SaulLM-54B & SaulLM-141B: Scaling Up Domain Adaptation for the Legal Domain

Pierre Colombo equall
Equall
MICS - CentraleSupelec

Telmo Pires equall
Equall

Malik Boudiaf equall
Equall

Rui Melo equall
Equall

Dominic Culver equall
Equall

Etienne Malaboeuf
CINES

Gabriel Hautreux
CINES

Johanne Charpentier
CINES

Michael Desa equall
Equall

Abstract

In this paper, we introduce `SaulLM-54B` and `SaulLM-141B`, two large language models (LLMs) tailored for the legal sector. These models, which feature architectures of 54 billion and 141 billion parameters, respectively, are based on the Mixtral architecture. The development of `SaulLM-54B` and `SaulLM-141B` is guided by large-scale domain adaptation, divided into three strategies: (1) the exploitation of continued pretraining involving a base corpus that includes over 540 billion of legal tokens, (2) the implementation of a specialized legal instruction-following protocol, and (3) the alignment of model outputs with human preferences in legal interpretations. The integration of synthetically generated data in the second and third steps enhances the models' capabilities in interpreting and processing legal texts, effectively reaching state-of-the-art performance and outperforming previous open-source models on `LegalBench-Instruct`. This work explores the trade-offs involved in domain-specific adaptation at this scale, offering insights that may inform future studies on domain adaptation using strong decoder models. Building upon `SaulLM-7B`, this study refines the approach to produce an LLM better equipped for legal tasks. We are releasing base, instruct, and aligned versions on top of `SaulLM-54B` and `SaulLM-141B` under the MIT License to facilitate reuse and collaborative research.

1 Introduction

LLMs have demonstrated exceptional capabilities across various domains [1, 67, 60, 76, 39, 40, 77, 7, 28], excelling in tasks such as language translation [6], medical diagnostics [16, 11, 12], and automated code generation [4, 42, 32], among others. These achievements highlight the potential for human-like communication through large language models (LLMs). Despite the significant potential benefits, the adaptation of most recent LLMs for legal tasks has not been extensively examined, with only two recent studies cited from [18, 55, 86], and its impact on society could be substantial. Indeed, at a time when legal systems in many countries are overburdened [71], the development of robust and high-performing legal LLMs could provide critical support to lawyers and judicial systems [65, 10]. However, adapting LLMs to the legal domain presents unique challenges, particularly because of the vast scale involved, with hundreds of billions of existing legal data available.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-4120

Previous efforts to tailor LLMs to the legal sector have encountered significant challenges [18, 55, 86]: first, a limited model scale, capped at 7/12B parameters, which is considerably smaller than the largest open-source models [7, 40]; second, training datasets restricted to no more than 30 billion tokens, significantly fewer than potentially available tokens [30, 54]. Given the importance of scale and breadth in effectively adapting LLMs to new domains, this paper aims to answer the following research question:

How much can we improve the specialization of generic LLMs for legal tasks by scaling up both model and corpus size?

In this paper, we present an empirical study on the scalability and domain adaptation of LLMs in the legal sector. Relying on a corpus exceeding 500B tokens and models up to 141B parameters, our research seeks to address the gaps in the examination of legal applications. A novel aspect of our approach is the adaptation of large-scale Mixture of Experts (MoE) models with 54B and 141B parameters, which have gained significant traction in recent months [93, 22, 45, 90, 87, 61]. Formally, this study makes two principal contributions:

1. Comprehensive Analysis of Domain Adaptation Strategies for Legal LLMs Domain adaptation for legal LLMs remains a challenging and somewhat underexplored area. This work advances the field by specializing each step in the process of developing modern LLMs, from continued pretraining to instruction fine-tuning and alignment, relying on both synthetic and real data. This paper offers a fresh perspective on the efficacy of each step and its value for adapting to the legal domain, potentially guiding further research in the legal domain as well as in other expert domains.

2. `SaulLM-54B` & `SaulLM-141B`: Joining SaulLM-7b to form a Family of Legal LLMs under Permissive License[1] We specialize general-purpose, large-scale LLMs for the law. This work represents an ambitious advancement in terms of scale and leveraging the increasingly popular MoE architecture. While this architecture is widely used, its specific applications within focused domains, particularly the legal sector, are still largely unexplored. By releasing these models, we aim to foster further research in legal NLP and contribute to unlocking the full potential of LLMs.

2 Related Work

2.1 Domain Specialization For Large Language Models

The process of domain specialization for LLMs has demonstrated promising results in areas such as medicine [16], science [74], translation [6, 5], or code [66, 42, 4]. Models like SciBERT [9], PubMedBERT [75], Galactica [74] and Meditron [16] have been specifically trained on domain-related corpora to enhance their performance. Studies have identified that both the scale of the model and the size of the in-domain data are crucial for achieving strong domain adaptation [16, 66].

In the legal domain, earlier models such as LegalBERT [15], InCaseLawBERT [59], and SaulLM-7B [18], among others, while pioneering, have been constrained by their relatively small scale and the specificity of their training data, which covers a limited number of documents and jurisdictions. Our work aims to build on these efforts by deploying LLMs at an unprecedented scale, utilizing models of up to 141B parameters and a base corpus exceeding 500 billion tokens to significantly enhance the depth and breadth of legal language comprehension and generation.

2.2 Legal Domain Adaptation for Modern LLM

The field of legal domain adaptation has traditionally concentrated on refining models through pretraining on specialized corpora [15, 18, 21]. Yet, in the current paradigm, pretraining represents just one aspect of the solution, as LLMs often utilize techniques like instruction fine-tuning and alignment, employing algorithms such as DPO [63], PPO [68] or RLHF [57, 44].

Recent domain-adapted models, such as SaulLM or Legal-FLAN-T5 (a closed model), have tried to improve alignment with legal instructions. However, SaulLM is a smaller model, and Legal-FLAN-T5, is based on an outdated architecture and does not leverage the extreme scale pretraining that contemporary models do. Moreover, it not being publicly available stymies progress vital for advancing research and applications in the legal sector.

[1] Model will be made available at `https://huggingface.co/`.

We believe this work pioneers a holistic approach to domain adaptation by training modern LLMs specifically for the legal domain, from pretraining to instruction fine-tuning and legal preference alignment. We demonstrate that synthetic data can be effectively utilized for alignment, advancing beyond SaulLM-7B's use solely of instruction fine-tuning. The resulting models, `SaulLM-54B` and `SaulLM-141B`, lay the groundwork for further research and development, and expand access to high-performance legal LLMs.

3 Data Collection and Corpus Construction

This section outlines our approach to assembling and refining a comprehensive legal text corpus tailored for training large language models in the legal domain.

3.1 Pretraining Corpora

The diversity of legal systems worldwide, from common law to civil law traditions, presents unique challenges and opportunities [54, 35]. To address this, we compiled an extensive English-language corpus from various jurisdictions including the U.S., Europe, Australia, and others [3, 33], which comprises 500 billion tokens before cleaning and deduplication.

3.1.1 Legal Sources

Our base corpus combines various legal datasets [53] with newly sourced public domain documents. It includes significant collections such as the FreeLaw subset and the MultiLegal Pile, augmented with extensive web-scraped content. Table 1 summarizes the composition and scale of our dataset.

3.1.2 Other Sources

Replay Sources. To mitigate the risk of catastrophic forgetting during model training, we reintroduced data from earlier training distributions [48, 16, 73, 72, 25, 24]. This replay strategy incorporates general data sources such as Wikipedia, StackExchange, and GitHub, and makes up approximately 2% of the total training mix. These datasets are sampled from SlimPajama [69, 19, 70].

Additionally, we included 5% of math datasets in the pretraining mix using commercially available math sources. We found this approach usefull for retaining the reasoning performance of the final model and avoiding the weaker performance observed in previous research attempts like SaulLM-7B[2].

Table 1: **Sources of Legal Pretraining Data**

Source Name	Tokens (B)
FreeLaw Subset from The Pile	15
EDGAR Database	5
English MultiLegal Pile	50
English EuroParl	6
GovInfo Statutes, Opinions & Codes	11
Law Stack Exchange	0.019
Comm Open Australian Legal Corpus	0.5
EU Legislation	0.315
UK Legislation	0.190
Court Transcripts	0.350
UPSTO Database	4.7
Web Data (legal)	400
Other	30
Total	520

Instruction Sources. We found that incorporating conversational data during pretraining is advantageous, drawing inspiration from recent breakthroughs in neural machine translation [6]. Studies suggest that the enhanced translation capabilities of LLMs can be attributed to the presence of accidental parallel data within their training corpora. Accordingly, we have integrated the Super Natural Instruction [83] and FLAN collection [46] into our pretraining mix, enriching the dataset with diverse instructional content.

Data for Model Annealing. Model annealing is primarily achieved through a methodical reduction of the learning rate [58, 38], known as learning rate annealing.

In our experiments, model annealing with high-quality, domain-relevant data significantly enhanced performance. Conversely, repetitive synthetic data from initial instruction fine-tuning harmed performance. Therefore, we used the commercial portion of the LawInstruct dataset for model annealing,

[2]These findings also align with the high percentage of math and STEM in the pretraining mix from [56].

https://doi.org/10.52202/079017-4120

which proved more effective than for instruction finetuning. We also included UltraChat [26] as generic instructions during the annealing phase.

3.1.3 Data Preprocessing

Our data processing pipeline closely follows [18]. In particular, we do:
1. **Text extraction**: a significant fraction of the collected data is in PDF format. We used Poppler to extract the text.
2. **Data cleaning**: extraction from PDF files creates some artifacts like page and line numbers in the middle of sentences, as well as broken lines of text, non-normalized Unicode characters, etc.

- *Text normalization*. We normalize all text using the NFKC method, available through the `unicodedata` Python package.
- *Rule-based filters*. We created regex rules for filtering commonly undesirable but commonly recurring patterns, like page and line numbers in the middle of the text, HTML tags, etc. Following [18], we found that some of the most common 10-grams in our dataset were repeated characters and whitespace and removed them.
- *Perplexity filtering*. Similarly to [18] we used a KenLM [34] model trained on a small subset of carefully cleaned legal data to filter documents with high perplexity. Concretely, we filtered any document whose normalized perplexity was larger than 1500.

3. **Text deduplication**: we used [50] to remove duplicates and near-duplicate examples from the training set. We used default parameters except for the similarity threshold, which we set to 0.5.

Finally, we packed the individual documents together to build 8192 tokens-long training examples. Documents longer than this value were chunked into several examples.

3.2 Instruction Data

Instruction fine-tuning is essential for making an LLM follow instructions and optimize the performance of pretrained models across a variety of tasks [81, 84, 17, 29, 26, 82]. To this end, we employ a strategic mix of general and domain-specific (legal) instructions, aimed at enhancing the model's ability to precisely interpret and execute commands, with a particular focus on legal scenarios.
General Instructions Our methodology for sourcing general instructions involves the integration of a diverse array of datasets, each selected to augment different aspects of the model's capabilities across various domains [14, 92]:
1. *General Instruction from UltraInteract [88]:* UltraInteract is an extensive, high-quality dataset designed to foster complex reasoning, featuring structured instructions that include preference trees, reasoning chains, and multi-turn interaction trajectories.
2. *General Instruction from Dolphin* [3]*:* This dataset provides additional conversational data, further broadening the model's exposure to diverse communication styles and contexts.

Each dataset is subjected to rigorous filtering, deduplication, and curation processes, culminating in a refined compilation of approximately $1,000,000$ instructions meticulously prepared for the instruction fine-tuning phase.
Legal Instruction Construction For legal instructions, we synthesize dialogues and question/answer pairs that capture key legal concepts and document types to emulate legal analysis. In accordance with the model scale, we used `Mistral-54B-Instruct` for `SaulLM-54B` and `Mistral-141B-Instruct` for `SaulLM-141B`. The generation follows the recipe from [18] and begins with a three-turn sequence: (1) a user inquires about a legal document, (2) the assistant reformulates this inquiry by integrating metadata such as document type or issue date, and (3) the user asks for further explanation of the assistant's reasoning. The dialogue progressively deepens, with the assistant methodically unpacking the legal reasoning in response to increasingly nuanced questions from the user.

3.3 Preference Data

We enhance our models' adaptability and precision by incorporating preference data from both general and legal-specific sources [78, 62, 52, 80]. *General datasets* are UltraFeedback [20] and Orca.

[3] https://huggingface.co/datasets/cognitivecomputations/dolphin

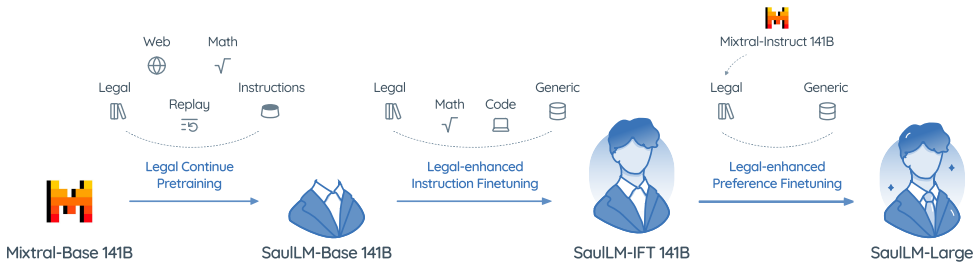

Figure 1: Domain adaptation method model for turning a Mixtral to a `SaulLM-141B`. Training involves different stages: legal domain pretraining, instruction filtering, and preference filtering.

For the *legal domain*, we employ synthetic scenarios crafted to simulate complex legal reasoning and generate accepted/rejected responses. The `Mixtral-142B-Instruct` model evaluates these responses based on factual accuracy, relevance, and logical coherence, selecting the most appropriate responses as preferred outcomes (similar to [91]).

4 Implementation Details & Evaluation Protocol

4.1 Model Selection

We used Mixtral models [40], which are built on a Transformer architecture [79] enhanced with a Mixture of Experts to improve computational efficiency and adaptability for handling extensive contexts. The `Mixtral-54B` and `Mixtral-141B` architecture respectively consists of 32 (resp. 56) layers, a model dimension of 4096 (resp. 6144), and a hidden dimension of $14,336$ (resp. 16384). Although it supports a context length of up to $32,768$ (resp. 65536) tokens, we continue pretraining on $8,192$ tokens. Extending the context length is beyond the scope of this paper. The MoE layers in Mixtral rely on 8 experts with 2 active experts selectively based on the input, efficiently utilizing computational resources and providing significant model capacity. Interestingly, Mixtral is the only model available in dual configurations (`Mixtral-54B` and `Mixtral-141B`), allowing us to evaluate the scalability of our domain adaptation approaches.

At the time of the training, Mixtral was the most powerful decoder in its class, surpassing all competitors including Llama [76, 77, 89], Yi, Qwen [7], and CroissantLLM [28] in terms of both cost-effectiveness and performance.

4.2 Engineering Details

Codebase Configuration Our training framework uses PyTorch. The integration of DeepSpeed [64] (level 3) and Flash attention [23] mechanisms enhances our training efficiency and scalability. We make our models available through the Huggingface hub [85].
Compute Infrastructure The computational backbone for the continuous pretraining phase of our project consists of 384 AMD MI250 GPUs. We can reach 40% GPU utilization with our implementation. For instruction fine-tuning and preference optimization, we rely on 64 AMD MI250 GPUs. Evaluation protocols are executed on a single node of AMD MI250 GPU.
Synthetic Data Generation For synthetic data generation, we used vLLM on a node of NVIDIA-A100, primarily due to limited support of libraries on MI250[4].

4.3 Training Details

The model training process is divided into three distinct phases: continued pretraining, instruction finetuning (IFT), and preference alignment using domain-specific optimization (DPO). A full schema of the pipeline can be found in Figure 1.
Continued Pretraining For continued pretraining, we use the AdamW [41, 47, 8] optimizer with

[4]VLLM is not supported on AMD-MI250 and HuggingFace's text-generation-inference had a few bugs that prevented its use.

https://doi.org/10.52202/079017-4120

hyperparameters $\beta_1 = 0.99$, $\beta_2 = 0.90$, and a learning rate of 2×10^{-5}. We utilize a cross-entropy loss function to optimize model predictions. The training protocol sets gradient accumulation to 4, with tailored batch sizes of 8 for `SaulLM-54B` and 4 for `SaulLM-141B`, optimizing both GPU utilization and training efficiency.

Instruction Fine-Tuning (IFT) IFT uses the AdamW optimizer (learning rate of 1×10^{-5}), reinitialized to reset training states and maintain training stability. We limit this phase to a single training epoch, as our experiments suggest this maximizes performance gains.

Preference Training Using DPO We adjust the learning rate of the AdamW optimizer to 1×10^{-6} during DPO. Our choice of DPO over IPO [13], KTO [27] or ORPO [37] was based on preliminary experiments.

4.4 Evaluation Protocol

LegalBench-Instruct We rely on `LegalBench-Instruct` [18], which refined the prompts from `LegalBench` [31] by eliminating distracting elements and specifying a response format to enhance precision. Like LegalBench, it evaluates LLMs across six types of legal reasoning: issue-spotting, rule-recall, rule-application, rule-conclusion, interpretation, and rhetorical understanding. Grounded in American legal frameworks but applicable globally, these categories provide a comprehensive evaluation of the models' legal reasoning capabilities. This structured approach helps in accurately assessing and guiding the enhancement of LLMs within and beyond American legal contexts. We follow previous work and rely on balanced accuracy as the primary metric across all tasks.

Massive Multitask Language Understanding (MMLU) Previous works utilize MMLU [36], a widely-recognized benchmark, focusing on its legal-specific tasks in *international law*, *professional law*, and *jurisprudence*, with 120, 1500, and 110 examples respectively. These tasks are crucial for assessing our models' understanding and application of complex legal concepts, highlighting their proficiency in nuanced legal environments.

Choice of Baseline & Model Naming For our evaluation, we aim for a direct, apples-to-apples comparison of models. It is important to note that not all competing models are open source, and detailed information on their alignment procedures and instruction fine-tuning processes is not available. This lack of transparency complicates the establishment of fully equitable baseline comparisons. In what follows, we use OpenAI's `GPT-4` as of 10 May 2024, Meta's `Llama3` (the Instruct variant) and the Instruct variants of `Mixtral-54B`, and `Mixtral-141B`. Additionally, `SaulLM-54B-IFT` is the IFT version built on `SaulLM-54B-base` and `SaulLM-medium` for the DPO version based on `SaulLM-54B-IFT`. `SaulLM-large` is the final version DPO and IFT based on `SaulLM-141B`.

5 Experimental Results

5.1 Global Results

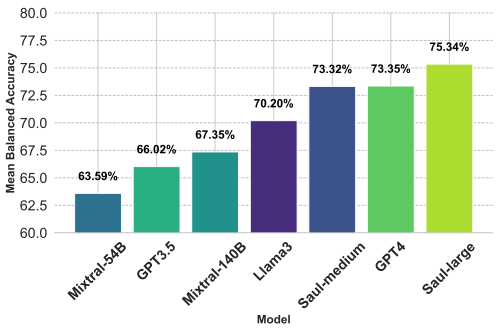

Figure 2: **Overall Results.** Comparison of SaulLM-large and SaulLM-medium with existing models.

Figure 2 presents the results of `SaulLM-large` and `SaulLM-medium` on `LegalBench-Instruct`, from which we make several observations:

Our domain adaptation strategy is achieving strong results. `SaulLM-medium` outperforms `Mixtral-54B`, and similar findings are observed with `SaulLM-large` compared to `Mixtral-141B`. Interestingly, domain adaptation at both the instruction tuning stage and preference alignment enables our smaller models to outperform larger ones, such as `GPT-4` and `LLama3-70B`. These results validate our approach and demonstrate that specializing the entire pipeline (*i.e.*, from continued pretraining to preference alignment) is a promising direction for improving performance in legal-related tasks.

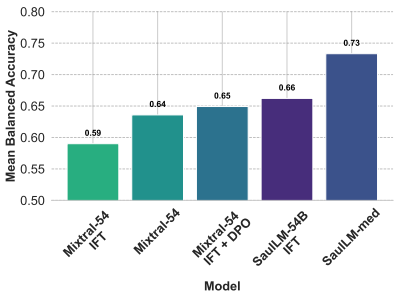

Figure 3: **Global Analysis.** Role of continued pretraining.

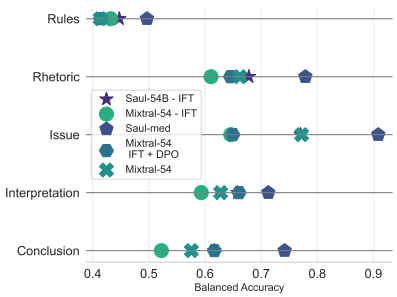

Figure 4: **Category Analysis:** Role of continue pretraining.

Domain adaptation works across scales and MoE models. The results from `SaulLM-medium` and `SaulLM-large` confirm previous findings from [18] and confirm that domain adaptation is effective across different scales, including on MoE models. Interestingly, most of the data collected for this work comes from public sources, which were likely seen during the pretraining of the base models.

A Path Towards Stronger Models. The results of LLama3-70B and the scalability of our methods suggest that applying the same approach to the LLama3-70B base model could lead to even better performance than our best model, `SaulLM-141B`. It is worth noting that `SaulLM-141B` has only 44B active parameters making it appealing for efficient serving.

Table 2: **Quantifying the role of DPO.** We report the percentage of tasks where the difference in performance (Δ) between `SaulLM-54B` and `SaulLM-54B-IFT` is positive (resp. negative).

Category	$\Delta \geq 0$	$\Delta \leq 0$
Conclusion	37.5%	62.5%
Interpretation	65.1%	34.9%
Rhetoric	44.4%	55.6%
Rules	60.0%	40.0%
Issue	100.0%	-

Table 3: **Quantifying the role of scaling.** We report the percentage of tasks where the difference in performance (Δ) between `SaulLM-medium` and `SaulLM-large` is positive (resp. negative).

Category	$\Delta \geq 0$	$\Delta \leq 0$
Conclusion	18.2%	81.8%
Interpretation	23.7%	76.3%
Rules	25.0%	75.0%
Issue	-	100.0%
Rhetoric	-	100.0%

5.2 How much does continued pretraining help for the legal domain?

Previous works on domain adaptation via continued pretraining primarily focused on instruction finetuning [16, 18, 55]. In Figure 3 and Figure 4, we report the performance of `Mixtral-54B` trained with the IFT mix described in subsection 3.2 (`Mixtral-54-IFT`) and subsequently aligned using the DPO dataset (`Mixtral-54-IFT+DPO`), as described in subsection 3.3. We also compare these results to the instruct version of Mixtral (`Mixtral-54B`), as outlined in [40].

Continuing pretraining significantly enhances model performance in the legal domain, benefiting both the IFT and DPO stages. From Figure 3, we observe that both IFT and DPO benefit from a notable improvement (approximately $+7\%$). Interestingly, this improvement is consistent across all five categories, as shown in Figure 4.

Adding legal data to the IFT and DPO datasets improves the model's legal capabilities. By comparing the performance of `Mixtral-54-IFT+DPO` and `Mixtral-54`, we observe that the mix used for IFT and DPO enhanced with legal data leads to stronger legal performance than that of `Mixtral-54`, which does not publicly describe the alignment methods used. This result aligns with findings reported in [55, 18].

https://doi.org/10.52202/079017-4120

5.3 How Much Does Legal Preference Alignment Help?

Our findings from Figure 3 indicate that alignment significantly improves the results. In particular, **DPO improvement is mostly consistent across tasks and categories.** As shown in Table 2, the alignment version `SaulLM-medium` demonstrates significant improvements over the IFT version across most tasks, including conclusion, rhetoric, rules, and issue tasks. We observe, however, a drop in performance in some interpretation tasks. Upon closer examination, we found that this decline is often due to the model becoming more verbose, which causes the evaluation process to fail in correctly parsing the answers , i.e., this issue primarily arises from a benchmark limitation. Addressing model verbosity and the challenge of more reliable benchmarks is beyond the scope of this work, but it is a well-known problem identified in many concurrent studies [29]. Enhancing the evaluation process is one of the key improvements we plan to contribute to in the future.

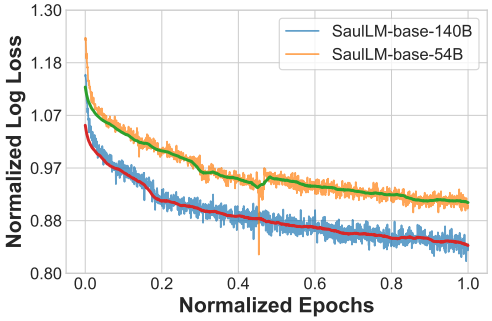

Figure 5: **Continue Pretraining Analysis.** Training loss for `SaulLM-141B-base` and `SaulLM-54B-base` over normalized epochs.

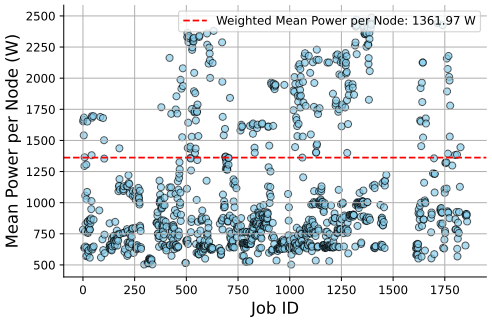

Figure 6: **Energy Consumption Analysis.** Mean Power per Node for training jobs on the ADASTRA Supercomputer.

5.4 Can We Achieve Further Improvements by Continuing Pretraining?

Training longer can potentially improve the results. Figure 5 illustrates the normalized log loss over normalized epochs for both model sizes, `SaulLM-54B-base` and `SaulLM-141B-base`. The figure presents both the raw and smoothed loss curves, which exhibit a clear downward trend throughout the training period, with no indication of saturation.

This observation suggests that continuing the pretraining process beyond the current `SaulLM-base` can lead to further improvements. The consistent decrease in loss implies that the models have not yet reached their full potential and that additional pretraining could enhance their performance further, which is consistent with findings from other works [60, 2, 51].

5.5 How Much Does Scaling Help?

Table 3 quantifies the impact of scaling the model and compares the performance between `SaulLM-medium` and `SaulLM-large`.

The main takeaway is that **scaling generally improves overall results, but we also observe inverse scaling on some legal tasks** [49, 43]. Unsurprisingly, for the majority of tasks across all categories, increasing the model size leads to improvements, but for tasks involving conclusion, interpretation, and rules, we observe a proportion of tasks (20%) that follow inverse scaling laws.

5.6 Energy Consumption

The training was conducted on Adastra, ranked 3rd in the Green500 since November 2022[5], as one of the world's most efficient machines in terms of performance per watt.

Experiments for training SaulLM were performed between February 20th and May 15th. Energy consumption for each job was meticulously tracked, and we calculated and displayed the average

[5] `https://top500.org/lists/green500/2023/11/`

power used per node for each job involved in this training in Figure 6[6]. The mean power usage ranged from 600W to 2500W, reflecting the varying utilization of the GPUs. Each node contains four MI250X GPUs, which have a theoretical Thermal Design Power (TDP) of 560W. This configuration explains the maximum consumption of 2500W during high-intensity GPU usage.

Overall, the project involves over $160,000$ hours of MI250 for debugging, continued pretraining, instruction finetuning and preference alignment. The total energy consumed was $65,480.4$kWh. Consumption is significantly lower than the typical energy requirements for full LLM training, showing that continued pretraining is an effective strategy for specializing in new LLMs while optimizing energy efficiency.

6 Conclusion & Limitations

6.1 Conclusion

This study released two new legal LLMs under the MIT license: `SaulLM-54B` and `SaulLM-141B`. They leverage the Mixtral architecture and continued pretraining on a large legal corpus. Our findings show significant advancements in processing and understanding complex legal documents. Through continued pretraining, instruction fine-tuning, and preference alignment using domain-specific optimization, we have demonstrated substantial improvements compared to `GPT-4`, `Llama3` and original Mixtral models as measured on `LegalBench-Instruct`.

6.2 Limitations

Our experiments suggest that the instruction finetuning and alignment processes utilized by `Mixtral-Instruct` and `Llama3` are advanced and challenging to replicate. These processes often rely on proprietary datasets and significant computational resources that are not readily available in open-source frameworks. Although both `SaulLM-54B` and `SaulLM-141B` achieve stronger performances than `Llama3` and `Mixtral Instruct` on legal benchmarks, we found that they are slightly weaker at following generic instructions.

Looking forward, we aim to continue our work on enhancing the `SaulLM` family, particularly focusing on integrating `Llama3` and improving the alignment procedure. Our goal is to improve the alignment of these models with legal tasks, refining their ability to process and understand legal language with even greater accuracy and relevance. This future work will strive to address the current limitations by developing more robust methods for instruction finetuning and alignment that are accessible to the broader research community.

Acknowledgments and Disclosure of Funding

This research was supported by computing grants from Adastra and Jeanzay. We extend our special thanks to Michael Robert, head of CINES, for his invaluable support and confidence in our work.

Our models have been trained on ADASTRA, with minor experimentation conducted on Jeanzay. The utilization of HPC resources was made possible through the Jeanzay grants 101838, 103256, and 103298, as well as the Adastra grants C1615122, CAD14770, and CAD15031.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*, 2024.

[6]The high number of jobs (over 1800) for this project arise from (i) the technical difficulties in efficiently using AMD GPUs, and (ii) numerous hardware failures when scaling up the number of nodes.

 https://doi.org/10.52202/079017-4120

[3] Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ computer science*, 2:e93, 2016.

[4] Loubna Ben Allal, Raymond Li, Denis Kocetkov, Chenghao Mou, Christopher Akiki, Carlos Munoz Ferrandis, Niklas Muennighoff, Mayank Mishra, Alex Gu, Manan Dey, et al. Santacoder: don't reach for the stars! *arXiv preprint arXiv:2301.03988*, 2023.

[5] Duarte M Alves, Nuno M Guerreiro, João Alves, José Pombal, Ricardo Rei, José GC de Souza, Pierre Colombo, and André FT Martins. Steering large language models for machine translation with finetuning and in-context learning. *arXiv preprint arXiv:2310.13448*, 2023.

[6] Duarte M. Alves, José Pombal, Nuno M. Guerreiro, Pedro H. Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and André F. T. Martins. Tower: An open multilingual large language model for translation-related tasks. 2024.

[7] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023.

[8] Anas Barakat and Pascal Bianchi. Convergence and dynamical behavior of the adam algorithm for nonconvex stochastic optimization. *SIAM Journal on Optimization*, 31(1):244–274, 2021.

[9] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[10] Rohan Bhambhoria, Samuel Dahan, Jonathan Li, and Xiaodan Zhu. Evaluating ai for law: Bridging the gap with open-source solutions. *arXiv preprint arXiv:2404.12349*, 2024.

[11] Elliot Bolton, Abhinav Venigalla, Michihiro Yasunaga, David Hall, Betty Xiong, Tony Lee, Roxana Daneshjou, Jonathan Frankle, Percy Liang, Michael Carbin, et al. Biomedlm: A 2.7 b parameter language model trained on biomedical text. *arXiv preprint arXiv:2403.18421*, 2024.

[12] Elliot Bolton, Betty Xiong, Vijaytha Muralidharan, Joel Schamroth, Vivek Muralidharan, Christopher D Manning, and Roxana Daneshjou. Assessing the potential of mid-sized language models for clinical qa. *arXiv preprint arXiv:2404.15894*, 2024.

[13] Daniele Calandriello, Daniel Guo, Remi Munos, Mark Rowland, Yunhao Tang, Bernardo Avila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, et al. Human alignment of large language models through online preference optimisation. *arXiv preprint arXiv:2403.08635*, 2024.

[14] Yihan Cao, Yanbin Kang, and Lichao Sun. Instruction mining: High-quality instruction data selection for large language models. *arXiv preprint arXiv:2307.06290*, 2023.

[15] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*, 2020.

[16] Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, et al. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*, 2023.

[17] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

[18] Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre F. T. Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and Michael Desa. Saullm-7b: A pioneering large language model for law. 2024.

[19] Together Computer. Redpajama: an open dataset for training large language models, 2023.

[20] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*, 2023.

[21] Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *arXiv preprint arXiv:2306.16092*, 2023.

[22] Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

[23] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

[24] Maxime Darrin, Pablo Piantanida, and Pierre Colombo. Rainproof: An umbrella to shield text generators from out-of-distribution data. *arXiv preprint arXiv:2212.09171*, 2022.

[25] Maxime Darrin, Guillaume Staerman, Eduardo Dadalto Câmara Gomes, Jackie CK Cheung, Pablo Piantanida, and Pierre Colombo. Unsupervised layer-wise score aggregation for textual ood detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17880–17888, 2024.

[26] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

[27] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

[28] Manuel Faysse, Patrick Fernandes, Nuno Guerreiro, António Loison, Duarte Alves, Caio Corro, Nicolas Boizard, João Alves, Ricardo Rei, Pedro Martins, et al. Croissantllm: A truly bilingual french-english language model. *arXiv preprint arXiv:2402.00786*, 2024.

[29] Manuel Faysse, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Revisiting instruction fine-tuned model evaluation to guide industrial applications. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2023.

[30] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling. 2020.

[31] Neel Guha, Daniel E Ho, Julian Nyarko, and Christopher Ré. Legalbench: Prototyping a collaborative benchmark for legal reasoning. *arXiv preprint arXiv:2209.06120*, 2022.

[32] Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y Wu, YK Li, et al. Deepseek-coder: When the large language model meets programming–the rise of code intelligence. *arXiv preprint arXiv:2401.14196*, 2024.

[33] Asier Gutiérrez-Fandiño, Jordi Armengol-Estapé, Aitor Gonzalez-Agirre, and Marta Villegas. Spanish legalese language model and corpora. *arXiv preprint arXiv:2110.12201*, 2021.

[34] Kenneth Heafield. KenLM: Faster and smaller language model queries. In Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar F. Zaidan, editors, *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, July 2011. Association for Computational Linguistics.

 https://doi.org/10.52202/079017-4120

[35] Peter Henderson, Mark Krass, Lucia Zheng, Neel Guha, Christopher D Manning, Dan Jurafsky, and Daniel Ho. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset. *Advances in Neural Information Processing Systems*, 35:29217–29234, 2022.

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

[37] Jiwoo Hong, Noah Lee, and James Thorne. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*, 2024.

[38] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

[39] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[40] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

[41] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[42] Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*, 2023.

[43] Xianhang Li, Zeyu Wang, and Cihang Xie. An inverse scaling law for clip training. *Advances in Neural Information Processing Systems*, 36, 2024.

[44] Zihao Li, Zhuoran Yang, and Mengdi Wang. Reinforcement learning with human feedback: Learning dynamic choices via pessimism. *arXiv preprint arXiv:2305.18438*, 2023.

[45] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024.

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

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

[48] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press, 1989.

[49] Ian R McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, et al. Inverse scaling: When bigger isn't better. *arXiv preprint arXiv:2306.09479*, 2023.

[50] Chenghao Mou, Chris Ha, Kenneth Enevoldsen, and Peiyuan Liu. Chenghaomou/text-dedup: Reference snapshot, September 2023.

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

[52] Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, et al. Nash learning from human feedback. *arXiv preprint arXiv:2312.00886*, 2023.

[53] Joel Niklaus and Daniele Giofré. Budgetlongformer: Can we cheaply pretrain a sota legal language model from scratch? *arXiv preprint arXiv:2211.17135*, 2022.

[54] Joel Niklaus, Veton Matoshi, Matthias Stürmer, Ilias Chalkidis, and Daniel E. Ho. Multilegalpile: A 689gb multilingual legal corpus. 2023.

[55] Joel Niklaus, Lucia Zheng, Arya D McCarthy, Christopher Hahn, Brian M Rosen, Peter Henderson, Daniel E Ho, Garrett Honke, Percy Liang, and Christopher Manning. Flawn-t5: An empirical examination of effective instruction-tuning data mixtures for legal reasoning. *arXiv preprint arXiv:2404.02127*, 2024.

[56] Aitor Ormazabal, Che Zheng, Cyprien de Masson d'Autume, Dani Yogatama, Deyu Fu, Donovan Ong, Eric Chen, Eugenie Lamprecht, Hai Pham, Isaac Ong, et al. Reka core, flash, and edge: A series of powerful multimodal language models. *arXiv preprint arXiv:2404.12387*, 2024.

[57] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[58] Jupinder Parmar, Shrimai Prabhumoye, Joseph Jennings, Mostofa Patwary, Sandeep Subramanian, Dan Su, Chen Zhu, Deepak Narayanan, Aastha Jhunjhunwala, Ayush Dattagupta, et al. Nemotron-4 15b technical report. *arXiv preprint arXiv:2402.16819*, 2024.

[59] Shounak Paul, Arpan Mandal, Pawan Goyal, and Saptarshi Ghosh. Pre-trained language models for the legal domain: A case study on indian law. In *Proceedings of 19th International Conference on Artificial Intelligence and Law - ICAIL 2023*, 2023.

[60] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*, 2023.

[61] Maciej Pióro, Kamil Ciebiera, Krystian Król, Jan Ludziejewski, and Sebastian Jaszczur. Moe-mamba: Efficient selective state space models with mixture of experts. *arXiv preprint arXiv:2401.04081*, 2024.

[62] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

[63] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[64] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506, 2020.

[65] Richard M Re and Alicia Solow-Niederman. Developing artificially intelligent justice. *Stan. Tech. L. Rev.*, 22:242, 2019.

 https://doi.org/10.52202/079017-4120

[66] Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

[67] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

[68] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[69] Zhiqiang Shen, Tianhua Tao, Liqun Ma, Willie Neiswanger, Joel Hestness, Natalia Vassilieva, Daria Soboleva, and Eric Xing. Slimpajama-dc: Understanding data combinations for llm training. *arXiv preprint arXiv:2309.10818*, 2023.

[70] Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. Slimpajama: A 627b token cleaned and deduplicated version of redpajama, 2023.

[71] Tania Sourdin, Bin Li, and Donna Marie McNamara. Court innovations and access to justice in times of crisis. *Health policy and technology*, 9(4):447–453, 2020.

[72] Guillaume Staerman, Pavlo Mozharovskyi, Pierre Colombo, Stéphan Clémençon, and Florence d'Alché Buc. A pseudo-metric between probability distributions based on depth-trimmed regions. *arXiv preprint arXiv:2103.12711*, 2021.

[73] Jingyuan Sun, Shaonan Wang, Jiajun Zhang, and Chengqing Zong. Distill and replay for continual language learning. In *Proceedings of the 28th international conference on computational linguistics*, pages 3569–3579, 2020.

[74] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

[75] Robert Tinn, Hao Cheng, Yu Gu, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Fine-tuning large neural language models for biomedical natural language processing. *Patterns*, 4(4), 2023.

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

[77] 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. Llama 2: Open foundation and fine-tuned chat models, 2023.

[78] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

[79] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[80] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020.

[81] Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. *arXiv preprint arXiv:2306.04751*, 2023.

[82] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2023.

[83] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*, 2022.

[84] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[85] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[86] Yunzhi Yao, Shaohan Huang, Wenhui Wang, Li Dong, and Furu Wei. Adapt-and-distill: Developing small, fast and effective pretrained language models for domains. *arXiv preprint arXiv:2106.13474*, 2021.

[87] Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. *arXiv preprint arXiv:2403.11549*, 2024.

[88] Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, Zhenghao Liu, Bowen Zhou, Hao Peng, Zhiyuan Liu, and Maosong Sun. Advancing llm reasoning generalists with preference trees, 2024.

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

[90] Rongyu Zhang, Yulin Luo, Jiaming Liu, Huanrui Yang, Zhen Dong, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, Yuan Du, et al. Efficient deweahter mixture-of-experts with uncertainty-aware feature-wise linear modulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16812–16820, 2024.

[91] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

[92] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*, 2023.

[93] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

 https://doi.org/10.52202/079017-4120

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer:

   Justification: We propose a new domain adaptation recipe.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer:

   Justification: See limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer:

Justification: No theoretical paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer:

   Justification: Dataset and model are available with a list of hyperparameter in training details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

 https://doi.org/10.52202/079017-4120

Answer:

Justification: Models are released under a commercial license and data are described.

Guidelines:

- The answer NA means that the paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer:

   Justification: We relied on previous works that are cited.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer:

   Justification: We follow previous works.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer:

   Justification: See model details and energy analysis.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer:

   Justification: We fully comply.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer:

    Justification: See limitation.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

https://doi.org/10.52202/079017-4120

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer:

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer:

    Justification: MIT-liscence

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer:

Justification: We introduce new models that will be publically released under MIT liscence.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer:

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer:

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

A Energy Analysis

Figure 7, Figure 9, Figure 8, and Figure 10 show different analysis of energy consumption while developing and training `SaulLM-54B` and `SaulLM-141B`.

https://doi.org/10.52202/079017-4120

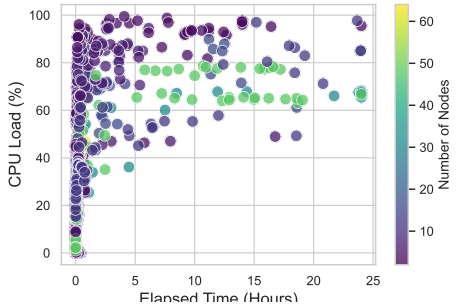

Figure 7: **Energy Analysis.** GPU Load vs Elapsed Time for Different Numbers of Nodes.

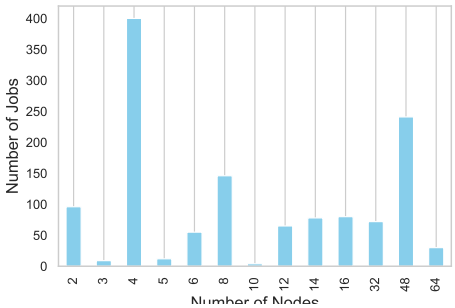

Figure 8: **Energy Analysis.** Number of Jobs vs Number of Nodes.

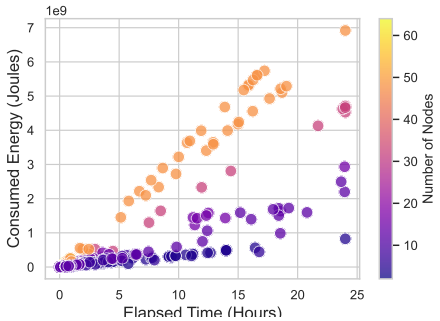

Figure 9: **Energy Analysis.** Consumed Energy vs Elapsed Time for Different Numbers of Nodes.

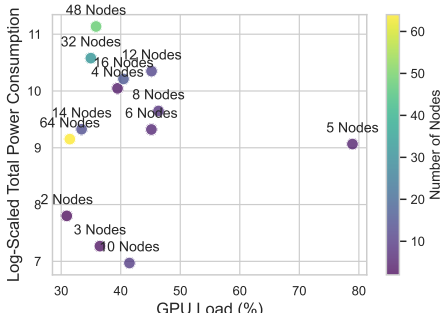

Figure 10: **Energy Analysis.** Log-Scaled Total Power Consumption vs GPU Load for Different Numbers of Nodes.

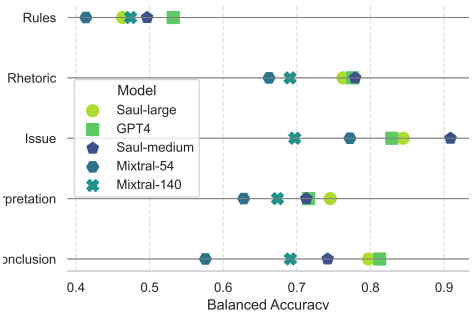

Figure 11: **Category Analysis on** `LegalBench-Instruct`**.**

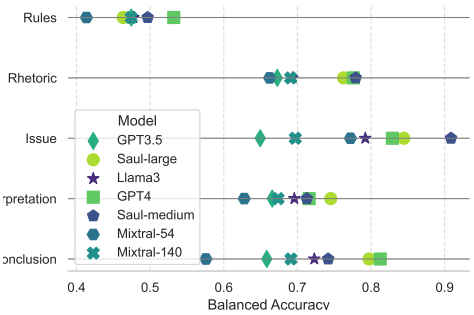

Figure 12: **Category Analysis on** `LegalBench-Instruct`**.**

B Further Results on LegalBench

B.1 Per Category Analysis

B.2 Per Task Analysis

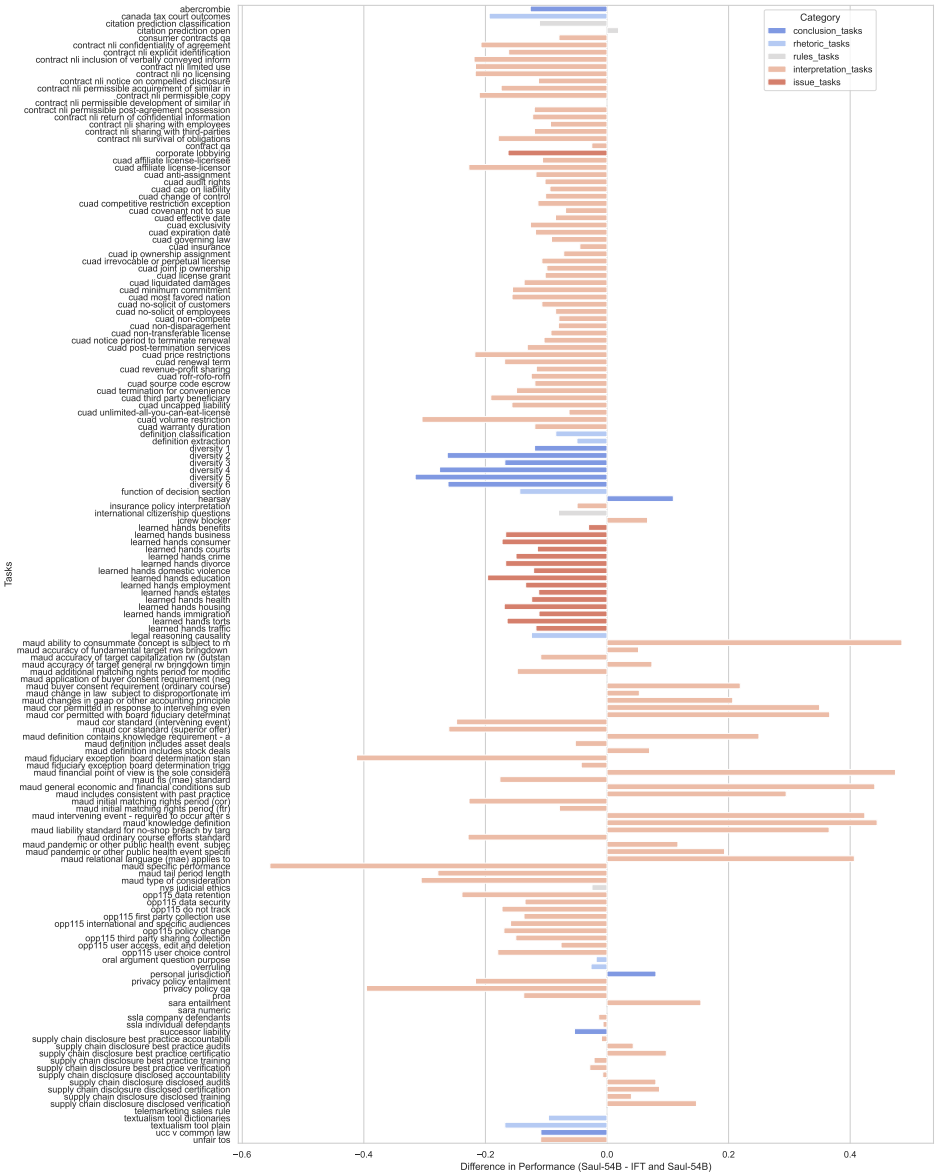

Figure 13: **Per Task Analysis:** Quantifying the role of DPO w.r.t. instruction fine-tuning only.

https://doi.org/10.52202/079017-4120

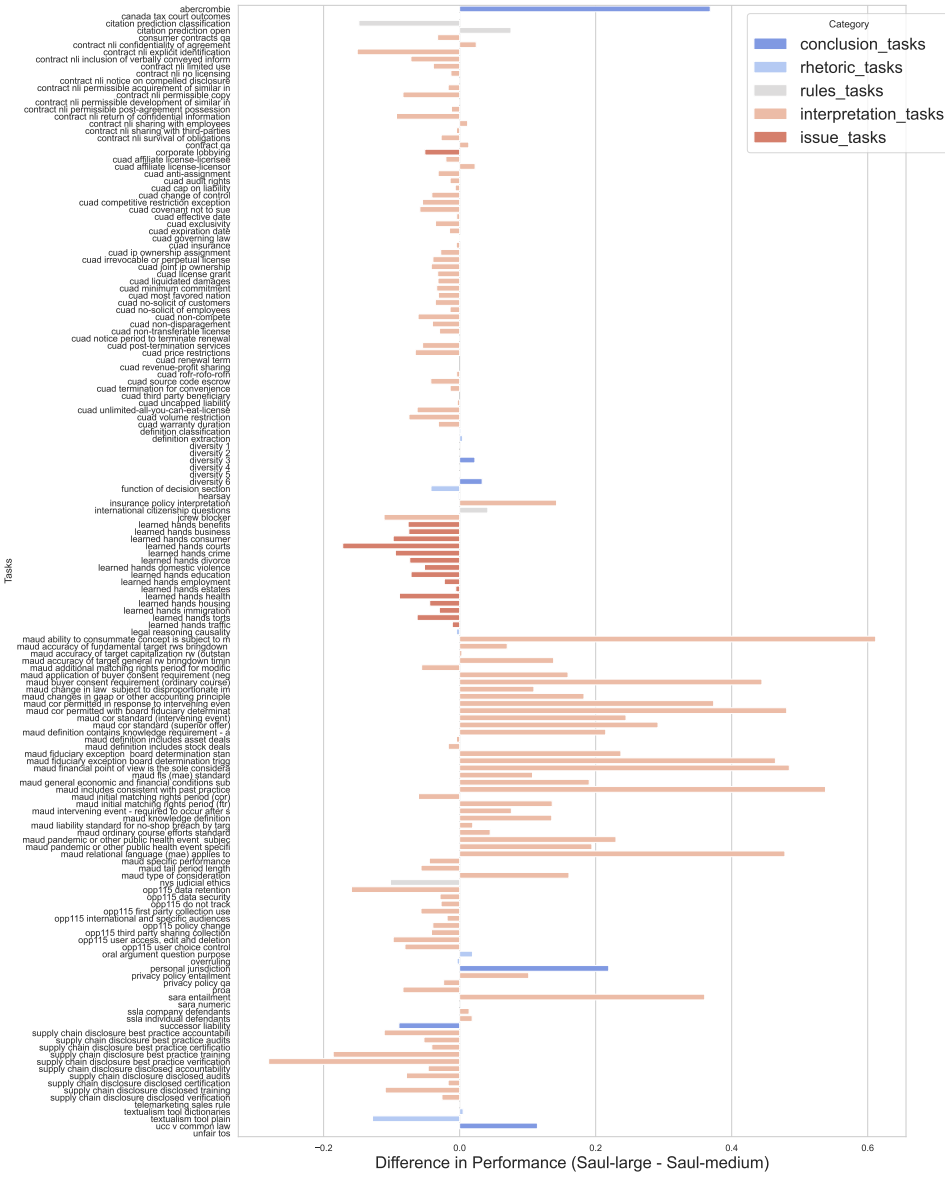

Figure 14: **Per Task Analysis:** Quantifying the role of DPO w.r.t. instruction fine-tuning only.