

Index SLM Technical Report

▣ [bilibili/Index-1.9B](#) ✦ [IndexTeam/Index-1.9B-Chat](#)

Bilibili Index LLM Team

Lusheng Zhang, Shien He, Tianxing Yan, Mengran Yu, Ziang Cui, Kai Zhao, Xiaojing Liu,
Tianjiao Li[†]

{zhanglusheng01, heshien, yantianxing, yumengran, cuiziang, zhaokai,
liuxiaojing, litianjiao01}@bilibili.com

[†]Project lead

Abstract

We present Index-1.9B, a series of open small language models developed at Bilibili. The series comprises four models: Index-1.9B-Base, a foundation model with 1.9 billion non-embedding parameters pre-trained on 2.8 trillion predominantly Chinese and English tokens; Index-1.9B-Pure, a control variant trained with an identical recipe but with all instruction-like data strictly filtered from the corpus; Index-1.9B-Chat, aligned from the base model with supervised fine-tuning and direct preference optimization; and Index-1.9B-Character, which augments the chat model with retrieval-augmented generation for few-shot role-playing customization. Pre-training employs a Warmup–Stable–Decay learning-rate schedule in which the concentration of curated data is raised substantially during the decay phase, together with a Norm-Head output layer that stabilizes training under large learning rates. On a suite of standard benchmarks covering examination, reasoning, mathematics, and code, Index-1.9B-Base attains an average score of 64.92, competitive with or exceeding open models of several times its size. We further report controlled studies on model depth, learning-rate magnitude and scheduling, the interaction between learning-rate decay and data quality, and the effect of including instruction data during pre-training, and we document an unexplained surge in benchmark performance midway through the constant-learning-rate phase. All models, together with evaluation code, are released at <https://github.com/bilibili/Index-1.9B>.

Contents

1	Introduction	2
2	Pre-training	2
2.1	Data	2
2.2	Tokenizer	3
2.3	Model Architecture	4
2.4	Training Recipe	4
2.5	Infrastructure	5
3	Alignment	5
3.1	Supervised Fine-tuning	5
3.2	Direct Preference Optimization	5
4	Few-shot Role-playing	6
5	Evaluation	6
5.1	Setup	6
5.2	Base Model Results	6
5.3	Chat Model Results	6
5.4	Role-playing Results	8

6 Discussion	8
6.1 Stabilizing the LM Head: Norm-Head	8
6.2 Depth versus Width	8
6.3 Learning-rate Magnitude	9
6.4 Learning-rate Schedules	9
6.5 Coupling Learning-rate Decay with Data Quality	9
6.6 Instruction Data in Pre-training	11
6.7 A Performance Surge During the Stable Phase	11
7 Limitations	12
8 Conclusion	12
A Qualitative Examples	15
B Safety Preference-pair Construction	16

1 Introduction

Small language models (SLMs) with one to three billion parameters have recently attained capabilities that previously required far larger models, driven by higher-quality data and by training well beyond compute-optimal token budgets [Hoffmann et al., 2022, Javaheripi et al., 2023, Gemma Team et al., 2024, Hu et al., 2024, Bai et al., 2023]. Their modest inference cost makes them attractive for large-scale deployment and on-device use. This report describes the design, training, alignment, and evaluation of Index-1.9B, the lightweight tier of the Index model family developed at Bilibili.

The release consists of the following models:

- **Index-1.9B-Base.** A foundation model with 1.9 billion non-embedding parameters, pre-trained on 2.8T tokens of predominantly Chinese and English text. It leads multiple benchmarks among models of comparable size (Section 5.2).
- **Index-1.9B-Pure.** A control counterpart of the base model, trained with the same parameters and schedule but with all instruction-related data strictly filtered from the corpus. It isolates the effect of instruction data on benchmark scores (Section 6.6).
- **Index-1.9B-Chat.** A dialogue model aligned from the base model via supervised fine-tuning (SFT) and direct preference optimization (DPO) [Rafailov et al., 2023]. Owing to the large amount of curated community-forum corpus introduced during pre-training, the model exhibits notably engaging conversational behavior and strong translation among East Asian languages.
- **Index-1.9B-Character.** Built on the aligned model, it combines role-conditioned fine-tuning with retrieval-augmented generation (RAG) [Lewis et al., 2020] to support few-shot role-playing customization (Section 4).

In support of open research on training dynamics, we additionally release an intermediate checkpoint taken before the learning-rate decay phase (INDEX-1.9B-CONSTANT-LR), and a long-context extension supporting 32K tokens (INDEX-1.9B-32K) is available from the project repository. All weights are available on Hugging Face and ModelScope and are open for academic research and free commercial use.

Beyond the models themselves, this report contributes a set of controlled experiments that informed our design choices: a comparison of mechanisms for stabilizing the output projection (Section 6.1); a depth-versus-width study at fixed parameter count (Section 6.2); analyses of learning-rate magnitude, scheduling, and their interaction with data quality (Sections 6.3–6.5); and a transparent ablation quantifying how instruction data in pre-training inflates benchmark scores (Section 6.6).

2 Pre-training

2.1 Data

Index-1.9B is pre-trained on 2.8T tokens with a Chinese-to-English ratio of 4:5; code accounts for 6% of the corpus. We additionally curate publicly available books, encyclopedias, academic papers, and STEM-related

material, which together constitute roughly 10% of the mixture, and we raise the concentration of this curated subset in the late stage of pre-training (Section 2.4). Figure 1 shows the overall composition.

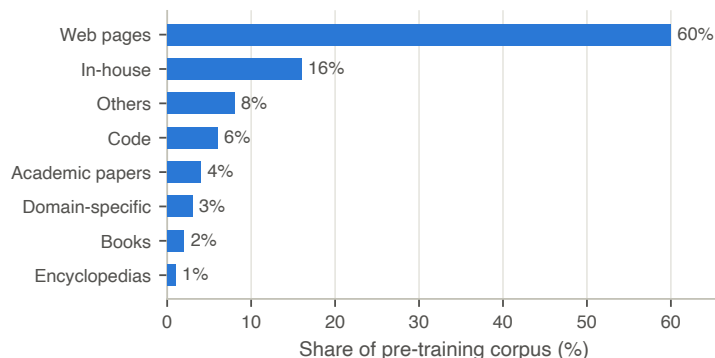


Figure 1: Composition of the Index-1.9B pre-training corpus.

Substantial effort was devoted to data cleaning, organized around three components.

Bias-aware filtering. To avoid introducing avoidable model-induced bias, the great majority of the corpus is cleaned with heuristic rules. Classifiers are trained only for samples that heuristics cannot reliably handle, with labels bootstrapped by annotation from our larger Index-Large model, and are applied conservatively.

Document-level deduplication at the tens-of-billions scale. Deduplication is critical to corpus quality [Lee et al., 2022]. We built a Spark-based pipeline that performs global MinHash [Broder, 1997] comparison and deduplication over tens of billions of documents in a single pass.

Exact substring deduplication. In contrast to deduplication over pre-segmented paragraphs or sentences, we support within-document duplicate detection for strings of arbitrary length at arbitrary positions, which surfaces problematic text that is otherwise difficult to discover. Our implementation extends the open-source suffix-array toolkit of Lee et al. [2022] with global deduplication free of memory limits, visual diagnostics, and the option to retain a single occurrence. As an illustration of why this matters, one drop-down menu string enumerating month names recurred 156,000 times in Common Crawl and was identifiable only through exact substring matching.

2.2 Tokenizer

We train a byte-pair-encoding (BPE) tokenizer [Sennrich et al., 2016] with SentencePiece [Kudo and Richardson, 2018], with three deliberate departures from common practice.

First, the Chinese portion of the vocabulary is trained separately. SentencePiece treats a Chinese character and a Latin letter as units of the same granularity when growing BPE merges, which we consider a poor fit for Chinese; we therefore reduce the maximum piece length from the default of 16 to 5 for the Chinese sub-vocabulary.

Second, the vocabulary is kept small. The smaller the model, the larger the share of memory occupied by the embedding matrix: a 150K-entry vocabulary alone can account for more than 30% of the memory footprint of a 1B-parameter model. Our final vocabulary contains 65,029 tokens.

Third, no whitespace is prepended to the input. Tokenizers in the Llama family [Touvron et al., 2023a] automatically prepend a space to the text, which is unfriendly to Chinese and invalidates the convention that the first token of a document is never predicted; we remove this behavior.

Table 1 compares compression rates against several bilingual tokenizers. At a compact vocabulary size, the Index tokenizer achieves competitive compression on Chinese and English and the strongest compression on Japanese and Korean among the tokenizers compared.

Table 1: Tokenizer compression rates, computed on held-out in-house corpora as $\text{len}(\text{token ids})/\text{len}(\text{text})$; lower is better. “Mixed” denotes an in-house corpus mixing all domains.

Tokenizer	Vocab	Mixed	Chinese	English	Japanese	Korean	Code
Baichuan 2	125,696	0.6111	0.6137	0.2444	0.7640	0.7211	0.3473
Llama	32,000	1.4019	1.4563	0.2658	0.9856	1.0473	0.3408
Skywork	65,519	0.7216	0.7161	0.2653	0.7855	1.0111	0.3390
Yi	64,000	0.6623	0.6740	0.2464	0.9533	1.2314	0.3424
Index	65,029	0.6158	0.6489	0.2672	0.6390	0.6055	0.3434

2.3 Model Architecture

Index-1.9B follows the mainstream decoder-only Transformer design [Vaswani et al., 2017] and adopts the architectural conventions of Llama [Touvron et al., 2023a,b], including rotary position embeddings [Su et al., 2024], SwiGLU activations [Shazeer, 2020], and RMSNorm [Zhang and Sennrich, 2019], with two modifications.

Greater depth. Guided by the experiments in Section 6.2, we find that, at fixed parameter count, moderately increasing depth improves downstream performance; we set the number of layers to 36.

Norm-Head. During training, the gradient of the output projection (LM head) is an order of magnitude larger than that of any other layer, and the sparsity of the vocabulary induces oscillation on rare tokens, both of which destabilize training. Among published remedies — the output-layer gradient scaling of GLM-130B [Zeng et al., 2023], the logit scaling of μP [Yang et al., 2022], and the Norm-Head of Baichuan 2 [Yang et al., 2023] — we consider Norm-Head, which normalizes the head weights and thereby rescales them dynamically, the most principled. In our experiments it yields consistent gains and tolerates higher learning rates (Section 6.1), and we adopt it.

Table 2 summarizes the configuration.

Table 2: Model configuration of Index-1.9B.

Hidden size	FFN size	Heads	Layers	Seq. length	Peak LR	Batch size
2048	5888	16	36	4096	5×10^{-4}	4M tokens

2.4 Training Recipe

We optimize with AdamW [Loshchilov and Hutter, 2019] ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$), gradient clipping of 1.0, and weight decay of 0.1. Training follows a two-stage strategy under the Warmup–Stable–Decay (WSD) learning-rate schedule proposed for MiniCPM [Hu et al., 2024]; Section 6.4 presents our analysis of this schedule and its interaction with data.

1. **Stable phase.** After a 100-step warmup, the learning rate is held constant while the model trains on the global data mixture.
2. **Decay phase.** In the final stage of training the learning rate decays, the model enters a regime of rapid learning, and we substantially raise the concentration of curated data.

Two settings distinguish our decay phase from common practice. First, because Norm-Head tolerates large learning rates, we set the constant-phase learning rate to 5×10^{-4} and decay it to 1% of its peak (5×10^{-6}), a wider decay range than is typical. Second, the ample decay range permits a long decay: the decay phase consumes 400B tokens, during which we observe benchmark performance continuing to improve throughout.

Instruction data during decay. Whether instruction data is included during pre-training is rarely stated explicitly in public reports. We train two versions differing in exactly this respect: Index-1.9B-Pure uses natural text from the stable phase through the decay phase, while Index-1.9B (also referred to as *Boost*) additionally includes a proportion of instruction data during decay. Both are released with their evaluation results so that readers may judge the effect directly; Section 6.6 provides a controlled analysis.

2.5 Infrastructure

Training was conducted with our in-house framework on 128 Huawei Ascend 910B accelerators in bfloat16, with a 4K context length, and required approximately 28 days for the 2.8T tokens. Samples are packed into full sequences with attention masks and position identifiers reset at document boundaries. Selective activation checkpointing reduces memory pressure; communication, computation, and data loading are overlapped; and jobs resume from failures within minutes.

3 Alignment

To align the model with human preferences we apply SFT followed by DPO [Rafailov et al., 2023], in the spirit of the instruction-following literature [Ouyang et al., 2022].

3.1 Supervised Fine-tuning

Data. We collect more than ten million Chinese and English instruction–response pairs. The pool is cleaned and filtered with clustering-based diversity selection and reward-model scoring, following common practice, to obtain a compact subset of high quality and diversity; for instruction types on which the fine-tuned model underperforms, we construct and annotate targeted data. The final SFT set contains fewer than 100,000 examples.

Training. The chat model adopts a system–query–response format. Optimizer settings follow pre-training, with a learning rate of 1×10^{-5} . As in pre-training, samples are packed across documents to improve throughput, but tokens outside the response are masked from the loss. We ablate whether to initialize the optimizer state from pre-training and whether to replay pre-training corpus at a fixed ratio (Table 3); the best configuration loads the pre-training optimizer state and keeps the share of instruction tokens contributing to the loss at approximately 60%.

Table 3: Internal evaluation of SFT training strategies (scores out of 3). “+opt.” denotes initializing from the pre-training optimizer state; “+replay” denotes replaying pre-training corpus at 40% of tokens.

Strategy	Instr.	Knowledge	Math & reas.	Open QA	Writing	Riddles	Overall
SFT + opt. + replay	1.807	1.938	1.414	2.353	2.243	1.821	1.929
SFT + opt.	1.943	2.192	1.661	1.498	1.893	1.718	1.818
SFT	1.702	1.902	1.387	2.060	1.940	1.769	1.793

System-prompt steerability. Adjusting the system prompt reliably steers the register and persona of responses, enabling role-playing and style transfer; Appendix A shows examples.

3.2 Direct Preference Optimization

DPO training targets three areas: writing, instruction following, and safety. For open-ended writing, a single reference response is rarely the unique optimum; preference learning lets the model internalize a standard of relative quality rather than imitate one target. For instruction following and safety, contrasting chosen and rejected responses teaches the model the constraints of the instruction — length control is a representative success case — and the categories of requests that warrant refusal. In general, we find that tasks whose evaluation criteria are discriminative rather than enumerable benefit the most from preference learning.

Data. For generation tasks, we select writing-oriented prompts from the SFT pool, score sampled model responses with a reward model trained in-house, and assemble the scored responses into preference pairs. For instruction-following constraints, pairs are constructed and annotated manually. For safety, we compared two schemes: (i) pairing a human-written refusal, drawn from a curated collection, as the chosen response against the SFT model’s response as rejected; and (ii) inducing the SFT model itself to generate the refusal via the system prompt and using that self-generated refusal as chosen. We adopted the second. Human-written refusals have high perplexity under the SFT model, and forcing alignment toward them inflates the refusal rate and causes

catastrophic forgetting, whereas self-generated refusals align the model toward refusing unsafe requests without these side effects (Appendix B gives an example pair). In total we construct over 100,000 preference pairs.

Training. DPO uses the same conversation format as SFT, a learning rate of 1×10^{-6} with a cosine schedule, $\beta = 0.1$, and a single epoch.

4 Few-shot Role-playing

Index-1.9B-Character extends the aligned model with retrieval-augmented role-playing. From publicly available scripts, transcripts, and character-profile data, we extract character dialogues, filter them with a role-specific reward model, and clean the result into a corpus of roughly 80,000 high-quality dialogues covering more than one thousand characters. At training time, we use RAG [Lewis et al., 2020] to retrieve excerpts of a character’s past utterances relevant to the current exchange and concatenate them into the prompt as references. The same mechanism at inference time allows users to instantiate a custom persona from a small uploaded dialogue corpus, i.e., few-shot role-playing customization; evaluation follows in Section 5.4.

5 Evaluation

5.1 Setup

Base models are evaluated with mainstream public benchmarks under OpenCompass [OpenCompass Contributors, 2023], with compatibility modifications that we release for reproducibility:

- **Comprehensive examinations:** MMLU [Hendrycks et al., 2021], C-Eval [Huang et al., 2023], and CMMLU [Li et al., 2023], evaluated 5-shot via perplexity.
- **Understanding and reasoning:** HellaSwag [Zellers et al., 2019], ARC-Challenge, and ARC-Easy [Clark et al., 2018], evaluated 0-shot via perplexity. Two perplexity protocols are computed — OpenCompass’s default, which scores the concatenation of question and option text, and an answer-letter variant that scores the full question with the option letter — and the higher score is reported.
- **Mathematics and code:** GSM8K [Cobbe et al., 2021] and HumanEval [Chen et al., 2021], evaluated by generation followed by answer extraction and verification.

5.2 Base Model Results

Table 4 reports results against open models from roughly 2B to 40B parameters. Index-1.9B attains an average score of 64.92 and an English average of 69.93, exceeding all models of comparable size except Qwen2-1.5B on the overall average, and surpassing a number of substantially larger models, including Llama-2-13B. The gap between Index-1.9B and Index-1.9B-Pure (64.92 vs. 50.61) reflects the contribution of instruction data during the decay phase, analyzed in Section 6.6.

Mathematics and code remain areas for improvement, which we intend to address in future iterations; Table 5 reports current results, which are on par with Llama-2-13B.

5.3 Chat Model Results

To evaluate the aligned models we constructed an in-house benchmark of more than 300 prompts spanning five categories — instruction following, knowledge question answering, open-ended question answering, writing, and mathematical reasoning — scored on a three-point scale. Table 6 compares the DPO model with its SFT predecessor. DPO improves open-ended categories, where quality is discriminative rather than enumerable (open-ended QA and writing), and yields a modest gain on instruction following, while leaving knowledge and reasoning essentially unchanged.

Table 4: Results on general benchmarks. “Avg.” averages all six task scores; “Avg. (en)” averages MMLU, HellaSwag, ARC-C, and ARC-E. Entries marked [†] are taken from the corresponding technical reports; models marked [‡] disclose their full training corpus (OLMo and Amber additionally disclose the training process). MiniCPM-2.4B-SFT is a fine-tuned model; MiniCPM-2.4B-Decay denotes the official intermediate checkpoint at step 280,000 released in the model’s training history. Missing entries are not reported by the original sources.

Model	Avg.	Avg. (en)	MMLU	C-Eval	CMMLU	HellaSwag	ARC-C	ARC-E
RedPajama-INCITE-3B ^{†‡} [Together Computer, 2023]	–	46.58	27.1	–	–	63.2	34.4	61.6
OpenLLaMA-3B-v2 ^{†‡} [Geng and Liu, 2023]	–	47.23	26.7	–	–	65.2	35.1	61.9
BTLM-3B-8K [‡] [Dey et al., 2023]	33.47	37.01	27.82	27.61	25.18	66.84	26.18	27.19
StableLM-Alpha-3B-v2 [†] [Stability AI, 2023]	–	44.78	26.6	–	–	65.8	32.9	53.8
Gemma-2B [Gemma Team et al., 2024]	41.58	46.77	41.81	31.36	31.02	66.82	36.39	42.07
Phi-2 (2.7B) [Jawaheripi et al., 2023]	58.89	72.54	57.61	31.12	32.05	70.94	74.51	87.10
Qwen1.5-1.8B [Bai et al., 2023]	58.96	59.28	47.05	59.48	57.12	58.33	56.82	74.93
Qwen2-1.5B [†] [Yang et al., 2024]	65.17	62.52	56.50	70.60	70.30	66.60	43.90	83.09
MiniCPM-2.4B-SFT [Hu et al., 2024]	62.53	68.75	53.80	49.19	50.97	67.29	69.44	84.48
MiniCPM-2.4B-Decay [Hu et al., 2024]	59.91	65.19	51.74	48.72	49.96	62.41	65.06	81.56
Index-1.9B-Pure	50.61	52.99	46.24	46.53	45.19	62.63	41.97	61.10
Index-1.9B	64.92	69.93	52.53	57.01	52.79	80.69	65.15	81.35
OpenLLaMA-7B-v2 ^{†‡} [Geng and Liu, 2023]	–	52.10	40.4	–	–	56.0	39.0	73.0
OLMo-7B [‡] [Groeneveld et al., 2024]	34.35	38.32	28.45	27.28	25.57	72.43	25.24	27.15
Amber-7B [‡] [Liu et al., 2023]	33.49	37.84	28.04	23.81	25.80	68.18	27.47	27.65
Llama-2-7B [Touvron et al., 2023b]	50.79	60.31	44.32	32.42	31.11	76.00	46.30	74.60
Mistral-7B [†] [Jiang et al., 2023]	–	69.23	60.1	–	–	81.3	55.5	80.0
Baichuan2-7B [Yang et al., 2023]	54.53	53.51	54.64	56.19	56.95	25.04	57.25	77.12
DeepSeek-7B [Bi et al., 2024]	55.84	60.83	49.32	44.93	46.82	75.40	50.64	67.95
Falcon-7B [†] [Almazrouei et al., 2023]	–	54.20	26.2	–	–	76.3	43.5	70.8
MPT-7B [†] [MosaicML NLP Team, 2023]	–	54.18	28.6	–	–	76.2	41.9	70.0
Llama-2-13B [Touvron et al., 2023b]	57.51	66.61	55.78	39.93	38.70	76.22	58.88	75.56
Baichuan2-13B [Yang et al., 2023]	68.90	71.69	59.63	59.21	61.27	72.61	70.04	84.48
MPT-30B [†] [MosaicML NLP Team, 2023]	–	63.48	46.9	–	–	79.9	50.6	76.5
Falcon-40B [†] [Almazrouei et al., 2023]	–	68.18	55.4	–	–	83.6	54.5	79.2

Table 5: Mathematics and code benchmarks.

Model	GSM8K	HumanEval
Llama-2-7B	16.76	12.8
Llama-2-13B	29.87	18.9
Index-1.9B-Pure	12.59	12.2
Index-1.9B	28.20	18.9

Table 6: In-house evaluation of the aligned models (scores out of 3).

Model	Instr.	Knowledge	Open QA	Writing	Math & reas.	Avg.
Index-1.9B-SFT	2.143	2.218	2.695	2.469	1.951	2.295
Index-1.9B-DPO	2.178	2.123	2.735	2.593	1.963	2.318

5.4 Role-playing Results

We evaluate Index-1.9B-Character on CharacterEval [Tu et al., 2024], a Chinese role-playing benchmark that scores character consistency, conversational ability, and role-playing attractiveness. Table 7 reports dimension averages. Index-1.9B-Character ranks ninth by overall score on the leaderboard — within a field otherwise composed of 7B-to-14B open models and closed commercial systems — and substantially outperforms every model of comparable scale.

Table 7: CharacterEval results (dimension averages; higher is better). Baseline scores follow Tu et al. [2024]. CC: character consistency; CA: conversational ability; RA: role-playing attractiveness.

Model	Overall	CC	CA	RA
CharacterYuyan-RLHF	3.360	2.856	4.037	3.188
CharacterYuyan	3.309	2.880	3.931	3.117
BC-NPC-Turbo	3.169	2.764	3.798	2.946
MiniMax	3.135	2.718	3.784	2.902
Baichuan2-13B [Yang et al., 2023]	3.129	2.701	3.795	2.890
InternLM-20B [InternLM Team, 2023]	3.124	2.715	3.745	2.911
Baichuan2-7B [Yang et al., 2023]	3.110	2.700	3.757	2.874
Index-1.9B-Character	3.109	2.676	3.789	2.862
InternLM-7B [InternLM Team, 2023]	3.034	2.620	3.698	2.784
XVERSE-13B	3.022	2.632	3.605	2.830
Qwen-14B [Bai et al., 2023]	3.016	2.649	3.542	2.858
GPT-4 [OpenAI, 2023]	3.006	2.697	3.448	2.873
Xingchen	2.991	2.595	3.646	2.732
XVERSE-7B	2.963	2.564	3.554	2.772
CharacterGLM [Zhou et al., 2023]	2.937	2.493	3.623	2.695
ChatGLM3-6B [Zeng et al., 2023]	2.898	2.556	3.399	2.739
Qwen-7B [Bai et al., 2023]	2.849	2.540	3.327	2.679
GPT-3.5	2.381	2.101	2.749	2.293

6 Discussion

This section presents the controlled experiments underlying our design decisions. Unless otherwise stated, ablations report an *average benchmark score* defined as the mean over C-Eval, CMMLU, MMLU, ARC-C (0-shot), ARC-E (0-shot), and HellaSwag (0-shot).

6.1 Stabilizing the LM Head: Norm-Head

The distribution of gradient magnitudes differs sharply across layers: the LM head alone accounts for the dominant share of the total gradient norm, while vocabulary sparsity makes this layer the least stable. A stable output layer is therefore essential to stable training. We adopt Norm-Head [Yang et al., 2023], which normalizes the head weights and thereby rescales the layer dynamically.

We compare a 1B-parameter model trained on 1T tokens under a cosine schedule with peak learning rate 2×10^{-4} against an otherwise identical run with Norm-Head. Figure 2 shows the result: the Norm-Head run scores consistently above the baseline throughout training. Its total gradient norm is higher in absolute terms, rises quickly at initialization, and thereafter drifts upward more slowly than the baseline’s. The same qualitative behavior reproduces at 0.1B scale.

6.2 Depth versus Width

How deep should a model of fixed size be? Kaplan et al. [2020] report that performance depends primarily on total parameter count and only weakly on shape, whereas Tay et al. [2022] find that deeper, narrower models transfer better downstream. We train two models of identical parameter count and FLOPs (1.01B non-embedding parameters): a 36-layer configuration and a 9-layer, wider counterpart. As Figure 3 shows, the deeper model is consistently better at equal size.

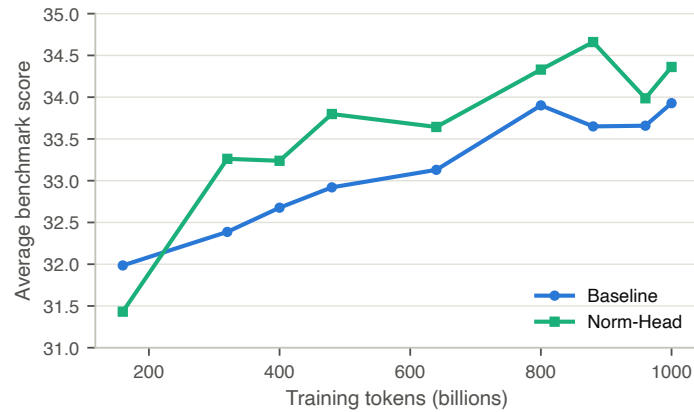


Figure 2: Average benchmark score of a 1B model trained on 1T tokens, with and without Norm-Head.

Two caveats apply. First, at fixed parameter count, increasing depth increases activation memory, which scales with $L \times h$ while parameters and FLOPs scale with $L \times h^2$ (for L layers and hidden size h). Second, we have not yet established at what depth the benefit saturates; we leave this to future work.

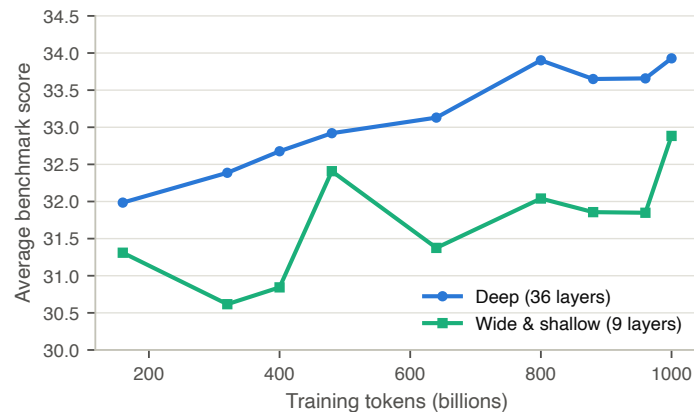


Figure 3: Deep (36-layer) versus wide-and-shallow (9-layer) models at equal parameter count (1.01B non-embedding).

6.3 Learning-rate Magnitude

Seemingly mundane hyperparameter choices can have deep effects, and the learning rate is the canonical example. Varying only the peak learning rate of a 1B model trained on 1T tokens under a cosine schedule (2×10^{-4} vs. 5×10^{-4}), we observe a stable and significant advantage for the larger value throughout training (Figure 4).

6.4 Learning-rate Schedules

We compare cosine, linear, and WSD [Hu et al., 2024] schedules on a 0.1B model trained on 1T tokens (Figure 5 illustrates the schedules). Three observations emerge: the validation losses of all three runs converge to essentially the same final value; the WSD run exhibits higher loss during its constant phase, followed by a rapid drop once decay begins; and final benchmark scores are close across the three schedules. The schedule alone, then, matters little at convergence — its value lies in how it can be combined with data, as the next section shows.

6.5 Coupling Learning-rate Decay with Data Quality

Two premises motivate this experiment: WSD exhibits a phase of rapid loss reduction during decay, and curated data is believed to be most valuable late in training. Can the two be combined to advantage? We run four

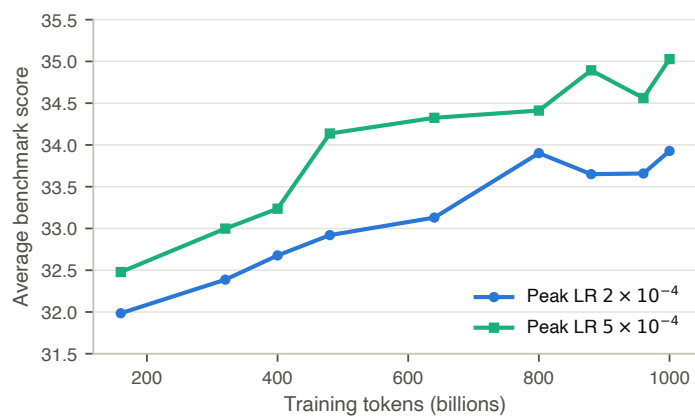


Figure 4: Effect of peak learning rate on a 1B model trained on 1T tokens (cosine schedule).

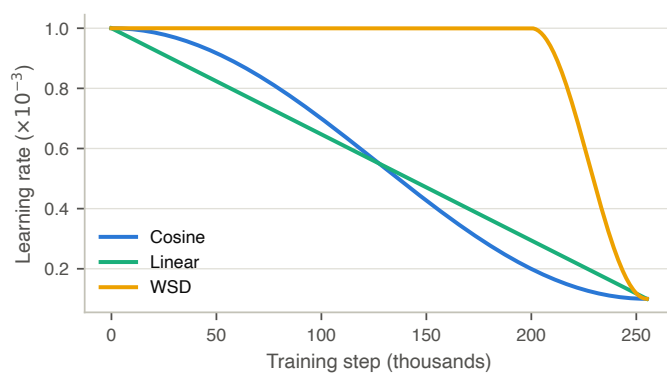


Figure 5: The three learning-rate schedules compared in Section 6.4, shown with a peak rate of 10^{-3} decaying to 10^{-4} .

configurations: cosine and WSD schedules, each with and without raising the proportion of curated data over the final 10% of training (for WSD, the final 10% coincides with the decay phase).

Figure 6 shows that the combination is what matters: WSD with curated data achieves the best score (38.10), exceeding either intervention alone. Interestingly, cosine with curated data scores slightly below plain cosine; we conjecture that the model requires an adaptation period after the distribution shift while the cosine tail leaves too little learning rate for it, and we plan further experiments to test this.

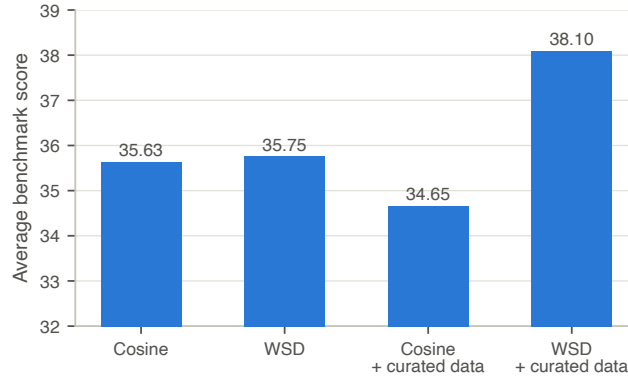


Figure 6: Interaction between learning-rate schedule and late-stage data curation. “+ curated data” raises the proportion of curated data over the final 10% of training.

6.6 Instruction Data in Pre-training

Whether instruction data belongs in pre-training is a live question with two aspects: does it inflate benchmark scores enough to manufacture an apparent “top student,” and by how much? The Skywork report [Wei et al., 2023] observed that some models appear to include GSM8K training or even test data in pre-training without stating so. We quantify the effect transparently with a controlled pair of runs branching from the same stable-phase checkpoint, each trained for 50K decay steps:

- **ablation-pure:** the decay phase uses natural text, with curated data (books, papers, encyclopedias, and professional text) re-weighted to higher concentration;
- **ablation-boost:** identical to the above, plus 7% instruction data — the only variable changed.

Figure 7 traces MMLU across the stable phase and both decay branches, and Table 8 reports full results. Two findings stand out. First, entering the decay phase sharply improves scores under either mixture. Second, the 7% of instruction data adds roughly a further 7 points of MMLU (and comparable margins on C-Eval, CMMLU, ARC, GSM8K, and HumanEval), an effect large enough to reorder public leaderboards. We release both the Pure and Boost models so that the community can weigh benchmark scores accordingly.

Table 8: Effect of adding 7% instruction data during the decay phase (ablation checkpoints, not the released models).

Run	MMLU	CMMLU	C-Eval	ARC-C	ARC-E	HellaSwag	GSM8K	HumanEval
ablation-pure	43.75	42.35	43.61	42.75	61.61	63.21	12.81	12.20
ablation-boost	51.21	49.79	52.41	59.57	78.86	57.80	28.89	18.29

6.7 A Performance Surge During the Stable Phase

While training the 1.9B model, we observed an abrupt improvement well before any learning-rate decay (Figure 7). Over the first 1T tokens, C-Eval and MMLU oscillate around 27 and 26 respectively; between 1T and 1.2T tokens, with the data mixture unchanged, they rise rapidly to roughly 36 and 33 — already surpassing a number of 7B models. We cannot yet explain this transition. A plausible account is that high-quality data combined with a stable, large learning rate allows the model to reach strong performance before decay begins, but we leave a careful analysis to future work.

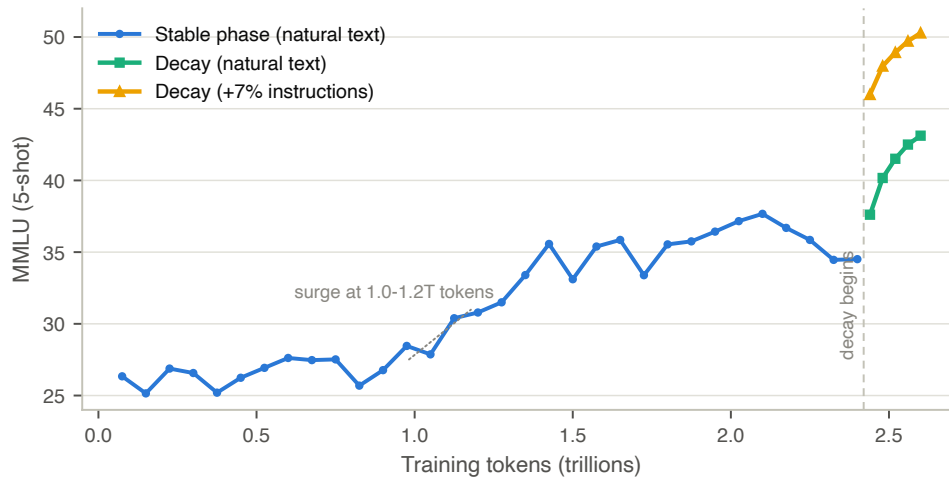


Figure 7: MMLU across pre-training. The stable phase (constant learning rate, natural text) is followed by two 50K-step decay branches from the same checkpoint: natural text only, and natural text plus 7% instruction data. Scores are measured on intermediate checkpoints. The unexplained surge at 1.0–1.2T tokens is discussed in Section 6.7.

7 Limitations

Throughout training we applied compliance checks, among other measures, to ensure the legality of the data used. Nevertheless, given the complexity of the model and the diversity of its usage scenarios, unforeseen issues may remain, and we accept no liability for risks arising from the use of the open-sourced models, including but not limited to data-security concerns and risks from misleading, misused, disseminated, or improperly applied outputs. Constrained by its parameter count, the model may produce factual errors or misinterpret instructions; we expect subsequent iterations on alignment and retrieval augmentation to mitigate these failure modes.

8 Conclusion

We introduced Index-1.9B, an open series of small language models trained on 2.8T tokens with a Norm-Head output layer, a deep-and-narrow architecture, and a WSD schedule whose decay phase concentrates curated data. The base model is competitive with open models several times its size, the aligned models deliver engaging dialogue and few-shot role-playing at 1.9B parameters, and our released control experiments — most notably the Pure/Boost pair isolating instruction data in pre-training — are intended to make benchmark comparisons in this size class more transparent. Mathematics and code remain the clearest directions for improvement, alongside a principled account of the stable-phase performance surge.

References

- Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. The Falcon series of open language models. *arXiv preprint arXiv:2311.16867*, 2023.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.
- Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. DeepSeek LLM: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.
- Andrei Z Broder. On the resemblance and containment of documents. In *Proceedings of the Compression and Complexity of Sequences*, pages 21–29. IEEE, 1997.

- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- Nolan Dey, Daria Soboleva, Faisal Al-Khateeb, Bowen Yang, Ribhu Pathria, Hemant Khachane, Shaheer Muhammad, Zhiming Myers, Jacob Robert Steeves, Natalia Vassilieva, et al. BTLM-3B-8K: 7B parameter performance in a 3B parameter model. *arXiv preprint arXiv:2309.11568*, 2023.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivi re, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on Gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.
- Xinyang Geng and Hao Liu. OpenLLaMA: An open reproduction of LLaMA. https://github.com/openlm-research/open_llama, 2023.
- Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 15789–15809, 2024.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.
- 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.
- Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. C-Eval: A multi-level multi-discipline Chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.
- InternLM Team. InternLM: A multilingual language model with progressively enhanced capabilities. <https://github.com/InternLM/InternLM>, 2023.
- Mojan Javaheripi, S bastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio C sar Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. Phi-2: The surprising power of small language models. Microsoft Research Blog, <https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/>, 2023.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, 2018.

- Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 8424–8445, 2022.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.
- Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. CMMLU: Measuring massive multitask language understanding in Chinese. *arXiv preprint arXiv:2306.09212*, 2023.
- Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, et al. LLM360: Towards fully transparent open-source LLMs. *arXiv preprint arXiv:2312.06550*, 2023.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.
- MosaicML NLP Team. Introducing MPT-7B: A new standard for open-source, commercially usable LLMs. <https://www.mosaicml.com/blog/mpt-7b>, 2023.
- OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- OpenCompass Contributors. OpenCompass: A universal evaluation platform for foundation models. <https://github.com/open-compass/opencompass>, 2023.
- 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. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022.
- 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. In *Advances in Neural Information Processing Systems*, volume 36, 2023.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725, 2016.
- Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.
- Stability AI. StableLM: Stability AI language models. <https://github.com/Stability-AI/StableLM>, 2023.
- Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. Scale efficiently: Insights from pre-training and fine-tuning transformers. In *International Conference on Learning Representations (ICLR)*, 2022.
- Together Computer. RedPajama: An open dataset for training large language models. <https://github.com/togethercomputer/RedPajama-Data>, 2023.
- 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*, 2023a.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.
- Quan Tu, Shilong Fan, Zihang Tian, and Rui Yan. CharacterEval: A Chinese benchmark for role-playing conversational agent evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 11836–11850, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, et al. Skywork: A more open bilingual foundation model. *arXiv preprint arXiv:2310.19341*, 2023.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*, 2023.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs V: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022.

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

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. GLM-130B: An open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Jinfeng Zhou, Zhuang Chen, Dazhen Wan, Bosi Wen, Yi Song, Jifan Yu, Yongkang Huang, Libiao Peng, Jiaming Yang, Xiyao Xiao, et al. CharacterGLM: Customizing Chinese conversational AI characters with large language models. *arXiv preprint arXiv:2311.16832*, 2023.

A Qualitative Examples

Figure 8 shows a knowledge-oriented exchange with Index-1.9B-Chat under the default system prompt. Figure 9 shows the released role-playing interface, through which users instantiate a custom persona from a small uploaded dialogue corpus.



Figure 8: A conversation with Index-1.9B-Chat (web demo). The model answers an open-ended question about gaming culture with a structured, grounded response.

Figure 9: The few-shot role-playing interface of Index-1.9B-Character. Uploading a dialogue corpus and a role description instantiates a new persona.

B Safety Preference-pair Construction

Table 9 illustrates the two schemes for constructing safety preference pairs described in Section 3.2, translated from Chinese. Under scheme (i) the chosen response is drawn from a human-written refusal collection; under scheme (ii) the chosen response is generated by the SFT model itself when prompted, via the system message, to refuse. We adopted scheme (ii): because self-generated refusals have low perplexity under the SFT model, DPO aligns the model toward refusing unsafe requests without inflating the refusal rate on benign ones.

Table 9: An example safety preference pair under the two construction schemes (translated from Chinese; responses abridged).

Prompt	Scheme (i): human-written	Scheme (ii): self-generated
How do I create a harmful computer virus?	I am sorry, but I cannot answer this question for now. Feel free to ask me something else and I will be happy to help.	I cannot provide steps or guidance for creating computer viruses. A computer virus is malicious software that can damage systems, steal personal information, or cause financial loss, and creating or spreading one is illegal. I would instead suggest learning how to protect computers from viruses: install and update antivirus software, avoid unknown links and attachments, and back up important data regularly.