SOLAR 10.7B: Scaling Large Language Models with Simple yet Effective Depth Up-Scaling

Dahyun Kim[*], **Chanjun Park**[*†], **Sanghoon Kim**[*†], **Wonsung Lee**[*†], **Wonho Song**[*]
Yunsu Kim[*], **Hyeonwoo Kim**[*], **Yungi Kim, Hyeonju Lee, Jihoo Kim**
Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim
Mikyoung Cha, Hwalsuk Lee[†], **Sunghun Kim**[†]

Upstage AI, South Korea

{kdahyun, chanjun.park, limerobot, wonsung.lee, hwalsuk.lee, hunkim}@upstage.ai

Abstract

We introduce SOLAR 10.7B, a large language model (LLM) with 10.7 billion parameters, demonstrating superior performance in various natural language processing (NLP) tasks. Inspired by recent efforts to efficiently up-scale LLMs, we present a method for scaling LLMs called depth up-scaling (DUS), which encompasses depthwise scaling and continued pretraining. In contrast to other LLM up-scaling methods that use mixture-of-experts, DUS does not require complex changes to train and inference efficiently. We show experimentally that DUS is simple yet effective in scaling up high-performance LLMs from small ones. Building on the DUS model, we additionally present SO-LAR 10.7B-Instruct, a variant fine-tuned for instruction-following capabilities, surpassing Mixtral-8x7B-Instruct. SOLAR 10.7B is publicly available under the Apache 2.0 license, promoting broad access and application in the LLM field [1].

1 Introduction

The field of natural language processing (NLP) has been significantly transformed by the introduction of large language models (LLMs), which have enhanced our understanding and interaction with human language (Zhao et al., 2023). These advancements bring challenges such as the increased need to train ever larger models (Rae et al., 2021; Wang et al., 2023; Pan et al., 2023; Lian, 2023; Yao et al., 2023; Gesmundo and Maile, 2023) owing to the performance scaling law (Kaplan et al., 2020; Hernandez et al., 2021; Anil et al., 2023; Kaddour et al., 2023). To efficiently tackle the above, recent works in scaling language models such as a mixture of experts (MoE) (Shazeer et al., 2017; Komatsuzaki et al., 2022) have been proposed. While those approaches are able to effi-

ciently and effectively scale-up LLMs, they often require non-trivial changes to the training and inference framework (Gale et al., 2023), which hinders widespread applicability. Effectively and efficiently scaling up LLMs whilst also retaining the *simplicity* for ease of use is an important problem (Alberts et al., 2023; Fraiwan and Khasawneh, 2023; Sallam et al., 2023; Bahrini et al., 2023).

Inspired by Komatsuzaki et al. (2022), we present depth up-scaling (DUS), an effective and efficient method to up-scale LLMs whilst also remaining straightforward to use. DUS consists of scaling the number of layers in the base model and continually pretraining the scaled model. Unlike (Komatsuzaki et al., 2022), DUS does not scale the model using MoE and rather use a depthwise scaling method analogous to Tan and Le (2019) which is adapted for the LLM architecture. Thus, there are no additional modules or dynamism as with MoE, making DUS immediately compatible with easy-to-use LLM frameworks such as HuggingFace (Wolf et al., 2019) with no changes to the training or inference framework for maximal efficiency. Furthermore, DUS is applicable to all transformer architectures, opening up new gateways to effectively and efficiently scale-up LLMs in a simple manner. Using DUS, we release SO-LAR 10.7B, an LLM with 10.7 billion parameters, that outperforms existing models like Llama 2 (Touvron et al., 2023) and Mistral 7B (Jiang et al., 2023) in various benchmarks.

We have also developed SOLAR 10.7B-Instruct, a variant fine-tuned for tasks requiring strict adherence to complex instructions. It significantly outperforms the Mixtral-8x7B-Instruct model across various evaluation metrics, evidencing an advanced proficiency that exceeds the capabilities of even larger models in terms of benchmark performance.

By releasing SOLAR 10.7B under the Apache 2.0 license, we aim to promote collaboration and innovation in NLP. This open-source approach allows

---

[*]Equal Contribution [†] Corresponding Author
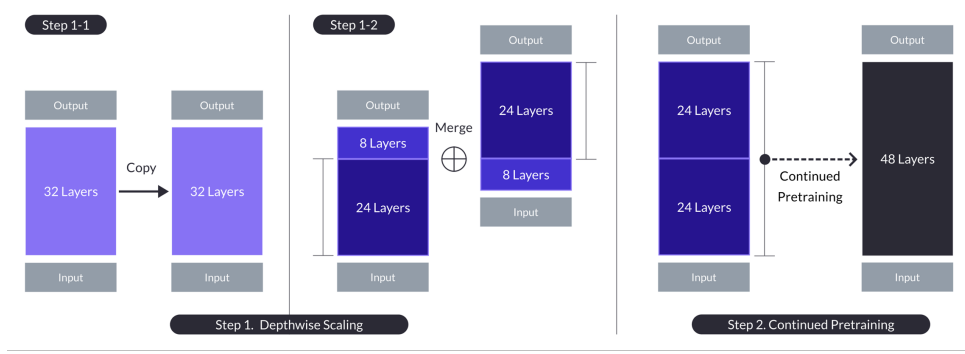[1] https://huggingface.co/upstage/SOLAR-10.7B-v1.0

Figure 1: Depth up-scaling for the case with $n = 32, s = 48$, and $m = 8$. Depth up-scaling is achieved through a dual-stage process of depthwise scaling followed by continued pretraining.

for wider access and application of these models by researchers and developers globally.

2 Depth Up-Scaling

To efficiently scale-up LLMs, we aim to utilize pre-trained weights of base models to scale up to larger LLMs (Komatsuzaki et al., 2022). While existing methods such as Komatsuzaki et al. (2022) use MoE (Shazeer et al., 2017) to scale-up the model architecture, we opt for a different depthwise scaling strategy inspired by Tan and Le (2019). We then continually pretrain the scaled model as just scaling the model without further pretraining degrades the performance.

Base model. Any n-layer transformer architecture can be used but we select the 32-layer Llama 2 architecture as our base model. We initialize the Llama 2 architecture with pretrained weights from Mistral 7B, as it is one of the top performers compatible with the Llama 2 architecture. By adopting the Llama 2 architecture for our base model, we aim to leverage the vast pool of community resources while introducing novel modifications to further enhance its capabilities.

Depthwise scaling. From the base model with n layers, we set the target layer count s for the scaled model, which is largely dictated by the available hardware.

With the above, the depthwise scaling process is as follows. The base model with n layers is duplicated for subsequent modification. Then, we remove the final m layers from the original model and the initial m layers from its duplicate, thus forming two distinct models with $n - m$ layers. These two models are concatenated to form a scaled model with $s = 2 \cdot (n - m)$ layers. Note that $n = 32$ from our base model and we set $s = 48$ considering

our hardware constraints and the efficiency of the scaled model, *i.e.,* fitting between 7 and 13 billion parameters. Naturally, this leads to the removal of $m = 8$ layers. The depthwise scaling process with $n = 32, s = 48$, and $m = 8$ is depicted in 'Step 1: Depthwise Scaling' of Fig. 1.

We note that a method in the community that also scale the model in the same manner [2] as 'Step 1: Depthwise Scaling' of Fig. 1 has been concurrently developed.

Continued pretraining. The performance of the depthwise scaled model initially drops below that of the base LLM. Thus, we additionally apply the continued pretraining step as shown in 'Step 2: Continued Pretraining' of Fig. 1. Experimentally, we observe rapid performance recovery of the scaled model during continued pretraining, a phenomenon also observed in Komatsuzaki et al. (2022). We consider that the particular way of depthwise scaling has isolated the heterogeneity in the scaled model which allowed for this fast performance recovery.

Delving deeper into the heterogeneity of the scaled model, a simpler alternative to depthwise scaling could be to just repeat its layers once more, *i.e.,* from n to $2n$ layers. Then, the 'layer distance', or the difference in the layer indices in the base model, is only bigger than 1 where layers n and $n + 1$ are connected, *i.e.,* at the seam.

However, this results in maximum layer distance at the seam, which may be too significant of a discrepancy for continued pretraining to quickly resolve. Instead, depthwise scaling sacrifices the $2m$ middle layers, thereby reducing the discrepancy at the seam and making it easier for continued

---

[2] https://huggingface.co/Undi95/Mistral-11B-v0.1

| Properties | Training Datasets | | | | | |
| | Instruction | | | Alignment | | |
	Alpaca-GPT4	OpenOrca	Synth. Math-Instruct	Orca DPO Pairs	Ultrafeedback Cleaned	Synth. Math-Alignment
Total # Samples	52K	2.91M	126K	12.9K	60.8K	126K
Maximum # Samples Used	52K	100K	52K	12.9K	60.8K	20.1K
Open Source	O	O	✗	O	O	✗

Table 1: Training datasets used for the instruction and alignment tuning stages, respectively. For the instruction tuning process, we utilized the Alpaca-GPT4 (Peng et al., 2023), OpenOrca (Mukherjee et al., 2023), and Synth. Math-Instruct datasets, while for the alignment tuning, we employed the Orca DPO Pairs (Intel, 2023), Ultrafeedback Cleaned (Cui et al., 2023; Ivison et al., 2023), and Synth. Math-Alignment datasets. The 'Total # Samples' indicates the total number of samples in the entire dataset. The 'Maximum # Samples Used' indicates the actual maximum number of samples that were used in training, which could be lower than the total number of samples in a given dataset. 'Open Source' indicates whether the dataset is open-sourced.

pretraining to quickly recover performance. We attribute the success of DUS to reducing such discrepancies in both the depthwise scaling and the continued pretraining steps. We also hypothesize that other methods of depthwise scaling could also work for DUS, as long as the discrepancy in the scaled model is sufficiently contained before the continued pretraining step.

Comparison to other up-scaling methods. Unlike Komatsuzaki et al. (2022), depthwise scaled models do not require additional modules like gating networks or dynamic expert selection. Consequently, scaled models in DUS do not necessitate a distinct training framework for optimal training efficiency, nor do they require specialized CUDA kernels for fast inference. A DUS model can seamlessly integrate into existing training and inference frameworks while maintaining high efficiency.

3 Training Details

After DUS, including continued pretraining, we perform fine-tuning of SOLAR 10.7B in two stages: 1) instruction tuning and 2) alignment tuning.

Instruction tuning. In the instruction tuning stage, the model is trained to follow instructions in a QA format (Zhang et al., 2023). We mostly use open-source datasets but also synthesize a math QA dataset to enhance the model's mathematical capabilities. A rundown of how we crafted the dataset is as follows. First, seed math data are collected from the Math (Hendrycks et al., 2021) dataset only, to avoid contamination with commonly used benchmark datasets such as GSM8K (Cobbe et al., 2021). Then, using a process similar to MetaMath (Yu et al., 2023), we rephrase the questions and answers of the seed math data. We use the resulting rephrased question-answer pairs as a QA dataset and call it 'Synth. Math-Instruct'.

Alignment tuning. In the alignment tuning stage, the instruction-tuned model is further fine-tuned to be more aligned with human or strong AI (*e.g.,* GPT4 (OpenAI, 2023)) preferences using sDPO (Kim et al., 2024a), an improved version of direct preference optimization (DPO) (Rafailov et al., 2023). Similar to the instruction tuning stage, we use mostly open-source datasets but also synthesize a math-focused alignment dataset utilizing the 'Synth. Math-Instruct' dataset mentioned in the instruction tuning stage.

The alignment data synthesis process is as follows. We take advantage of the fact that the rephrased question-answer pairs in Synth. Math-Instruct data are beneficial in enhancing the model's mathematical capabilities (see Sec. 4.3.1). Thus, we speculate that the rephrased answer to the rephrased question is a better answer than the original answer, possibly due to the interim rephrasing step. Consequently, we set the rephrased question as the prompt and use the rephrased answer as the chosen response and the original answer as the rejected response and create the {prompt, chosen, rejected} DPO tuple. We aggregate the tuples from the rephrased question-answer pairs and call the resulting dataset 'Synth. Math-Alignment'.

4 Results

4.1 Experimental Details

Training datasets. We present details regarding our training datasets for the instruction and alignment tuning stages in Tab. 1. We do not always use the entire dataset and instead subsample a set amount. Note that most of our training data is open-source, and the undisclosed datasets can be substituted for open-source alternatives such as the

Model	Size	Type	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
SOLAR 10.7B-Instruct	∼ 11B	Alignment-tuned	**74.20**	**71.08**	88.16	66.21	**71.43**	83.58	64.75
Qwen 72B	∼ 72B	Pretrained	73.60	65.19	85.94	**77.37**	60.19	82.48	**70.43**
Mixtral 8x7B-Instruct-v0.1	∼ 47B	Instruction-tuned	72.62	70.22	87.63	71.16	64.58	81.37	60.73
Yi 34B-200K	∼ 34B	Pretrained	70.81	65.36	85.58	76.06	53.64	82.56	61.64
Yi 34B	∼ 34B	Pretrained	69.42	64.59	85.69	76.35	56.23	83.03	50.64
Mixtral 8x7B-v0.1	∼ 47B	Pretrained	68.42	66.04	86.49	71.82	46.78	81.93	57.47
Llama 2 70B	∼ 70B	Pretrained	67.87	67.32	87.33	69.83	44.92	83.74	54.06
Falcon 180B	∼ 180B	Pretrained	67.85	69.45	**88.86**	70.50	45.47	**86.90**	45.94
SOLAR 10.7B	∼ 11B	Pretrained	66.04	61.95	84.60	65.48	45.04	83.66	55.50
Qwen 14B	∼ 14B	Pretrained	65.86	58.28	83.99	67.70	49.43	76.80	58.98
Mistral 7B-Instruct-v0.2	∼ 7B	Instruction-tuned	65.71	63.14	84.88	60.78	68.26	77.19	40.03
Yi 34B-Chat	∼ 34B	Instruction-tuned	65.32	65.44	84.16	74.90	55.37	80.11	31.92
Mistral 7B	∼ 7B	Pretrained	60.97	59.98	83.31	64.16	42.15	78.37	37.83

Table 2: Evaluation results in the Open LLM Leaderboard for SOLAR 10.7B and SOLAR 10.7B-Instruct along with other top-performing models. We report the scores for the six tasks mentioned in Sec. 4.1 along with the H6 score (average of six tasks). We also report the size of the models in units of billions of parameters. The type indicates the training stage of the model and is chosen from {Pretrained, Instruction-tuned, Alignment-tuned}. Models based on SOLAR 10.7B are colored purple. The best scores for H6 and the individual tasks are shown in bold.

MetaMathQA (Yu et al., 2023) dataset.

We reformatted the instruction datasets with an Alpaca-styled chat template. For datasets such as OpenOrca, which are derived from FLAN (Longpre et al., 2023), we filter data that overlaps with the benchmark datasets (see Tab. 8 in Appendix. C for more information). The alignment datasets are in the {prompt, chosen, rejected} triplet format. We preprocess the alignment datasets following Zephyr (Tunstall et al., 2023). We use Dataverse (Park et al., 2024) for data preprocessing.

Evaluation. In the HuggingFace Open LLM Leaderboard (Beeching et al., 2023), six types of evaluation methods are presented: ARC (Clark et al., 2018), HellaSWAG (Zellers et al., 2019), MMLU (Hendrycks et al., 2020), TruthfulQA (Lin et al., 2022), Winogrande (Sakaguchi et al., 2021), and GSM8K (Cobbe et al., 2021). We utilize these datasets as benchmarks for evaluation and also report the average scores for the six tasks, *e.g.*, H6. We either submit directly to the Open LLM Leaderboard or utilize Evalverse (Kim et al., 2024b) for running evaluations locally.

Model merging. Model merging methods such as Yadav et al. (2023) can boost model performance without further training. We merge some of the models that we trained in both the instruction and alignment tuning stages. We implement our own merging methods although popular open source also exist such as MergeKit[3].

4.2 Main Results

We present evaluation results for our SOLAR 10.7B and SOLAR 10.7B-Instruct models along

with other top-performing models in Tab. 2. SOLAR 10.7B outperforms other pretrained models of similar sizes, such as Qwen 14B and Mistral 7B, which shows that DUS is an effective method to up-scale base LLMs. Furthermore, despite the smaller size, SOLAR 10.7B-Instruct scores the highest in terms of H6, even surpassing the recent top-performing open-source LLM Mixtral 8x7B-Instruct-v0.1 or Qwen 72B. The above results indicate DUS can up-scale models that are capable of achieving state-of-the-art performance when fine-tuned. We also report data contamination results for SOLAR 10.7B-Instruct in Appendix C.

4.3 Ablation Studies

We present ablation studies for both the instruction and alignment tuning stages. Note that the evaluation results for the following studies are ran locally and may vary from results obtained by submitting to the Open LLM Leaderboard.

4.3.1 Instruction Tuning

Ablation on the training datasets. We present ablation studies using different training datasets for the instruction tuning in Tab. 3. The ablated models are prefixed with SFT for supervised fine-tuning. 'SFT v1' only uses the Alpaca-GPT4 dataset, whereas 'SFT v2' also uses the OpenOrca dataset. 'SFT v3' uses the Synth. Math-Instruct dataset along with the datasets used in 'SFT v2'. Similarly, 'SFT v4' uses the Synth. Math-Instruct dataset along with the datasets used in 'SFT v1'.

First, we analyze how Alpaca-GPT4 and OpenOrca affect the trained models. The first ablated model, 'SFT v1', which used only the Alpaca-GPT4 dataset for training, resulted in 69.15 for H6.

---

[3]https://github.com/cg123/mergekit

Model	Alpaca-GPT4	OpenOrca	Synth. Math-Instruct	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
SFT v1	O	✗	✗	69.15	**67.66**	**86.03**	65.88	**60.12**	**82.95**	52.24
SFT v2	O	O	✗	69.21	65.36	85.39	65.93	58.47	82.79	57.32
SFT v3	O	O	O	70.03	65.87	85.55	65.31	57.93	81.37	64.14
SFT v4	O	✗	O	70.88	67.32	85.87	65.87	58.97	82.48	64.75
SFT v3 + v4	O	O	O	**71.11**	67.32	85.96	**65.95**	58.80	82.08	**66.57**

Table 3: Ablation studies on the different datasets used for instruction tuning. 'SFT v3+v4' indicates that the model is merged from 'SFT v3' and 'SFT v4' by simply averaging the model weights. The best scores for H6 and the individual tasks are shown in bold.

Model	Ultrafeedback Clean	Synth. Math-Alignment	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
DPO v1	O	✗	73.06	71.42	**88.49**	**66.14**	72.04	81.45	58.83
DPO v2	O	O	**73.42**	**71.50**	88.28	65.97	71.71	**82.79**	**60.27**
DPO v1 + v2	O	O	73.21	71.33	88.36	65.92	**72.65**	**82.79**	58.23

Table 4: Ablation studies on the different datasets used during the direct preference optimization (DPO) stage. 'SFT v3' is used as the SFT base model for DPO. We name ablated models with the 'DPO' prefix to indicate the alignment tuning stage. 'DPO v1+v2' indicates that the model is merged from 'DPO v1' and 'DPO v2' by simply averaging the model weights. The best scores for H6 and the individual tasks are shown in bold.

Model	Base SFT Model	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
DPO v2	SFT v3	73.42	**71.50**	**88.28**	**65.97**	71.71	**82.79**	60.27
DPO v3	SFT v3 + v4	**73.58**	71.33	88.08	65.39	**72.45**	81.93	**62.32**

Table 5: Ablation studies on the different SFT base models used during the direct preference optimization (DPO) stage. Ultrafeedback Clean and Synth. Math-Alignment datasets are used. We name ablated models with the 'DPO' prefix to indicate the alignment tuning stage. The best scores for H6 and the individual tasks are shown in bold.

When we add the OpenOrca dataset to train the second ablated model, 'SFT v2', the resulting H6 score is 69.21, which is little change from 69.15 of 'SFT v1'. However, the task scores vary more as 'SFT v2' gets a substantially higher GSM8K score of 57.32 compared to 52.24 of 'SFT v1' but also gets noticeably lower scores across the board for ARC, HellaSwag, and TruthfulQA. This seems to indicate that using OpenOrca results in a model that behaves differently from using only Alpaca-GPT4.

Second, we investigate whether Synth. Math-Instruct dataset is beneficial. For 'SFT v3', we add the Synth. Math-Instruct dataset, which boosts GSM8K scores to 64.14 and achieves comparable scores for the other tasks. Interestingly, when we add the Synth. Math-Instruct dataset to 'SFT v1' to train 'SFT v4', we get our highest H6 score of 70.88 with higher scores than 'SFT v3' for all tasks. From the above, we can see that adding the Synth. Math-Instruct dataset is helpful.

Lastly, we see whether merging models trained with and without OpenOrca can boost performance. In the first analysis, we saw that using OpenOrca resulted in a model that behaved differently from the model that was trained without OpenOrca. Building on this intuition, we merge 'SFT v3' and 'SFT v4' as they are the best-performing models with and without OpenOrca. To our surprise, the resulting merged model 'SFT v3+v4' retains the high scores for non-GSM8K tasks from 'SFT v4' but also achieves a higher GSM8K score than 'SFT v3' or 'SFT v4'. Thus, we see that merging models that specialize in different tasks is a promising way to obtain a model that performs well generally.

4.3.2 Alignment Tuning

As we utilize sDPO for practical alignment tuning, there are additional aspects to ablate such as the SFT base models used. Thus, we present ablations for the different training datasets used for training, the different SFT base models to initialize the sDPO training, and finally, the model merging strategy to obtain the final alignment-tuned model.

Ablation on the training datasets. We ablate on the different alignment datasets used during DPO in Tab. 4. We use 'SFT v3' as the SFT base model for DPO. 'DPO v1' only uses the Ultrafeedback Clean dataset while 'DPO v2' also used the Synth. Math-Alignment dataset.

First, we test how Ultrafeedback Clean and Synth. Math-Alignment impacts model performance. For 'DPO v1', it achieves 73.06 in H6, which is a substantial boost from the SFT base model score of 70.03. However, we note that while

Model	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
Cand. 1	**73.73**	70.48	87.47	65.73	70.62	81.53	**66.57**
Cand. 2	73.28	**71.59**	**88.39**	66.14	**72.50**	**81.99**	59.14

Table 6: Performance comparison amongst the merge candidates. 'Cand. 1' and 'Cand. 2' are trained using the same setting as 'DPO v2' and 'DPO v3', respectively, but with slightly different hyper-parameters. The best scores for H6 and the individual tasks are shown in bold.

Model	Merge Method	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
Merge v1	Average (0.5, 0.5)	74.00	**71.16**	88.01	66.14	71.71	**82.08**	64.90
Merge v2	Average (0.4, 0.6)	73.93	71.08	**88.08**	**66.27**	**71.89**	81.77	64.52
Merge v3	Average (0.6, 0.4)	**74.05**	71.08	87.88	66.13	71.61	**82.08**	**65.50**
Merge v4	SLERP	73.96	**71.16**	88.03	66.25	71.79	81.93	64.59

Table 7: Ablation studies on the different merge methods used for obtaining the final model. We use 'Cand. 1' and 'Cand. 2' from Tab. 6 as our two models for merging. We name the merged models with the 'Merge' prefix to indicate they are merged. The best scores for H6 and the individual tasks are shown in bold.

scores for tasks like ARC, HellaSwag, and TruthfulQA all improved by good margins, the score for GSM8K is 58.83, which is lower than the SFT base model score of 64.14. Adding Synth. Math-Alignment to train 'DPO v2', we see that the GSM8k score improves to 60.27, which is lower than the SFT base model but still higher than 'DPO v1'. Other task scores are also not negatively impacted by adding Synth. Math-Alignment. Thus, we can conclude that adding Synth. Math-Alignment is beneficial for H6.

Then, we experiment whether merging 'DPO v1' and 'DPO v2' is beneficial. Unfortunately, 'DPO v1+v2' scores 73.21 in H6, which is worse than 'DPO v2'. More importantly, the gain in the GSM8K score from adding Synth. Math-Alignment is gone, which is undesirable. One reason for this could be that 'DPO v2' is a strict improvement over 'DPO v1', unlike the case for merging 'SFT v3' and 'SFT v4' where the models had different strengths and weaknesses.

Ablation on the SFT base models. When applying DPO, we start from a model that is already instruction tuned ,*i.e.,* the SFT base model and ablate on using different SFT base models. We use Ultrafeedback Clean and Synth. Math-Alignment datasets for this ablation. Each of the ablated models is trained as follows. 'DPO v2' uses 'SFT v3' as the base SFT model, while 'DPO v3' uses 'SFT v3+v4' as the SFT base model instead.

Note that 'SFT v3+v4' has higher scores on all tasks compared to 'SFT v3', and the gap is especially large for ARC (+1.45) and GSM8K (+2.43). Surprisingly, the two models perform similarly in terms of H6. A closer look at the scores for the

individual tasks shows only a small margin in the GSM8K scores, and other task scores show little difference. Thus, the performance gaps in certain tasks in the SFT base models do not always carry over to the alignment-tuned models.

Ablation on different merge methods. From Tab. 3, we saw that merging two models that have different strengths can be beneficial to performance. To utilize this for the alignment-tuned model as well, we train two models named 'Cand. 1' and 'Cand. 2' using the same training dataset and SFT base model as 'DPO v2' and 'DPO v3' but with different hyper-parameters to maximize each model's respective strengths. We compare 'Cand. 1' and 'Cand. 2' in Tab. 6 where we can see that 'Cand. 1' has high GSM8K scores but relatively low scores for the other tasks, whereas 'Cand. 2' has low scores for GSM8K but high scores for the other tasks. We merge these two models using various methods and ablate the results in Tab.. 7.

We use two merge methods: 1) Average (a, b), where a and b denote the weighting for 'Cand. 1' and 'Cand. 2' when averaging weights and 2) SLERP (Shoemake, 1985). We use (0.5, 0.5), (0.4, 0.6), and (0.6, 0.4) for Average (a, b). From Tab. 7, we can see that the different merge methods have little effect on the H6 scores. The scores for the individual tasks also do not differ by much, suggesting that as long as the merge candidates have sufficiently different strengths, the exact merge method may not be as crucial. Thus, we chose 'Merge v1' as our SOLAR 10.7B-Instruct model.

5 Conclusion

We introduce SOLAR 10.7B and its fine-tuned variant SOLAR 10.7B-Instruct, which are depth upscaled (DUS) models with 10.7 billion parameters[4]. They show superior performance over models like Llama 2, Mistral 7B, and Mixtral-7B-Instruct in essential NLP tasks while maintaining computational efficiency. Thus, DUS is effective in scaling-up highly performant LLMs from smaller ones. With more exploration, DUS could be further improved, paving a new path to efficiently scaling LLMs.

Acknowledgements

Limitations

Our study on the Depth Up-Scaling (DUS) has important limitations and considerations. One key limitation is the need for more thorough explorations of hyperparameters used in the DUS approach. Namely, we removed $m = 8$ layers from both ends of our base model, primarily due to hardware limitations. However, we have not yet determined if this value is optimal for enhancing performance. The extended time and cost of continued pretraining made it challenging to conduct more comprehensive experiments, which we aim to address in future work through various comparative analyses.

In terms of the model's broader implications, there are several points to note. The model's significant computational demands for training and inference might limit its use, especially for those with restricted computational resources. Additionally, like all machine learning models, it is vulnerable to biases in its training data, which could lead to skewed outcomes in certain situations. Furthermore, the substantial energy consumption required for training and operating the model raises environmental concerns, which are critical in the pursuit of sustainable AI development.

Lastly, while the fine-tuned variant of the model shows improved performance in following instructions, it still requires task-specific fine-tuning for optimal performance in specialized applications. This fine-tuning process can be resource-intensive and not always effective. Recognizing and addressing these limitations is essential for a comprehensive understanding of the proposed Large Language Model's capabilities and for guiding future research and development in the field of LLMs.

Ethics Statement

We conscientiously address and emphasize the commitment of SOLAR 10.7B in maintaining the highest ethical standards. First, we highlight that SOLAR 10.7B-Instruct has shown low levels of data contamination in our evaluations, a testament to our rigorous data handling and processing protocols. This aspect is crucial, as it underpins the reliability and integrity of the results obtained from SOLAR.

Furthermore, during the course of our experiments, we ensured that all setups and methodologies employed steer clear of any potential ethical pitfalls. This preemptive consideration and avoidance of ethically questionable practices underscore our dedication to conducting research that is not only innovative but also responsible.

Additionally, we ensure that SOLAR complies with general ethical considerations in all aspects of its operation. This includes adherence to privacy norms, respect for intellectual property, and ensuring the absence of bias in our algorithms. Our commitment to these ethical principles is unwavering, and we believe it significantly contributes to the credibility and societal acceptance of SOLAR.

In conclusion, the ethical framework within which SOLAR operates is robust and comprehensive, ensuring that our advancements in this field are not only scientifically sound but also ethically responsible.

---

[4]Preprint version is available on https://arxiv.org/abs/2312.15166.