Konstantin Dobler and Gerard de Melo

Tokenizer Swapping Works and bfloat16 is Enough

HPI Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Language Adaptation on a Tight Academic Compute Budget

Tight Academic Compute Budgets

Since **GPUs are a shared resource**, any individual often can only access:

- Limited **number of GPUs** (such as only two or four)
- Limited **GPU memory** (such as 40GB vs. 80GB Nvidia A100)
- For a **limited time** (long runs only on weekends / holidays)

Approach

We focus in two approaches to **increase efficiency** of language adaptation **on a tight compute budget**:

- **Pure** `bfloat16` instead of mixed precision: mixed precision requires `float32` **optimizer states** and a **model copy** → we need to use **expensive memory-saving techniques** like paged optimizers or gradient checkpointing
- **Tokenizer swapping:** a specialized tokenizer has better *fertility*, allowing us to train **on more words** (although not tokens) **for the same compute**

Experimental Setup

We adapt Mistral-7B-v0.1 [Jiang et al., 2023] to German and Arabic. We start from the LeoLM [Plüster et al., 2023] recipe but modify it for a tight compute budget: only 8 billion training tokens, smaller batch size, 4k sequence length, 4e-5 learning rate.

Main experiments: Adapt to German with data from OSCAR23.01 [Abadji et al., 2022]. We run cartesian product of {pure bfloat16, mixed precision} X {original tokenizer, tokenizer swapping}.

Hindsight runs: German and Arabic from CulturaX [Nguyen et al., 2024]. Only tokenizer swapping and pure `bfloat16`. Ablation: use mixed precision just for final lr annealing. Prevent cross-document attention.

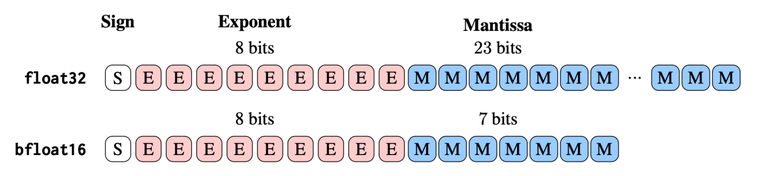
Key Findings

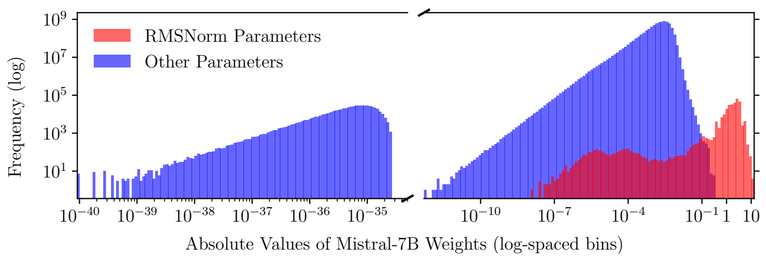
Pure `bfloat16` is very useful in tight budget settings!

Precision	GPUs	Best Config	GPU Hours	Speedup
mixed	1	OOM	OOM	–
pure	1	(1, no, N/A, N/A, paged)	228.3	∞
mixed	2	(4, yes, full, sync, paged)	317.0	–
pure	2	(1, no, grad_op, no_sync, no_paged)	227.7	39.2%
mixed	4	(8, yes, full, sync, no_paged)	295.7	–
pure	4	(1, no, grad_op, no_sync, no_paged)	225.8	31.0%
mixed	8	(8, yes, full, sync, paged)	298.0	–
pure	8	(1, no, grad_op, no_sync, no_paged)	229.6	29.8%

Numerics of pure `bfloat16`

Weight updates for **RMSNorm parameters are flushed to zero.**





bfloat16	Layer type	Avg. parameter change
mixed	RMSNorm	0.0048
pure	RMSNorm	0.000004
mixed	not RMSNorm	0.0015
pure	not RMSNorm	0.0012

Table 4: Average change of parameter values at the end of training compared to their starting values depending on layer type (RMSNorm or others) and pure or mixed-precision `bfloat16` training.

Pure `bfloat16` performs as well as mixed precision!

Pure `bfloat16` **matches mixed precision closely in terms of loss** and actually **outperforms on benchmarks**. At the very end of training, we see that pure `bfloat16` loss does not improve. This is due to `bfloat16` numerics – we run an ablation that used mixed precision instead at the end but do not see conclusive benefits.

Tokenizer swapping works but doesn't improve results.

Tokenizer swapping **matches performance of original vocabulary** but **does not result in better performance** despite training on more total words (due to tokenizer fertility) for the same compute.

Language adaptation is not always helpful.

Intuitively, „focusing" model capacity just on the target language might be beneficial. However **for German, all our adapted models underperform base** Mistral-7B (**our Arabic models outperform**!).

		NLL at % of training						
bfloat16	Tokenizer	0%	10%	30%	50%	70 %	90%	100%
mixed	German	5.84	1.96	1.76	1.67	1.60	1.56	1.55
pure	German	5.84	1.99	1.76	1.67	1.61	1.59	1.59
mixed	original	2.56	1.96	1.79	1.70	1.62	1.58	1.57
pure	original	2.56	1.98	1.79	1.69	1.62	1.60	1.60

Table 3: Word-normalized negative log-likelihood (NLL) of a held-out test set throughout continued pretraining of Mistral-7B on German text.

	ACVA	AlGhafa	MMLU-AR	AlGhafa-T	Macro Avg.
Arabic Mistral-7B (w/ tokenizer swapping & pure `bfloat16`, see Section 3.2)					
pure	**73.2**	62.5	**37.9**	52.4	53.6
pure++†	70.7	**63.3**	37.7	**52.8**	**53.8**
Baselines					
Mistral-7B	63.0	**57.7**	**33.9**	**46.3**	**47.4**
AceGPT-7B	**71.0**	52.6	27.0	44.6	45.8
Llama 2-7B	66.3	45.6	27.4	41.3	42.4

Table 7: Results on Arabic downstream task suites. The average is a macro-average that includes the individual benchmarks in AlGhafa-T. †: For pure++ `bfloat16`, mixed precision was used just for the final annealing phase of the learning rate schedule.

		German translations (Plüster, 2023)				German test splits		
Tokenizer	bfloat16	MMLU	HellaSwag	TruthfulQA	ARC	LAMBADA	PAWS-X	Avg.
Main experiments (see Section 3.1)								
German	mixed	33.6	**60.0**	37.5	43.0	40.3	62.3	46.1
German	pure	**35.9**	59.7	**39.4**	**44.1**	**40.6**	**63.6**	**47.2**
original	mixed	32.6	**59.5**	**43.0**	40.8	38.8	63.5	46.4
original	pure	**37.2**	59.4	39.2	**41.6**	**39.3**	**63.9**	**46.8**
Improved hindsight runs (see Section 3.2)								
German	pure	43.5	63.4	**39.8**	**47.9**	**37.9**	**62.5**	**49.1**
German	pure++†	**43.6**	**63.5**	39.5	46.7	**37.9**	**62.5**	48.9

Table 5: Effectiveness of models based on Mistral-7B on German downstream tasks. The best result in each section is **bolded** and the overall best result of the main experiments is additionally **underlined**. †: for pure++ `bfloat16`, mixed precision was used just for the final annealing phase of the learning rate schedule.

Paper, code, checkpoints on GitHub

GitHub

SPONSORED BY THE
Federal Ministry of Education and Research
KI Service Zentrum by Hasso-Plattner-Institut
HPI