained, where $z_j^i = 0$ indicates pruning of the j-th substructure. Pruning is applied at various granularities, including transformer layers, hidden dimensions, attention heads, and FFN intermediate dimensions.

To parameterize the masks, the ℓ_0 regularization method (Louizos et al., 2018) with hard concrete distributions is used to concentrate probability mass at 0 or 1. Lagrange multipliers are then used to ensure the pruned model meets the target configuration. Specifically, if exactly t^i parameters must be retained for $\mathbf{z}^i$, the following constraint is imposed:

$$\tilde{\ell}^i = \lambda^i \left(\sum_j z_j^i - t^i \right) + \phi^i \left(\sum_j z_j^i - t^i \right)^2 . \tag{1}$$

The final training loss integrates these constraints with the language modeling loss of the pruned model, jointly optimizing the model parameters θ and pruning masks $\mathbf{z}$, with $\mathbf{z}$ typically uses a higher learning rate. After pruning, the highest-scoring components are retained.

2.2 Distributionally Robust Optimization

To mitigate uneven domain performance after pruning, we apply distributionally robust optimization (DRO; Oren et al., 2019; Sagawa et al., 2019) to improve the model's robustness to distribution shifts. DRO seeks a model θ that performs well across a set of potential test distributions $\mathcal{Q}$ over n domains:

$$\underset{\theta}{\text{minimize}} \sup_{Q \in \mathcal{Q}} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim Q}[\ell(\mathbf{x}, \mathbf{y}; \theta)]. \tag{2}$$

To solve the min-max optimization, the iterative best response algorithm (Fudenberg and Levine, 1998) is used. Each iteration consists of first performing the empirical risk minimization on the current data distribution $\mathbf{q}^t$, followed by updating the data distribution using worst-case weights based on the current parameters. Formally,

$$\begin{aligned} \theta^{t+1} &\leftarrow \underset{\theta}{\operatorname{argmin}} \sum_i q_i^t \ell\left(\theta; D_i\right), \\ \mathbf{q}^{t+1} &\leftarrow \underset{\mathbf{q}=\{q_1,\ldots,q_n\}\in\mathcal{Q}}{\operatorname{argmax}} \sum_i q_i \ell\left(\theta^{t+1}; D_i\right). \end{aligned} \tag{3}$$

3 Our Proposed DRPruning Method

To address the challenges of LLMs in handling heterogeneous and multi-tasking data, we propose DRPruning, illustrated in Figure 1. Each evaluation phase is treated as an iteration: the evaluation loss first updates the **reference loss**, which, along with the previous **reference data ratio**, serves as input to the DRO process. This yields a new data proportion for the next training step, and the reference data ratio is updated accordingly.

3.1 Distributionally Robust Pruning

We first introduce the overall procedure by applying naïve DRO to design an effective pruning and continued pretraining method. Specifically, we adopt common techniques (Ma et al., 2023; Li et al., 2024), first applying structured pruning to reduce model parameters, followed by continued pretraining to restore capabilities. Compared to training from scratch, this approach requires fewer unlabeled data to restore the model's performance (Zhang et al., 2024b).

Integrate DRO into pruning and continued pretraining. During training, we use DRO to dynamically adjust the data ratio to improve the model's robustness and convergence speed. Specifically, to prevent overfitting, we compile a validation set and use the evaluation loss as the loss score. After each evaluation, we update the data ratio based on the evaluation loss using the DRO method, as in Eqn. 3, guiding training to focus more on underperforming domains.

Further improvement. Next, we optimize the loss function ℓ (Section 3.2) and potential distributions $\mathcal{Q}$ (Section 3.3) to ensure robust training, as shown in Figure 1. In contrast, Sheared Llama employs a dynamic scheduling strategy that forces

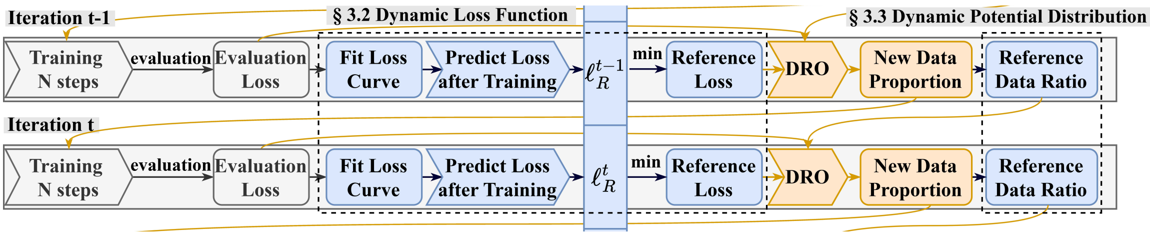

Figure 1: Data proportion update procedure for DRPruning. The gray part represents the standard training process, the yellow part represents the normal process for DRO, and the blue part represents our newly added module.

the model to strictly adhere to the relative loss magnitudes of larger LLMs, without placing any constraints on the potential distributions. This leads to suboptimal results, particularly in multilingual settings with significant distribution shifts.

3.2 Dynamic Loss Function

To stabilize DRO training and prevent domains with slow convergence from disproportionately influencing the weights, the use of a ***reference loss*** ℓ_R is a common approach (Oren et al., 2019; Zhou et al., 2021). This reference loss establishes the minimum acceptable performance for a domain. Furthermore, we update the loss score as $\ell(\theta; D) \leftarrow \ell(\theta; D) - \ell_R$. Proper tuning of ℓ_R can significantly improve performance (Jiao et al., 2022). However, determining an appropriate value remains a challenging task.

Minimum performance estimation. To address this, we predict the model's loss at the end of training as an estimate of the minimum acceptable performance. Specifically, we leverage scaling laws to capture training dynamics and forecast the loss based on evaluation loss trends (Kaplan et al., 2020; Zhang et al., 2024a). Given the number of parameters P and the current training step T, the predicted training loss is estimated by:

$$\hat{\ell}(P, T) = A \cdot \frac{1}{P^\alpha} \cdot \frac{1}{T^\beta} + E, \tag{4}$$

where A, E, α, and β are trainable parameters. For each domain, after each evaluation, we collect a data point, refit the curve to all collected points, and use the predicted curve to estimate the loss at the end of training as the predicted minimum performance. Following Hoffmann et al. (2022a), we estimate using the Huber loss ($\delta = 0.001$) and the L-BFGS algorithm, and select the average of the best-fitting three from a grid of initializations. To ensure sufficient data points, we start predictions only after 20% of training is complete.

Reference loss adjustment. Subsequently, we set the reference loss using the predicted minimum performance. In our preliminary experiments, this approach exhibits strong numerical stability. To accelerate convergence, we adopt the minimum value as the reference loss. This dynamically evaluates domains with poorer performance, allowing DRO to assign higher weights to these domains, thereby promoting faster model convergence.

3.3 Dynamic Potential Distribution

Sagawa et al. (2019) consider robustness to arbitrary subpopulations, which is overly conservative and degenerates into training only on the highest-loss domain. To address this issue, Zhou et al. (2021) propose a more reasonable assumption by restricting $\mathcal{Q}$ in Eqn. 2 to an f-divergence ball (Csiszár, 1967) around a ***reference data ratio*** $\mathbf{p}_R$. This yields promising results, better ensuring domain balance (Jiao et al., 2022). Formally,

$$\mathcal{Q} = \left\{ \mathbf{q} : \chi^2\left(\mathbf{q}, \mathbf{p}_R\right) \leq \rho \right\}. \tag{5}$$

However, this assumption can be too restrictive, necessitating a carefully chosen reference data ratio $\mathbf{p}_R$. An unreasonable choice may reduce the model's robustness to distributional shifts.

Reference data ratio adjustment. To address this, we propose a method that combines the strengths of the aforementioned approaches. We still employ Eqn. 5 to constrain the distribution within a limited range, while gradually shifting the reference data ratio towards domains with higher losses to improve the model's robustness to more challenging distributions. To ensure adequate training across all traversed potential distributions, we gradually update the reference ratio.

Compared to existing reference ratios, the DRO method dynamically assigns higher weights $\mathbf{q}$ to domains with higher losses. This method shows good numerical stability, which we leverage to update the reference ratio. Formally, we update:

$$\mathbf{p}_R^{t+1} = \delta \cdot \mathbf{q}^t + (1 - \delta) \cdot \mathbf{p}_R^t. \tag{6}$$

Finally, to prevent the method from degenerating into training solely on the highest-loss domain, we constrain the reference ratio of each domain to lie between $\frac{1}{n}$ and n times the initial ratio. Formally, we set $\frac{1}{n} \cdot \mathbf{p}_R^0 \leq \mathbf{p}_R^t \leq n \cdot \mathbf{p}_R^0$. We apply this method after 40% of the training is completed, ensuring the model sufficiently converges near the initial reference ratio and that the reference loss stabilizes.

4 Experiments

4.1 Experimental Setup

Model. Llama2-7B model (Touvron et al., 2023b) is used as the base model. We employ the same target architecture as Sheared Llama for structured pruning to ensure a fair comparison. We compare our method, i.e., **DRPruning**, to strong open-source models of similar sizes, including **Pythia**-1.4B and 2.8B (Biderman et al., 2023) and **Sheared Llama**-1.3B and 2.7B. Additionally, we reproduce Sheared Llama, using the same data settings to control for other variables (**ReSheared**). Further details are provided in Appendix A.1.

Data. To ensure comparability with Sheared Llama, we align most of our settings with its approach. However, due to insufficient documentation of its data filtering method, we are unable to replicate the results under the 2.7B setting. Therefore, our comparison primarily focuses on ReSheared and DRPruning, using a similar data setting for our reproduction. We allocate 0.4 billion tokens for pruning, utilizing the publicly available pruning dataset of Sheared Llama. We employ 50 billion tokens for continued pretraining, and use SlimPajama (Shen et al., 2023), a filtered version of RedPajama (Computer, 2023), and use its training split for continued pretraining.

Downstream task evaluation. We use the lm-evaluation-harness package (Gao et al., 2024) to evaluate on an extensive suite of downstream tasks:

- We follow Llama2 to report the 0-shot performance on PIQA (Bisk et al., 2020), WinoGrande (WinoG, Sakaguchi et al., 2020), ARC Easy (ARCE, Clark et al., 2018), SQuAD (Rajpurkar et al., 2018), BoolQ (Clark et al., 2019), TruthfulQA (TruthQA, Lin et al., 2022), and 5-shot performance on Natural Questions (NQ, Kwiatkowski et al., 2019) and TriviaQA (TriQA, Joshi et al., 2017).

- We follow Pythia to report the 0-shot performance of LAMBADA (LAMB, Paperno et al., 2016), LogiQA (Liu et al., 2020), SciQ (Welbl et al., 2017), and WSC (Kocijan et al., 2020).

- We follow Sheared Llama to report performance of the tasks used by Open LLM Leaderboard, including 10-shot HellaSwag (HelS, Zellers et al., 2019), 25-shot ARC Challenge (ARCC, Clark et al., 2018), and 5-shot MMLU (Hendrycks et al., 2021).

Instruction tuning evaluation. To further explore the potential applications of the base model, we follow Sheared Llama by training with 10k instruction-response pairs sampled from the ShareGPT dataset and using another 1k instructions for evaluation. We follow Wang et al. (2024) and Sheared Llama to employ LLMs, specifically GPT-4o, as an evaluator to compare the responses of the two models and report the win rates.

Method	From	To	PPL ↓	Task ↑
Sheared Llama	7B	1.3B	10.05	34.89
ReSheared	7B	1.3B	10.42	34.85
DRPruning	7B	1.3B	**9.83**	**35.60**
Sheared Llama	7B	2.7B	7.64	39.75
ReSheared	7B	2.7B	7.83	39.98
DRPruning	7B	2.7B	**7.40**	**40.18**

Table 1: Perplexity (PPL) and downstream task performance (Task) of pruned models. The "Task" performance represents the macro-average across 15 tasks, as detailed in Section 4.1. "From" and "To" indicate the model size before and after pruning, respectively.

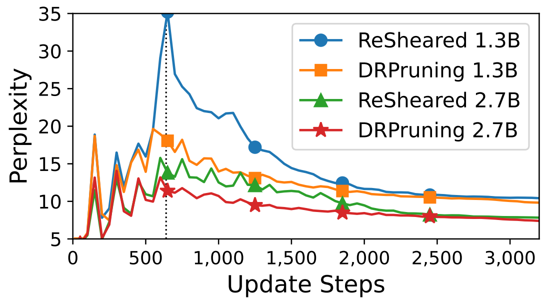

Figure 2: The curve of PPL changes during pruning from 7B. Over the first 640 iterations (the vertical dash line), the model size is gradually reduced from 7B to the target size, which causes an initial increase in PPL.

Tasks	**7B**	**2.7B**				**1.3B**			
	Llama2†	**Pythia**†	**Sheared**†	**ReSheared**	**DRPrun.**	**Pythia**†	**Sheared**†	**ReSheared**	**DRPrun.**
WSC	36.54	38.46	**48.08**	36.54	46.15	36.54	36.54	40.38	**50.00**
TriQA (5)	64.16	27.17	42.92	40.14	**43.33**	18.19	26.03	24.98	**28.10**
NQ (5)	25.98	7.12	14.85	13.49	**15.82**	4.79	8.75	8.39	**10.44**
TruthQA	32.09	28.79	**30.21**	28.41	30.13	**30.75**	29.12	28.09	29.68
LogiQA	30.11	28.11	28.26	26.27	**28.73**	27.50	27.50	28.11	**28.88**
BoolQ	77.71	64.50	**65.99**	64.92	65.08	63.30	62.05	61.01	**63.36**
LAMB	73.90	64.76	**68.21**	66.18	66.91	**61.67**	61.09	58.84	60.28
MMLU (5)	44.18	**27.09**	26.63	25.70	26.99	26.75	25.70	26.60	**27.28**
SciQ	94.00	88.50	**91.10**	90.10	89.80	86.70	87.00	86.40	**87.70**
ARCE	76.35	64.27	67.34	**67.72**	67.13	60.40	**60.90**	60.35	**60.90**
ARCC (25)	52.65	36.35	**42.66**	40.10	40.53	33.02	33.96	**34.30**	33.62
PIQA	78.07	73.88	76.12	**76.71**	75.19	70.84	73.50	**74.59**	72.69
WinoG	69.06	59.83	**65.04**	63.38	64.72	57.38	57.85	**60.06**	58.01
SQuAD	40.02	26.81	**49.26**	49.17	44.69	22.66	29.57	**37.59**	35.06
HelS (10)	78.95	60.81	71.24	**72.03**	69.22	53.49	61.05	**63.06**	58.88
Average	58.25	46.43	**52.53**	50.72	51.63	43.60	45.37	46.18	**46.99**

Table 2: Performances of different models across 15 downstream tasks. "Sheared" refers to Sheared Llama. "ReSheared" is our reproduction of Sheared Llama. "DRPrun." refers to our method. The number of shots is indicated in parentheses, with 0-shot used when unspecified. A model marked with † indicates training on different data. **Bold** and underlined represent the best and second-best results, respectively, for each model size.

4.2 Main Results for Pruning

Our method surpasses ReSheared during pruning in PPL and downstream tasks.

DRPruning promotes convergence by increasing the weight of underperformed domains. To demonstrate this, we record the average PPL across different domains on the validation set. Table 1 shows that our method achieves lower PPL. Figure 2 confirms faster convergence as pruning proceeds, with further potential in later training stages, suggesting additional gains with extended training.

Our method better preserves the original model's performance during pruning. For a comprehensive analysis, we evaluate the model's downstream task performance post-pruning. Across 15 tasks, our method significantly improves performance on average, with +1.53% over the open-source model and +1.27% in a fair comparison.

The pruning similarity analysis is detailed in Appendix B.1, and Appendix B.2 shows that pruning larger LLMs offers no advantage.

4.3 Main Results for Continued Pretraining

DRPruning outperforms ReSheared on average across LM benchmarks and instruction tuning.

LLMs recovered by our method are better foundation models after continued pretraining. Table 2 presents the downstream performance of models with similar sizes, showing that our model surpasses most open-source models. We are unable to replicate the results of the 2.7B Sheared Llama due to differences in data, where our approach yields worse performance. Nevertheless, we outperform other open-source LLMs. On average, we achieve an improvement of +4.95% comparing to open-source LLMs. Additionally, under consistent experimental conditions, our method outperforms ReSheared, achieving +1.78% on average. Its statistical significance is confirmed by t-test in Appendix B.3. Moreover, we achieve significantly lower perplexity: 5.61 vs 5.97 for 1.3B, and 5.00 vs 5.27 for 2.7B, averaging a -5.60% reduction.

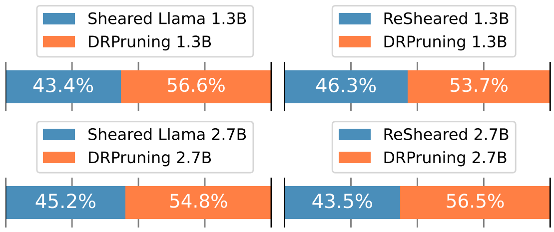

Figure 3: Win rate during instruction tuning. DRPruning outperforms Sheared Llama and ReSheared.

The effectiveness of DRPruning is further demonstrated by instruction tuning. We compare the win rates of our instruction-tuned model with Sheared Llama and ReSheared. Figure 3 shows our model achieves a 55.4% win rate, surpassing both the open-source Sheared Llama and ReSheared. This highlights that DRPruning offers a stronger foundation for further use.

		CC	C4	GitHub	Book	Wiki	ArXiv	StackEx
Data Ratio	**Constant**	67.0%	15.0%	4.5%	4.5%	4.5%	2.5%	2.0%
	DRO	54.6%	29.0%	3.7%	5.0%	3.8%	1.9%	1.9%
	DRPruning	55.7%	15.6%	2.4%	3.7%	18.4%	2.1%	2.1%
	Reference Ratio vs. Constant	47.1%	30.5%	2.9%	3.5%	9.1%	3.9%	3.0%
Loss	**Evaluation Loss vs. Constant**	+0.011	-0.010	+0.030	+0.007	-0.123	+0.003	+0.001
	Reference Loss vs. Constant	+0.067	+0.162	+0.093	+0.094	-0.108	+0.014	-0.053

Table 3: Data usage ratios, hyperparameter adjustments, and domain-specific loss for Constant, naïve DRO, and our strategy. The first three rows show the total data usage ratios during training, while the last three rows represent the hyperparameters and the evaluation loss at the end of training.

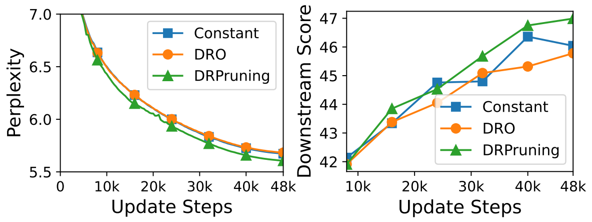

Figure 4: Effectiveness of our method compared to constant scheduling and naïve DRO during the 1.3B continued pretraining. Left figure: PPL trends; Right: average performance across 15 downstream tasks.

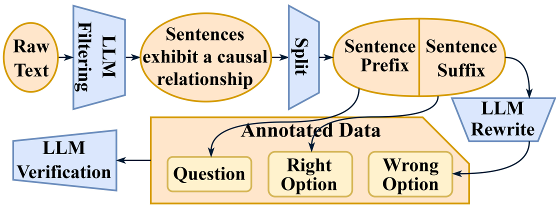

Figure 5: The generation procedure for our sentence continuation task. The orange nodes represent data storage nodes, while the blue trapezoidal nodes represent data processing nodes.

Question	If the latter described their efforts to adapt to European conditions,
Right Option	the former insisted that Muslims adhere to proper canons of learning and textual interpretation.
Wrong Option	it also highlighted the resilience and ingenuity that had brought them this far despite challenges.

Table 4: Case For the sentence continuation task.

Additionally, DRPruning introduces no extra GPU computation, with the only overhead stemming from data ratio calculation, contributing to less than 1.5% of the training time. Furthermore, we implemented parallel data ratio computation and demonstrated that it does not impact performance, as detailed in Appendix B.4.

5 Analysis

5.1 Ablation Study

We assess the impact of our dynamic data scheduling on 1.3B models, comparing it to constant scheduling and the naïve DRO method.

DRPruning significantly outperforms DRO method. As shown in Figure 4, our method consistently reduces PPL compared to the constant and DRO approach. In downstream tasks, DRPruning also outperforms both baselines, with performance improving steadily in the mid to late stages of training (-0.26 for DRO, +0.95 for ours). This underscores the sensitivity of DRO to hyperparameters and the necessity of dynamic adjustments.

Our strategy dynamically identifies underperforming domains. To highlight how our method differs from DRO, Table 3 presents data usage ratios, hyperparameter adjustments, and domain-specific loss. Our strategy identifies Wiki as underperforming compared to optimal, assigning it a lower reference loss (-0.11) and a higher reference ratio (+4.6%). This reduces loss on Wiki (-0.12) while keeping loss stable in other domains (maximum increase of +0.03). Besides, by dynamically adjusting the reference ratio, we improve the model's robustness across the full range of ratios from the initial to the new reference distribution. Compared to DRO, this offers robustness to a wider range of distribution shifts.

5.2 Robustness across Different Domains

While our method shows clear advantages in PPL, a gap remains between PPL and downstream performance. The lack of domain-specific test sets also limits further analysis. To address this, we create downstream tasks from unlabeled datasets for detailed domain-specific evaluation.

Automatic construction of sentence continuation tasks across domains. To assess base model performance, we use sentence continuation tasks

	Model	CC	C4	GitHub	Book	Wiki	ArXiv	StackExchange	Average
1.3B	**Constant**	35.00	40.00	88.00	47.75	21.75	82.00	72.50	55.29
	Sheared Llama	21.75	26.00	93.75	33.50	29.50	38.50	56.50	42.79
	ReSheared	30.50	30.25	89.25	32.00	23.00	81.00	47.50	47.64
	DRPruning	**44.00**	**51.50**	**94.75**	**48.00**	**33.50**	**86.50**	**90.00**	**64.04**
2.7B	**Sheared Llama**	81.25	**89.50**	95.50	**96.50**	**89.25**	90.50	82.75	**89.32**
	ReSheared	61.75	60.25	96.00	73.00	80.50	**93.75**	**92.00**	79.61
	DRPruning	**82.25**	77.75	**99.00**	86.75	87.50	79.50	89.25	86.00

Table 5: Domain-level results under the benchmark we generated. The abbreviations of tasks refer to the evaluation of seven domains used for training in RedPajama.

Base Model	Prune	PT	Method	EN	RU	ZH	JA	AR	TR	KO	TH	Average
XGLM-1.7B	X	X	-	55.06	52.97	51.02	51.00	42.89	37.99	49.00	38.63	47.32
Qwen1.5-1.8B	X	X	-	60.89	52.30	**56.13**	53.30	42.17	34.98	48.25	36.75	48.10
Qwen2-1.5B	X	X	-	61.58	57.83	55.72	55.30	43.31	35.98	49.25	36.02	49.37
Qwen2-1.5B	X	✓	**ReSheared**	**62.16**	58.95	54.93	**55.60**	43.91	37.27	**54.05**	39.96	50.85
Qwen2-1.5B	X	✓	**DRPruning**	61.67	**59.09**	54.01	54.95	45.14	**46.91**	52.65	**44.42**	**52.35**
Qwen2-7B	✓	✓	**DRPruning**	60.43	56.80	55.72	55.05	**45.69**	43.82	53.95	43.53	51.87

Table 6: The average performance on downstream tasks across multiple languages. "Prune" refers to the pruning procedure applied to the base model, while "PT" indicates continued pretraining on the provided dataset.

		EN	RU	ZH	JA	AR	TR	KO	TH
Data Ratio	**Default Reference Ratio**	27.7%	18.5%	13.0%	10.4%	9.1%	8.9%	6.7%	5.8%
	ReSheared	82.8%	5.9%	5.5%	2.2%	1.0%	0.8%	1.2%	0.6%
	DRPruning	19.0%	7.8%	12.9%	19.1%	9.2%	19.9%	6.7%	5.4%
	Reference Ratio vs. ReSheared	23.1%	9.2%	8.5%	20.3%	8.6%	17.8%	6.7%	5.8%
Loss	**Evaluation Loss vs. ReSheared**	+0.143	-0.067	-0.123	-0.330	-0.465	-0.841	-0.282	-0.304
	Reference Loss vs. ReSheared	+0.211	+0.008	-0.060	-0.267	-0.412	-0.804	-0.219	-0.229

Table 7: Data usage ratios, hyperparameter adjustments, and domain-specific loss for reproduction of Sheared Llama (ReSheared) and our approach (DRPruning) during continued pretraining from Qwen2 1.5B.

where the model selects the better continuation between two options. As shown in Figure 5, we use the SlimPajama test set as the correct sentence and have the model generate the incorrect alternatives. First, "LLM Filtering" selects consecutive sentences with a causal relationship, then "Split" each sentence into two parts: the first as the question, and the second as the right option. For this large-scale filtering, we use `GPT-4o-mini`.

Next, "LLM Rewrite" generates incorrect options. Since LLMs struggle to create incorrect but related content, we follow Deng et al. (2024) to first generate a reasonable continuation, then modify it to create an incorrect version. Finally, "LLMs Verification" scores both options and filters out cases where the scores don't match the true answers. We use `GPT-4o` in these procedures. We further select questions with the largest score differences. In total, we select 400 questions per domain. A case is shown in Table 4.

DRPruning outperforms the baselines consistently across the domains. As shown in Table 5, our method consistently outperforms the Constant and Sheared Llama scheduling strategies, achieving better downstream performance in most domains. This demonstrates that our approach not only improves learning in hard domains but also preserves or enhances performance in others. Additionally, our benchmark results align well with the average performance across 15 tasks, performing slightly below the open-source Sheared Llama 2.7B model. This consistency confirms the quality and reliability of our benchmark and results. Furthermore, we demonstrate that DRPruning benefits from finer domain segmentatio in Appendix B.5.

5.3 Robustness under Distribution Shifts

To verify our method under larger distribution shifts, we conduct experiments under the multilingual setting. To ensure a fair comparison, we reproduce DRPruning and ReSheared methods under the same experimental setup, detailed as follows:

Experimental setups. We use Qwen2 series models (Yang et al., 2024), which demonstrate su-

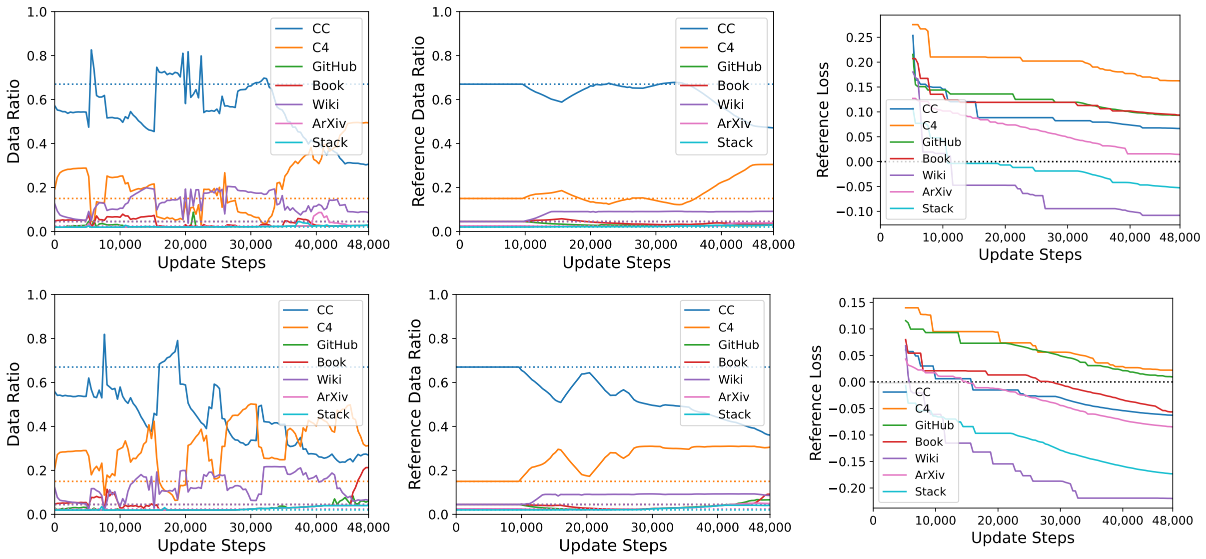

Figure 6: Variation of data ratio, reference data ratio, and reference loss during continued pretraining. The top three plots show results for pruning to 1.3B parameters, while the bottom three are for pruning to 2.7B parameters. Dashed lines in data ratio plots represent the initial reference data ratio for each domain. Reference loss plots display the difference from the initial value, with the dashed line at $y = 0$ indicating no change from the initial ones.

perior multilingual performance, as the base models and explore two approaches: (1) continued pretraining from Qwen2-1.5B, and (2) pruning Qwen2-7B and then continued pretraining. Due to grouped query attention differences, we keep the head dimension unchanged, resulting in a 1.8B target architecture. We use the CulturaX dataset (Nguyen et al., 2024) and select eight languages covered by Qwen2. The reference loss is initialized with Qwen2-7B on the validation set. For the reference ratio, we follow Conneau et al. (2020), upsampling low-resource languages with a smoothing rate of 0.3. We select various downstream tasks and report average performance. Detailed model configurations and metrics are provided in Appendix A.2.

Our method demonstrates superior distribution robustness compared to ReSheared. As shown in Table 6, our method outperforms ReSheared with an average gain of +1.50, and Qwen2-1.5B with +2.98. To analyze the source of these gains, as shown in Table 7, Sheared Llama focuses on hard-to-improve areas like English, where its loss remains above the reference loss, leading to a highly imbalanced data distribution. In contrast, our method dynamically identifies underperforming domains, increasing the reference loss for high-resource languages and lowering it for low-resource ones. This leads to more balanced data scheduling and better evaluation loss, further demonstrating the robustness of our approach in handling distribution shifts.

Continued pretraining from the pruned model underperforms from the pretrained ones. Continued pretraining from the pruned model result in a slight performance drop (average -0.48), despite the increased parameters (1.8B vs. 1.5B). This contrasts with Sheared Llama, where continued pretraining on a smaller model shows minimal gains. In our case, using different data and a stronger small model improves performance during continued pretraining, leading to different results. Furthermore, DRPruning demonstrates efficacy not only under continued pretraining but also under instruction tuning, detailed in Appendix B.6.

5.4 Analysis of Hyperparameter Adjustment

We analyze how our strategy adjusts training parameters, including the data ratio, reference data ratio, and reference loss, in the main experiment. The results are shown in Figure 6.

High reliability of our approach. The trends for the 1.3B and 2.7B targets are similar, with increased allocation to C4 and Wiki and reduced allocation to CC. This highlights limitations in the current hyperparameter settings while confirming the reliability of our dynamic scheduling. In the later training stages, the CC domain exhibits lower potential with slower loss convergence, prompting our strategy to reduce its weight and increase the weight for C4. Wiki data consistently shows higher potential, leading to a significantly higher reference data ratio and the largest reference loss reduction.

Effective real-time evaluation of reference loss. To accelerate convergence, we select the minimum predicted value, which raises concerns about the inability to increase the reference loss when domain potential decreases. However, the two rightmost figures show that our predictions are conservative and decrease gradually during training. When potential declines, the rate of decrease slows, resulting in a relatively higher reference loss. This favorable outcome arises because we use loss from a limited training duration instead of the fully trained loss used in scaling laws, leading to more cautious estimates. Further, the similar trends between the 1.3B and 2.7B models indicate that our method provides reasonable training expectations, potentially supporting model training across broader ranges.

6 Related Work

LLM Pruning. Unstructured pruning (Frankle and Carbin, 2019; Frantar and Alistarh, 2023; Sun et al., 2024) removes individual weights but offers limited speedup. This study focuses on structured pruning (Han et al., 2015; Wen et al., 2016), which removes entire structural components, making it more effective for improving efficiency (Liu et al., 2024; Rao et al., 2023). In task-specific models, extensive pruning can retain performance (Liu et al., 2017; Wang et al., 2020a; Lagunas et al., 2021; Xia et al., 2022; Kurtic et al., 2023). However, for LLMs, as training data increases (Hoffmann et al., 2022b), fewer redundant parameters remain, leading to significant performance degradation after pruning. To counter this, performance recovery techniques like continued pretraining are essential (Ma et al., 2023; Zhang et al., 2024b). However, continued pretraining of pruned models reduces loss at different rates across domains, resulting in less efficient data utilization (Xia et al., 2024). To address this, DRPruning dynamically adjusts the data distribution during training, ensuring balanced performance across domains.

Distributionally robust optimization (DRO). Overparameterized neural networks excel on i.i.d. test sets but struggle with underrepresented data groups (Hovy and Søgaard, 2015; Blodgett et al., 2016; Tatman, 2017). Unlike empirical risk minimization, which minimizes expected loss for a fixed distribution, MultiDDS (Wang et al., 2020c) optimizes the sampling distribution via gradient-based meta-learning but incurs higher computational and memory costs. In contrast, DRO (Delage and Ye, 2010; Ben-Tal et al., 2012; Bertsimas et al., 2014) improves performance without additional complexity (Hashimoto et al., 2018).

DRO finds a model that performs well across multiple possible test distributions. Group DRO (Sagawa et al., 2019) minimizes the worst-case loss over all domains without constraining potential distribution, while CVaR-Group DRO (Oren et al., 2019) averages the largest N group losses. These methods can be overly conservative, as they account for robustness to arbitrary subpopulations. Zhou et al. (2021) address this by constraining potential distribution within an f-divergence ball (Csiszár, 1967) around a reference data ratio, yielding promising results (Jiao et al., 2022).

DRO enhancement. DRO shows strong performance but relies on two main hyperparameters. The first is the reference loss, usually set by training an additional baseline model (Zhou et al., 2021; Jiao et al., 2022), though this is expensive for LLMs. Sheared Llama uses scaling laws of model size to predict the pruned model's performance. The second hyperparameter is the reference data ratio, often determined through temperature-based sampling (Arivazhagan et al., 2019; Conneau et al., 2020) or manually (Touvron et al., 2023a; Parmar et al., 2024). However, fixed ratios can hinder model convergence in challenging distributions. DRPruning shifts weight toward higher-loss domains, enhancing distribution robustness and improving downstream performance.

7 Conclusion

This paper presents DRPruning, a distributionally robust pruning method that addresses uneven performance degradation across domains during structured pruning. By utilizing and further improving distributionally robust optimization (DRO), our pruning method focuses more on domains with poorer performance, significantly accelerating performance recovery. It outperforms existing models and data scheduling methods in both monolingual and multilingual settings, achieving lower perplexity, higher task accuracy, and better instruction tuning outcomes. Further analysis demonstrates the robustness of our method against various domains and distribution shifts. Additionally, the dynamic adjustment of reference loss and data ratios exhibits broad applicability, with strong potential to support balanced training across diverse tasks.

Limitations

Exploration of smaller pruning ratios. Due to computational constraints, we are unable to explore pruning to larger models, i.e., employing smaller pruning ratios. Retaining a larger proportion of the model's parameters may lead to different outcomes in some experiments. For example, it remains to be investigated whether pruning larger models provides benefits, and whether it is better to continue pretraining from a pruned model or from a smaller, fully pretrained model.

More extensive continued pretraining. Xia et al. (2024) point out that pruned models exhibit higher training ceilings. Although good performance can be achieved with tens of billions of training samples, this study does not investigate whether training the models to full convergence using hundreds or thousands of billions of samples would yield better results than continuing pretraining from existing pretrained models under similar settings.

Validation in other scenarios. We have validated our method's effectiveness in the pruning phase, the pruning recovery phase, the continued pretraining phase, and the instruction tuning phase in Appendix B.6. Our method is expected to be applicable in broader contexts, such as pretraining from scratch and cross-domain post-training. Broader validation would further demonstrate the superiority of our approach.

Ethics Statement

Our work adheres to the ACL Ethics Policy and uses publicly available datasets for reproducibility. LLMs may exhibit racial and gender biases, so we strongly recommend users assess potential biases before applying the models in specific contexts. Additionally, due to the difficulty of controlling LLM outputs, users should be cautious of issues arising from hallucinations.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62206076), Guangdong S&T Program (Grant No. 2024B0101050003), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515011491), and Shenzhen Science and Technology Program (Grant Nos. ZDSYS20230626091203008, KJZD20231023094700001, KQTD20240729102154066). We would like to thank the anonymous reviewers and meta-reviewer for their insightful suggestions.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges.

Aharon Ben-Tal, Dick den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. 2012. Robust Solutions of Optimization Problems Affected by Uncertain Probabilities. *Management Science*.

Dimitris Bertsimas, Vishal Gupta, and Nathan Kallus. 2014. Data-Driven Robust Optimization.

Ning Bian, Xianpei Han, Le Sun, Hongyu Lin, Yaojie Lu, Ben He, Shanshan Jiang, and Bin Dong. 2024. Chatgpt is a knowledgeable but inexperienced solver: An investigation of commonsense problem in large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 3098–3110. ELRA and ICCL.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

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

Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic dialectal variation in social media: A case study of African-American English. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130, Austin, Texas. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge.

Together Computer. 2023. Redpajama: an open dataset for training large language models.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Imre Csiszár. 1967. Information-type measures of difference of probability distributions and indirect observation. *Studia Scientifica Mathematica Hungary*, 2:299–318.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Erick Delage and Yinyu Ye. 2010. Distributionally Robust Optimization Under Moment Uncertainty with Application to Data-Driven Problems. *Operations Research*, 58(3):595–612.

Hexuan Deng, Liang Ding, Xuebo Liu, Meishan Zhang, Dacheng Tao, and Min Zhang. 2023. Improving simultaneous machine translation with monolingual data. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 12728–12736. AAAI Press.

Hexuan Deng, Wenxiang Jiao, Xuebo Liu, Min Zhang, and Zhaopeng Tu. 2024. Newterm: Benchmarking real-time new terms for large language models with annual updates. In *Advances in Neural Information Processing Systems*, volume 37, pages 35760–35795. Curran Associates, Inc.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.

Simon Frieder, Luca Pinchetti, Alexis Chevalier, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. 2023. Mathematical capabilities of chatgpt. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Drew Fudenberg and David K Levine. 1998. *The theory of learning in games*, volume 2. MIT press.

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. 2024. A framework for few-shot language model evaluation.

Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. 2022. Scaling laws for neural machine translation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Trans. Assoc. Comput. Linguistics*, 10:522–538.

Song Han, Huizi Mao, and William J. Dally. 2015. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding.

Tatsunori B. Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. 2018. Fairness without demographics in repeated loss minimization. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1934–1943. PMLR.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022a. Training Compute-Optimal Large Language Models.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022b. Training Compute-Optimal Large Language Models.

Dirk Hovy and Anders Søgaard. 2015. Tagging performance correlates with author age. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 483–488, Beijing, China. Association for Computational Linguistics.

Wenxiang Jiao, Jen-tse Huang, Wenxuan Wang, Zhiwei He, Tian Liang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023a. ParroT: Translating during chat using large language models tuned with human translation and feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15009–15020, Singapore. Association for Computational Linguistics.

Wenxiang Jiao, Zhaopeng Tu, Jiarui Li, Wenxuan Wang, Jen-tse Huang, and Shuming Shi. 2022. Tencent's multilingual machine translation system for WMT22 large-scale African languages. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1049–1056, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023b. Is ChatGPT A Good Translator? A Preliminary Study. *ArXiv preprint*, abs/2301.08745.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. *ArXiv preprint*, abs/2001.08361.

Vid Kocijan, Thomas Lukasiewicz, Ernest Davis, Gary Marcus, and Leora Morgenstern. 2020. A Review of Winograd Schema Challenge Datasets and Approaches.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondrej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popovic, and Mariya Shmatova. 2023. Findings of the 2023 conference on machine translation (WMT23): llms are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation, WMT 2023, Singapore, December 6-7, 2023*, pages 1–42. Association for Computational Linguistics.

Eldar Kurtic, Elias Frantar, and Dan Alistarh. 2023. Ziplm: Inference-aware structured pruning of language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

François Lagunas, Ella Charlaix, Victor Sanh, and Alexander Rush. 2021. Block pruning for faster transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10619–10629, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shengrui Li, Junzhe Chen, Xueting Han, and Jing Bai. 2024. NutePrune: Efficient Progressive Pruning with Numerous Teachers for Large Language Models.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2021. Few-shot Learning with Multilingual Language Models.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3622–3628. ijcai.org.

Liangxin Liu, Xuebo Liu, Lian Lian, Shengjun Cheng, Jun Rao, Tengfei Yu, Hexuan Deng, and Min Zhang. 2024. Curriculum consistency learning for conditional sentence generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13865–13881, Miami, Florida, USA. Association for Computational Linguistics.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. 2017. Learning efficient convolutional networks through network slimming. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2755–2763. IEEE Computer Society.

Christos Louizos, Max Welling, and Diederik P. Kingma. 2018. Learning sparse neural networks through l_0 regularization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2024. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 4226–4237. ELRA and ICCL.

Yonatan Oren, Shiori Sagawa, Tatsunori B. Hashimoto, and Percy Liang. 2019. Distributionally robust language modeling. 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)*, pages 4227–4237, Hong Kong, China. Association for Computational Linguistics.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. 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)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Jupinder Parmar, Shrimai Prabhumoye, Joseph Jennings, Mostofa Patwary, Sandeep Subramanian, Dan Su, Chen Zhu, Deepak Narayanan, Aastha Jhunjhunwala, Ayush Dattagupta, Vibhu Jawa, Jiwei Liu, Ameya Mahabaleshwarkar, Osvald Nitski, Annika Brundyn, James Maki, Miguel Martinez, Jiaxuan You, John Kamalu, Patrick LeGresley, Denys Fridman, Jared Casper, Ashwath Aithal, Oleksii Kuchaiev, Mohammad Shoeybi, Jonathan Cohen, and Bryan Catanzaro. 2024. Nemotron-4 15B Technical Report.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Jun Rao, Xuebo Liu, Lian Lian, Shengjun Cheng, Yunjie Liao, and Min Zhang. 2024. CommonIT: Commonality-aware instruction tuning for large language models via data partitions. In *EMNLP*, pages 10064–10083, Miami, Florida, USA. Association for Computational Linguistics.

Jun Rao, Xv Meng, Liang Ding, Shuhan Qi, Xuebo Liu, Min Zhang, and Dacheng Tao. 2023. Parameter-efficient and student-friendly knowledge distillation. *IEEE Trans. Multim.*, pages 1–12.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. 2019. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Zhiqiang Shen, Tianhua Tao, Liqun Ma, Willie Neiswanger, Zhengzhong Liu, Hongyi Wang, Bowen

Tan, Joel Hestness, Natalia Vassilieva, Daria Soboleva, and Eric Xing. 2023. SlimPajama-DC: Understanding Data Combinations for LLM Training.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Anastasia Kozlova, Vladislav Mikhailov, and Tatiana Shavrina. 2024. mgpt: Few-shot learners go multilingual. *Trans. Assoc. Comput. Linguistics*, 12:58–79.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Rachael Tatman. 2017. Gender and dialect bias in YouTube's automatic captions. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 53–59, Valencia, Spain. Association for Computational Linguistics.

Alexey Tikhonov and Max Ryabinin. 2021. It's All in the Heads: Using Attention Heads as a Baseline for Cross-Lingual Transfer in Commonsense Reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3534–3546, Online. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and Efficient Foundation Language Models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open Foundation and Fine-Tuned Chat Models.

Chaoqi Wang, Guodong Zhang, and Roger B. Grosse. 2020a. Picking winning tickets before training by preserving gradient flow. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020b. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020c. Balancing training for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8526–8537, Online. Association for Computational Linguistics.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. 2016. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2074–2082.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2024. Sheared llama: Accelerating language model pre-training via structured pruning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Mengzhou Xia, Zexuan Zhong, and Danqi Chen. 2022. Structured pruning learns compact and accurate models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1513–1528, Dublin, Ireland. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing

Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 Technical Report.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. 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)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. 2024a. When scaling meets LLM finetuning: The effect of data, model and finetuning method. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. 2024b. LoRAPrune: Structured pruning meets low-rank parameter-efficient fine-tuning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 3013–3026, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. 2023. PyTorch FSDP: Experiences on Scaling Fully Sharded Data Parallel.

Chunting Zhou, Daniel Levy, Xian Li, Marjan Ghazvininejad, and Graham Neubig. 2021. Distributionally robust multilingual machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5664–5674, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Detailed Experimental Setup

A.1 Main Experiment

Model training. All experiments are conducted on 8 NVIDIA A100 40GB GPUs. The training hyperparameters for the main experiment are listed in Table 8, and the target model configuration for pruning is detailed in Table 9. Pruning takes approximately 12 hours for both the 1.3B and 2.7B models. Continued pretraining requires around 9 days for the 1.3B model and 18 days for the 2.7B model.

For both pruning and continued pretraining, we follow the configurations of Sheared Llama as closely as possible. We use fully sharded data parallel (Zhao et al., 2023) for parallel training and FlashAttention V1 (Dao et al., 2022) to speed up the training process. A cosine learning rate scheduler is employed, reducing the learning rate to 10% of its peak value.

DRO. We follow Sheared Llama to update the data ratio every 50 steps during pruning and every 400 steps during continued pretraining. For the DRO setup, we follow Zhou et al. (2021) closely. The constraint size ρ for the chi-square ball is set to $\{0.05, 0.1, 0.2\}$. Preliminary experiments show that $\rho = 0.1$ yields the best results, so we use this value in all experiments. Following their setup, we truncate the dynamic data ratio to prevent it from dropping below the minimum reference data ratio, which further ensures balanced domain training. We compute historical loss values using an exponential moving average, with the hyperparameter λ set to 0.1, which is also used for updating the reference data ratio. Besides, for the prediction of the reference loss, we maintain an average loss below 3×10^{-5}, demonstrating the effectiveness of our method.

Instruction tuning. For instruction tuning, the instruction begins with "You are a helpful assistant. Write a response that appropriately completes the request." We perform full-parameter fine-tuning for 5 epochs, with a learning rate of $5e^{-5}$, a warmup ratio of 3%, and a batch size of 128.

To evaluate instruction tuning, we follow the methodology of Sheared Llama, using LLMs to assess model performance. Given outputs from two models, we ask the LLM to determine which is better using the prompt: “Here is the user request: .... Here are the two outputs for this request: Output A: .... Output B: .... Which output is better, A or B?”. Since Wang et al. (2024) note that using GPT models as evaluators can lead to preference shifts when output order is reversed, we randomly switch the positions of the outputs to ensure each result appears as Output A or Output B equally. We report the average win rate to mitigate position bias. The model `gpt-4o-2024-08-06` is used for evaluation.

A.2 Multilingual Experiment

Model training. The experimental setup is largely consistent with Appendix A.1. Given the extended training duration, we standardize the training to 40,000 steps. The configuration of the target model for pruning is detailed in Table 10. To maintain a consistent ratio between the number of heads and KV heads required by the structured pruning method, we keep the head dimension unchanged and add two extra heads, increasing the number of parameters to 1.8B. Continued pretraining takes around 8 days for the 1.5B model and 12 days for the 1.8B model.

Data. We use the CulturaX dataset (Nguyen et al., 2024), a large multilingual resource with 6.3 trillion tokens across 167 languages, integrating mC4 and OSCAR, and meticulously cleaned and deduplicated. We select eight languages covered by Qwen2: English (EN), Russian (RU), Chinese (ZH), Japanese (JA), Arabic (AR), Turkish (TR), Korean (KO), and Thai (TH), representing diverse language families.

Metrics. We adopt the experimental setups from previous studies and evaluate performance on downstream tasks in a zero-shot setting. Specifically, we follow XGLM (Lin et al., 2021) and mGPT (Shliazhko et al., 2024), covering tasks such as natural language inference (XNLI; Conneau et al., 2018), Winograd schema challenge (XWINO; Tikhonov and Ryabinin, 2021), commonsense reasoning (XStoryCloze; Lin et al., 2021), and paraphrase detection (PAWSX; Yang et al., 2019). Task coverage varies across languages, and not all tasks include all languages in our training set. We report results for languages overlapping between tasks and our training set, providing average performance if a language appears in multiple tasks. The lm-evaluation-harness package (Gao et al., 2024) is used for the comprehensive evaluation of downstream tasks.

	Pruning	Contined Pretraining
Training Steps	3,200	48,000
Learning rate of z, ϕ, λ	1.0	-
Learning Rate of θ	0.0001	0.0001
LR warmup ratio	10%	3%
Batch size (tokens)	131K	1M
Ratio update interval m **(steps)**	50	400

Table 8: Training hyperparameters for the main experiment.

Model	#Param	#Layers	Hidden	Intermediate	#Heads	Head Dim
Pruned-0.5B	0.5B	24	1024	2816	8	128
Pruned-1.3B	1.3B	24	2048	5504	16	128
Pruned-2.7B	2.7B	32	2560	6912	20	128
Llama2-7B	6.7B	32	4096	11008	32	128

Table 9: The model configurations for the target model of pruning and the base models for the main experiment.

Model	#Param	#Layers	Hidden	Intermediate	#Heads	#KV Heads	Head Dim
Pruned-1.8B	1.8B	28	1536	8960	14	2	128
Qwen2-1.5B	1.5B	28	1536	8960	12	2	128
Qwen2-7B	7.6B	28	3584	18944	28	4	128

Table 10: The model configurations for the target model of pruning and the base models for the multilingual experiment.

B Supplement Experimental Results

B.1 Analysis of Mask Similarity

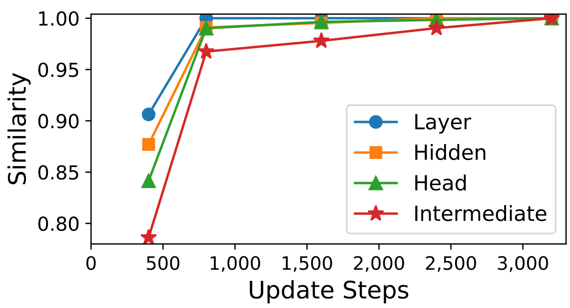

Figure 7: Convergence of masks over 3200 pruning steps. Similarity indicates the similarity between pruning decisions at a certain step and the final decisions at step 3200. “Layer”, “Hidden”, “Head”, and “Intermediate” correspond to the four pruning dimensions.

To perform a detailed analysis of the masks, we extract the masks generated during training and prune the model by removing components with the lowest scores, shaping the model according to the target specifications. We then calculate the probability of the model making consistent pruning decisions for each substructure and examine how different training steps or strategies influence the masks. Specifically, we apply Sheared Llama, constant, and our proposed strategies, using two distinct random seeds for pruning, which yields six unique 1.3B models. We then analyze the similarities across these models.

Structure	Same	Different
Layer	81.25±3.61	78.65±2.24
Hidden	60.33±0.78	65.02±1.47
Head	68.88±0.47	70.53±0.65
Intermediate	56.37±0.15	56.40±0.14

Table 11: Mask similarity mean values and the standard error of the mean under different data scheduling strategies and random seeds. “Same” indicates using identical data scheduling but different random seeds, while “Different” indicates using different data scheduling strategies.

The masks converge quickly during training. The convergence speed of the masks during training is illustrated in Figure 7. Pruning achieves over 75% similarity within 400 steps and over 95% within 800 steps, indicating that effective results can be obtained with relatively few pruning steps. While layer pruning converges rapidly, pruning intermediates of fully connected layers is slower, suggesting that coarser-grained decisions converge more quickly than finer-grained decisions.

Tasks	Pruning				Continued Pretraining			
	To: 1.3B		To: 2.7B		To: 1.3B		To: 2.7B	
	From: 7B	From: 13B	From: 7B	From: 13B	From: 7B	From: 13B	From: 7B	From: 13B
ARCC (25)	**23.21**	22.10	**30.29**	27.30	**33.62**	32.17	**40.53**	40.36
ARCE	**42.26**	40.07	**53.11**	47.31	**60.90**	58.92	**67.13**	66.58
BoolQ	59.69	**59.88**	59.36	**60.12**	**63.36**	56.88	65.08	**67.13**
HelS (10)	**35.27**	32.38	**48.07**	41.62	**58.88**	58.70	**69.22**	67.67
LAMB	**38.27**	34.08	**51.80**	46.56	**60.28**	59.87	**66.91**	66.23
LogiQA	**26.73**	25.50	**27.19**	24.42	**28.88**	25.65	28.73	**29.80**
MMLU (5)	24.82	**25.78**	24.86	**25.56**	**27.28**	26.76	26.99	**27.60**
NQ (5)	**1.91**	1.52	**4.02**	2.35	**10.44**	8.86	**15.82**	13.24
PIQA	**61.81**	61.32	**67.08**	64.15	**72.69**	72.31	**75.19**	74.27
SciQ	**79.80**	79.60	**86.30**	83.30	**87.70**	87.30	89.80	**91.00**
SQuAD	**13.80**	6.87	17.05	**18.61**	**35.06**	28.52	44.69	**46.55**
TriQA (5)	**5.26**	3.33	**11.75**	6.77	**28.10**	24.40	**43.33**	38.17
TruthQA	32.99	**34.15**	31.12	**31.50**	**29.68**	29.66	30.13	**30.75**
WinoG	**51.62**	49.09	**54.14**	53.83	**58.01**	56.83	**64.72**	62.04
WSC	**36.54**	**36.54**	**36.54**	**36.54**	50.00	**60.58**	**46.15**	36.54
Average	**35.60**	34.15	**40.18**	38.00	**46.99**	45.83	**51.63**	50.53

Table 12: The performance of pruning Llama2-7B and 13B models down to 1.3B and 2.7B parameters. "Pruning" refers to using the pruned model without continued pretraining, while "Continue Pretraining" means using the model after continued pretraining. "From" and "To" indicate the size of the source and target model, respectively. **Bold** indicates superior performance when pruning from 7B or 13B.

The randomness of pruning decisions is significant. We analyze mask similarity across training sessions with different random seeds under identical and distinct data scheduling strategies. The results, shown in Table 11, present mean values and standard error of the mean. The trend across pruning dimensions aligns with the previous findings: similarity is higher at the coarse-grained layer level and lower at the finer-grained intermediate level. Besides, comparisons across three pairs under identical settings and twelve pairs under different settings show consistently low similarity. However, no significant difference in perplexity is observed, suggesting that the model's interchangeable parameters allow similar outcomes despite different pruning decisions. This randomness obscures variations caused by differing data distributions.

B.2 Pruning from Larger LLMs

We investigate whether pruning from LLMs with a higher pruning ratio provides additional benefits. Experiments are conducted in the monolingual setting, consistent with the main text, to compare the effects of pruning from Llama2-7B and Llama2-13B.

The results, presented in Table 12, indicate that pruning from the 13B model consistently yields worse outcomes, regardless of whether continued pretraining is applied. On average, this approach results in a downstream performance decrease of 1.47. These findings suggest that pruning from a larger model leads to a more significant performance decline, often producing inferior results under a fixed training budget, especially under a high pruning ratio.

B.3 Robustness Verification

First, all comparisons in our experiments are mainly based on **DRPruning** and **ReSheared**, rather than the official open-source version of Sheared LLaMA. This is because, under the 2.7B configuration, we were unable to reproduce the results using RedPajama or the filtered SlimPajama dataset as the continued pretrained dataset. However, for the 1.3B model, our reproduced version achieved performance surpassing Sheared LLaMA. Therefore, to ensure a fair comparison, we conducted most comparisons against ReSheared.

As shown in Table 2, our method demonstrates relatively small improvements over ReSheared in downstream evaluations. To address this issue, we provide the following analysis. First, our results consistently outperform ReSheared across various metrics, including PPL, downstream task performance for both pruned and continued pretrained models, domain-specific evaluation, and win rate after instruction tuning. These consistent and stable improvements across multiple dimensions provide solid evidence of the effectiveness of our approach.

To further demonstrate the robustness of our method, we conduct significance testing. Specifically, we design five distinct prompts for each

Tasks	**ReSheared 1.3B**						**DRPruning 1.3B**					
	P1	**P2**	**P3**	**P4**	**P5**	**Avg.**	**P1**	**P2**	**P3**	**P4**	**P5**	**Avg.**
ARCC (25)	**34.30**	**33.62**	**34.04**	**33.53**	**35.15**	**34.13**	33.62	33.36	33.45	33.28	34.13	33.57
ARCE	60.35	**59.76**	61.03	59.05	60.98	60.23	**60.90**	58.00	**61.53**	**59.81**	**61.28**	**60.30**
BoolQ	61.01	58.90	**62.32**	**62.26**	62.48	61.38	**63.36**	**63.12**	61.50	59.11	**62.48**	**61.93**
HelS (10)	**63.06**	**63.05**	**63.12**	**62.99**	**63.10**	**63.06**	58.88	58.77	58.66	58.66	58.82	58.76
LAMB	58.84	59.83	59.65	60.02	60.02	59.67	**60.28**	**60.90**	**61.09**	**61.28**	**61.23**	**60.96**
LogiQA	28.11	28.11	**31.03**	28.42	**29.34**	29.00	**28.88**	**29.49**	29.65	**29.19**	27.80	**29.03**
MMLU (5)	26.60	25.92	25.69	25.56	25.61	25.87	**27.28**	**26.79**	**26.86**	**26.54**	**26.33**	**26.76**
NQ (5)	8.39	7.73	8.31	7.98	8.34	8.14	**10.44**	**9.58**	**9.75**	**9.11**	**10.08**	**9.79**
PIQA	**74.59**	**74.43**	**74.92**	**73.88**	**74.65**	**74.49**	72.69	71.98	72.09	72.20	72.47	72.27
SciQ	86.40	85.70	88.20	85.90	87.50	86.76	**87.70**	**88.30**	**89.40**	**88.20**	**89.50**	**88.64**
SQuAD	**37.59**	**34.25**	**40.39**	**44.09**	**35.76**	**38.43**	35.06	33.44	39.59	41.53	32.31	36.38
TriQA (5)	24.98	25.06	25.10	23.61	25.00	24.75	**28.10**	**27.62**	**28.45**	**26.47**	**28.01**	**27.72**
TruthQA	28.09	29.84	30.33	29.20	29.16	29.31	**29.68**	**32.07**	**31.69**	**30.44**	**30.87**	**30.95**
WinoG	**60.06**	**59.59**	59.12	**61.01**	**59.04**	**59.68**	58.01	59.27	**60.62**	59.35	58.64	59.21
WSC	40.38	36.54	36.54	36.54	41.35	38.27	**50.00**	**50.00**	**49.04**	**55.77**	**48.08**	**50.58**
Average	46.18	45.49	46.65	46.27	46.50	46.21	**46.99**	**46.85**	**47.56**	**47.40**	**46.80**	**47.12**

Table 13: Performance comparison between ReSheared 1.3B and DRPruning 1.3B across five different prompts. "P1" to "P5" represent five distinct prompts. Other abbreviations follow the definitions in Table 2. **Bold** indicates superior performance when comparing ReSheared and DRPruning.

Tasks	**ReSheared 2.7B**						**DRPruning 2.7B**					
	P1	**P2**	**P3**	**P4**	**P5**	**Avg.**	**P1**	**P2**	**P3**	**P4**	**P5**	**Avg.**
ARCC (25)	40.10	**39.85**	40.44	40.36	40.10	40.17	**40.53**	39.08	**40.44**	**40.96**	**40.96**	**40.39**
ARCE	**67.72**	**67.30**	**67.42**	63.55	67.38	**66.67**	67.13	64.52	67.26	**64.39**	**67.55**	66.14
BoolQ	64.92	66.48	64.43	62.97	63.12	64.37	**65.08**	**67.71**	**66.64**	**66.33**	**66.36**	**66.43**
HelS (10)	**72.03**	**72.05**	**72.06**	**72.00**	**72.12**	**72.05**	69.22	69.24	69.02	69.04	69.17	69.14
LAMB	66.18	66.31	66.19	66.43	67.01	66.41	**66.91**	**67.13**	**68.08**	**67.18**	**67.77**	**67.41**
LogiQA	26.27	26.27	**27.65**	29.95	27.50	27.50	**28.73**	**27.96**	27.19	**30.11**	**28.57**	**28.51**
MMLU (5)	25.70	24.81	25.21	25.22	25.65	25.32	**26.99**	**26.75**	**27.00**	**26.81**	**27.01**	**26.91**
NQ (5)	13.49	13.60	13.71	13.19	13.46	13.50	**15.82**	**16.23**	**15.96**	**15.84**	**16.09**	**15.99**
PIQA	**76.71**	**76.88**	**76.17**	**75.52**	**75.95**	**76.27**	75.19	74.21	75.19	74.86	74.70	74.83
SciQ	**90.10**	**90.30**	91.70	88.10	91.50	90.34	89.80	89.40	**92.70**	**89.10**	**91.80**	**90.56**
SQuAD	**49.17**	**44.33**	**50.18**	**51.86**	**37.49**	**46.60**	44.69	37.94	47.94	44.93	30.81	41.25
TriQA	40.14	40.11	40.06	39.72	40.43	40.09	**43.33**	**41.84**	**43.70**	**43.02**	**43.44**	**43.07**
TruthQA	28.41	30.40	29.74	**29.87**	**30.05**	29.71	**30.13**	**31.03**	**30.06**	29.80	29.61	**30.10**
WinoG	63.38	64.17	63.77	65.04	64.64	64.20	**64.72**	**64.64**	**65.59**	**66.54**	**65.04**	**65.29**
WSC	36.54	37.50	37.50	37.50	36.54	37.12	**46.15**	**57.69**	**43.27**	**63.46**	**51.92**	**52.31**
Average	50.72	50.69	51.08	50.75	50.20	50.69	**51.63**	**51.69**	**52.00**	**52.82**	**51.39**	**51.89**

Table 14: Performance comparison between ReSheared 2.7B and DRPruning 2.7B across five different prompts. Abbreviations follow the definitions in Table 13.

task to test its resilience to input perturbations, and conduct paired t-tests between ReSheared and DRPruning. The results under the 1.3B and 2.7B configurations are presented in Table 13 and 14, respectively. For the 1.3B model, the t-statistic is 2.0318 with a p-value of 0.0458, while for the 2.7B model, the t-statistic is 2.1962 with a p-value of 0.0312. When combined, the overall t-statistic reaches 2.9922 with a p-value of 0.0032. These results provide strong evidence of the statistical significance of our method, with a p-value below 0.05. The prompts used are given in Table 15.

B.4 Efficiency Discussion

DRPruning focuses solely on data distribution without introducing additional GPU computations. The only extra cost stems from data ratio calculation, which is entirely handled on CPU. During continued pretraining, each update takes 39.02s, while pruning with an additional parameter increases it to 99.52s. Over the full training process, pruning adds 1.8 hours, and continued pretraining adds 1.3 hours. This accounts for a 1.3% increase in training time for the 1.3B model and 0.7% for the 2.7B model.

To eliminate extra computation overhead, we implemented parallel data ratio calculation, ensur-

ARCC, ARCE, BoolQ, NQ, PIQA, SciQ, TriQA	[Passage]. Question: [Question]. Answer:
	[Passage]. Q: [Question]. A:
	[Passage]. Answer the question [Question]. Answer:
	[Passage]. Please respond to the following question: [Question]. Response:
	[Passage]. Please answer the following: [Question]. Answer:
HelS, LAMB, WinoG	[Sentence].
	Continue the narrative below: [Sentence].
	Provide a logical continuation for the text below: [Sentence].
	Extend the following scenario: [Sentence].
	Please carry on with the next part of the story: [Sentence].
LogiQA	Passage: [Passage]. Question: [Question]. Choices: A. [Choice1]. B. [Choice2]. C. [Choice3]. D. [Choice4]. Answer:
	Here is a passage: [Passage]. Based on the above, answer the following question: [Question]. Select the correct option: A. [Choice1]. B. [Choice2]. C. [Choice3]. D. [Choice4]. Your answer:
	Passage: [Passage]. **Question:** [Question]. **Choices:** - A. [Choice1]. - B. [Choice2]. - C. [Choice3]. - D. [Choice4]. **Answer:**
	Passage: [Passage]. ### Question: [Question]. #### Options: A) [Choice1]. B) [Choice2]. C) [Choice3]. D) [Choice4]. ### Answer:
	You are given the following passage: [Passage]. Answer the question based on the passage: [Question]. Select one of the following options: A) [Choice1]. B) [Choice2]. C) [Choice3]. D) [Choice4]. Your Answer:
MMLU	Q: [Question]. (A) [Choice1] (B) [Choice2] (C) [Choice3] (D) [Choice4] A:
	Please provide the correct answer to the math problem below: [Question]. A. [Choice1]. B. [Choice2]. C. [Choice3]. D. [Choice4]. Answer:
	Determine the solution to the following: [Question]. A. [Choice1]. B. [Choice2]. C. [Choice3]. D. [Choice4]. Answer:
	What is the correct answer to the following question? [Question]. A. [Choice1]. B. [Choice2]. C. [Choice3]. D. [Choice4]. Answer:
	What is the solution to this math problem? [Question]. Options: A) [Choice1]. B) [Choice2]. C) [Choice3]. D) [Choice4]. Answer:
SQuAD	Title: [Title]. Background: [Context]. Question: [Question]. Answer:
	Context: [Context]. Question: [Question]. Answer:
	Given the following text: [Context]. Answer the question below: [Question]. Answer:
	Information: [Title]. [Context]. Please answer the following: [Question]. Answer:
	Background Information: [Context]. Please address the following question: [Question]. Answer:
TruthQA	[TruthQA Few Shot]. Q: [Question]. A:
	[TruthQA Few Shot]. What is the answer to this question? [Question]. A:
	[TruthQA Few Shot]. Question: [Question]. Provide your answer:
	[TruthQA Few Shot]. Q: [Question]. Please provide the answer (A):
	[TruthQA Few Shot]. Provide an answer to the following question: [Question]. Answer:
WSC	Passage: [Passage]. Question: In the passage above, does the pronoun "*[Pronoun]*" refer to "*[Noun]*"? Answer:
	Analyze the following text: [Passage]. Question: Is the pronoun "*[Pronoun]*" referring to "*[Noun]*"? Answer:
	Examine the following passage: [Passage]. Question: In this passage, does the pronoun "*[Pronoun]*" refer to "*[Noun]*"? Answer:
	Passage Analysis: [Passage]. Question: Does the pronoun "*[Pronoun]*" in the passage refer to "*[Noun]*"? Answer:
	Analyze the text below: [Passage]. Question: Is the pronoun "*[Pronoun]*" referring to "*[Noun]*"? Answer:

Table 15: Prompts used for significance testing. For each task, we designed five prompts.

ing training remains uninterrupted. This introduces a one- to two-step update delay, which does not affect performance. To prove this, a small-scale experiment, following the main setup, is conducted with a 0.5B target model for 24k steps in continued pretraining.

Results are in Table 16. On four NVIDIA A800 80GB GPUs, our method requires less training time after parallelization. However, before parallelization, pruning takes 44.15 hours, which is longer than ReSheared. PPL is 17.01 and 16.88 before and after parallelization, respectively, demonstrating that parallelization improves efficiency without compromising performance.

Method	Pruning		Cont. PT		
	PPL ↓	Time ↓	PPL ↓	Task ↑	Time ↓
ReSheared	20.07	43.96	8.37	36.33	225.42
DRPruning	**16.88**	**43.74**	**7.68**	**36.49**	**221.85**

Table 16: PPL, training time (in hours), and downstream task performance (Task) of 0.5B pruned models.

Additionally, our method maintains a lower PPL, outperforming ReSheared. However, improvements in downstream tasks are marginal, with performance on many tasks approaching or even falling below random guessing. Given the extremely high PPL after pruning (16.88 for 0.5B, 9.83 for 1.3B, 7.40 for 2.7B), we conclude that pruning from 7B to 0.5B leads to a performance collapse, making effective recovery challenging.

B.5 Analysis of More Fine-Grained Domains

DRPruning demonstrates superior performance in fine-grained domain segmentation. To substantiate this claim, we conduct further analysis on pruning experiments from Llama-2-7B to 1.3B. We then perform a more detailed domain segmentation by dividing CC into 10 parts and C4 into 3 parts, ensuring that each domain accounts for approximately 5% of the total data, indicated as fine-grained (**Fine**). Specifically, we use the all-MiniLM-L6-v2 (Wang et al., 2020b) model to encode the input sentences, followed by k-means clustering to segment the data into smaller domains. For each fine-grained domain, we retrain 100 samples as the validation set. As for the test set, we still use the one provided by Sheared Llama, with 500 samples each for the coarse-grained domains. We conduct pruning for 1600 steps, with other experimental settings aligned with those in our paper. We compare this with the method before segmentation, indicated as coarse-grained (**Coarse**).

Results are in Table 17, where we report the validation set cross-entropy (CE) for the segmented domains, where the category names are summarized by Deepseek R1 (DeepSeek-AI, 2025) from 100 texts within each domain. Results demonstrate that fine-grained segmentation can accelerate the convergence, especially in the early stage of training, and obtain better performance across these domains. After more extensive training, the advantage is still maintained in the majority of domains.

We also report the results on the test set, as shown in Table 18, which lead to similar conclusions as those on the validation set across all domains. Furthermore, we migrate the fine-grained method to ReSheared and find that the performance is not satisfactory. The reason could be that, with more fine-grained domains, the hyperparameter settings for reference loss and data ratio become more complex, making our dynamic hyperparameter facility method more critical.

Domain	800 Steps		1600 Steps	
	Coarse	Fine	Coarse	Fine
CC-music	3.008	**2.974**	2.844	**2.835**
CC-technology	2.912	**2.877**	2.753	**2.736**
CC-sports	3.013	**2.969**	2.851	**2.827**
CC-environment	2.707	**2.698**	2.568	**2.567**
CC-medical	2.789	**2.734**	2.629	**2.590**
CC-corporate	2.743	**2.691**	2.581	**2.553**
CC-entertainment	3.070	**3.057**	**2.899**	2.911
CC-politics	3.033	**3.001**	2.865	**2.858**
CC-legal	3.065	**3.033**	2.892	**2.891**
CC-culture	3.099	**3.067**	2.936	**2.927**
C4-forum	3.112	**3.086**	2.955	**2.946**
C4-business	3.139	**3.107**	2.982	**2.969**
C4-lifestyle	3.178	**3.157**	3.019	**3.013**
Average	2.990	**2.958**	2.829	**2.817**

Table 17: Cross-entropy of using more fine-grained domains on the validation set. “Coarse” and “Fine” means using coarse-grained or fine-grained domain split. “800 Steps” and “1600 Steps” means the total training step for pruning.

B.6 Analysis on Instruction Tuning

DRPruning demonstrates efficacy not only under pruning and continued pretraining but also under instruction tuning. To verify this, we conduct experiments on machine translation tasks. Specifically, for training data, we employ the News-Commentary (Kocmi et al., 2023) dataset, and perform downsampling. For domains with larger data volumes, we sample 10% of the data, while for those with smaller volumes, we sample 50%, using the sampled data ratios as the default. For validation and testing, we select the FLORES-200 dev and devtest sets (Goyal et al., 2022), respectively. Following Jiao et al. (2023a), we adopt 33 prompts during training, and mask the instruction, training solely on the response. Regarding training, we utilize Qwen2-1.5B as the base model, with a batch size of 128 over 2k steps and a learning rate of 2e-5. The reference loss is computed based on Qwen2-7B, and other settings remain consistent with the continued pretraining described in the main paper.

The experimental results are shown in Table 19, reporting BLEU scores across all language directions. Our method demonstrates significant im-

Domain	800 Steps			1600 Steps		
	ReSheared Fine	DRPruning Coarse	DRPruning Fine	ReSheared Fine	DRPruning Coarse	DRPruning Fine
CC	3.373	2.886	**2.856**	3.196	2.723	**2.714**
C4	3.676	3.128	**3.102**	3.512	2.969	**2.959**
GitHub	2.189	1.446	**1.421**	2.059	1.286	**1.277**
Book	3.594	3.131	**3.117**	3.383	**2.946**	2.956
Wiki	2.978	3.550	**3.491**	**2.404**	3.261	3.268
ArXiv	3.184	2.185	**2.170**	3.034	2.022	**2.016**
StackExchange	**3.278**	2.496	2.473	3.157	2.321	**2.313**
Average	3.180	2.685	**2.658**	2.963	2.501	**2.497**

Table 18: Cross-entropy of using more fine-grained domains on the test set. All abbreviations used here align with Table 5 and Table 17.

Language	Data	ReSheared	DRPruning
EN-ZH	39.9k	25.4	**27.4**
ZH-EN	39.9k	21.9	**23.1**
EN-DE	39.3k	21.5	**26.2**
DE-EN	39.3k	32.8	**33.1**
EN-RU	34.0k	20.9	**24.1**
RU-EN	34.0k	26.7	**27.2**
EN-AR	17.4k	9.4	**9.6**
AR-EN	17.4k	21.0	**24.5**
EN-ID	2.9k	11.5	**20.9**
ID-EN	2.9k	**27.4**	26.8
EN-HI	2.2k	**5.6**	1.5
HI-EN	2.2k	7.5	**14.2**
EN-JA	0.8k	5.1	**8.5**
JA-EN	0.8k	15.0	**18.6**
Average		18.0	**20.4**

Table 19: BLEU Scores for instruction tuning analysis on machine translation. “Language” indicates the source and target language pair, connected by a hyphen. “Data” refers to the training data volume.

Language	Default Ratio	ReSheared	DRPruning
EN-ID	1.05	0.42	2.03
ID-EN	1.05	0.11	1.27
EN-HI	0.81	72.14	0.93
HI-EN	0.81	0.48	1.05
EN-JA	0.31	0.47	2.14
JA-EN	0.31	0.05	0.44

Table 20: Data ratios for instruction tuning analysis on machine translation. “Default Ratio” denotes the percentage of the total training data comprised by each domain’s data, as determined by its original volume.

provements over ReSheared. To analyze the reasons, we report the data ratios used during training in Table 20. When the performance in a certain domain consistently falls below expectations, ReSheared leads to distribution collapse in scenarios with distribution shift, resulting in excessive training epochs on a small amount of data. However, our method, by dynamically adjusting expectations (dynamic reference loss) and data ratio constraints (DRO + dynamic reference data ratio), robustly allocates more weight to low-resource languages without significantly deviating from the preset data distribution.