

NVIDIA Nemotron Nano 2: An Accurate and Efficient Hybrid Mamba-Transformer Reasoning Model

NVIDIA

Abstract. We introduce Nemotron-Nano-9B-v2, a hybrid Mamba-Transformer language model designed to increase throughput for reasoning workloads while achieving state-of-the-art accuracy compared to similarly-sized models. Nemotron-Nano-9B-v2 builds on the Nemotron-H architecture, in which the majority of the self-attention layers in the common Transformer architecture are replaced with Mamba-2 layers, to achieve improved inference speed when generating the long thinking traces needed for reasoning. We create Nemotron-Nano-9B-v2 by first pre-training a 12-billion-parameter model (Nemotron-Nano-12B-v2-Base) on 20 trillion tokens using an FP8 training recipe. After aligning Nemotron-Nano-12B-v2-Base, we employ the Minitron strategy to compress and distill the model with the goal of enabling inference on up to 128k tokens on a single NVIDIA A10G GPU (22GiB of memory, `bfloat16` precision). Compared to existing similarly-sized models (e.g., Qwen3-8B), we show that Nemotron-Nano-9B-v2 achieves on-par or better accuracy on reasoning benchmarks while achieving up to $6\times$ higher inference throughput in reasoning settings like 8k input and 16k output tokens (Figure 1). We are releasing Nemotron-Nano-9B-v2, Nemotron-Nano-12B-v2-Base, and Nemotron-Nano-9B-v2-Base checkpoints along with the majority of our pre- and post-training datasets on Hugging Face.

1. Introduction

We introduce NVIDIA Nemotron Nano 2, a hybrid Mamba-Transformer reasoning model (Waleffe et al., 2024; Lieber et al., 2024; DeepMind, 2025; NVIDIA, 2025) that achieves on-par or better benchmark accuracies at $3\times$ – $6\times$ higher throughput than Qwen3-8B (Yang et al., 2025) for generation-heavy scenarios like 1k input / 8k output or 8k input / 16k output tokens (Figure 1). Nemotron Nano 2 builds on the architecture of Nemotron-H (NVIDIA, 2025), but utilizes key new datasets and recipes for pre-training, alignment, pruning and distillation. We share these recipes, the checkpoints, as well as the majority of the pre- and post-training datasets.

The initial base model, Nemotron-Nano-12B-v2-Base, was pre-trained using FP8 precision (§2.4) over 20 trillion tokens using a Warmup-Stable-Decay (Hu et al., 2024) learning rate schedule (§2.5). It then underwent a continuous pre-training long-context extension phase to become 128k-capable without degrading other benchmarks (§2.6). Overall, new and improved datasets led to significant accuracy improvements over Nemotron-H-8B on math, multilingual, MMLU-Pro and other benchmarks (§2.2).

Nemotron Nano 2 was then post-trained through a combination of Supervised Fine-Tuning (SFT), Group Relative Policy Optimization (GRPO) (Shao et al., 2024), Direct Preference Optimization (DPO) (Rafailov et al., 2023), and Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022; Christiano et al., 2017). We applied multiple SFT stages across various domains, followed by targeted SFT on key areas such as tool use, long-context performance, and truncated (budgeted) training. GRPO and RLHF sharpened instruction-following and conversational ability, while additional DPO stages further strengthened tool use. Overall, post-training was performed on roughly 90 billion tokens, the majority in single-turn prompt–response format with reasoning

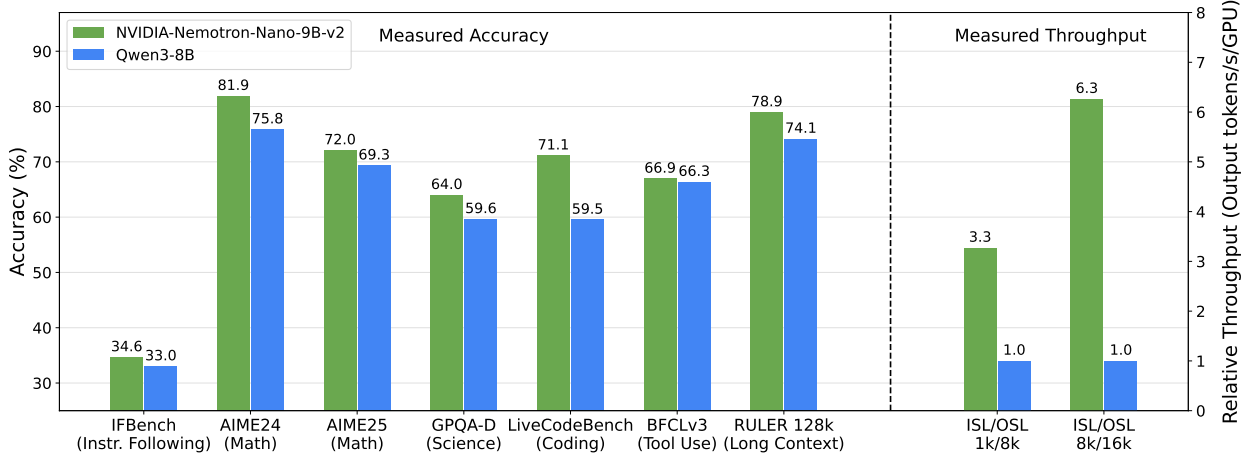


Figure 1 | Comparison of Nemotron Nano 2 and Qwen3-8B in terms of accuracy and throughput. Nemotron Nano 2 achieves comparable or better accuracies on complex reasoning benchmarks, while achieving up to $6.3\times$ higher throughput for such workloads. We abbreviate input sequence length to ISL and output sequence length to OSL and measure throughput on a single A10G GPU in `bfloat16`.

traces. About 5% of the data contained deliberately truncated reasoning traces, enabling fine-grained thinking budget control at inference time (§3.4).

Finally, both the base model and aligned model were compressed so as to enable inference over context lengths of 128k tokens on a single NVIDIA A10G GPU (22 GiB of memory, `bfloat16` precision). This was done by extending a compression strategy based on Minitron (Muralidharan et al., 2024; Sreenivas et al., 2024; Taghibakhshi et al., 2025) to compress reasoning models subject to constraints.

We are releasing the following models on Hugging Face:

- **NVIDIA-Nemotron-Nano-9B-v2**: the aligned and pruned reasoning model,
- **NVIDIA-Nemotron-Nano-9B-v2-Base**: a pruned base model,
- **NVIDIA-Nemotron-Nano-12B-v2-Base**: the base model before alignment or pruning.

Additionally, we are releasing the majority of our pre-training dataset in the **Nemotron-Pre-Training-Dataset-v1** collection of more than 6 trillion tokens:

- **Nemotron-CC-v2**: Follow-up to Nemotron-CC (Su et al., 2025) with eight additional Common Crawl snapshots (2024–2025), synthetic rephrasing, deduplication, and synthetic Q&A data translated into 15 languages.
- **Nemotron-CC-Math-v1**: 133B-token math dataset from Common Crawl using Lynx + LLM pipeline (Mahabadi et al., 2025). Preserves equations, standardizes to LaTeX, outperforms previous math datasets on benchmarks.
- **Nemotron-Pretraining-Code-v1**: Curated GitHub code references with multi-stage filtering, deduplication, and quality filters. Includes code Q&A data in 11 programming languages.
- **Nemotron-Pretraining-SFT-v1**: Synthetic SFT-style dataset covering STEM, multilingual, academic, and reasoning domains.

Finally, we are releasing an updated post-training dataset:

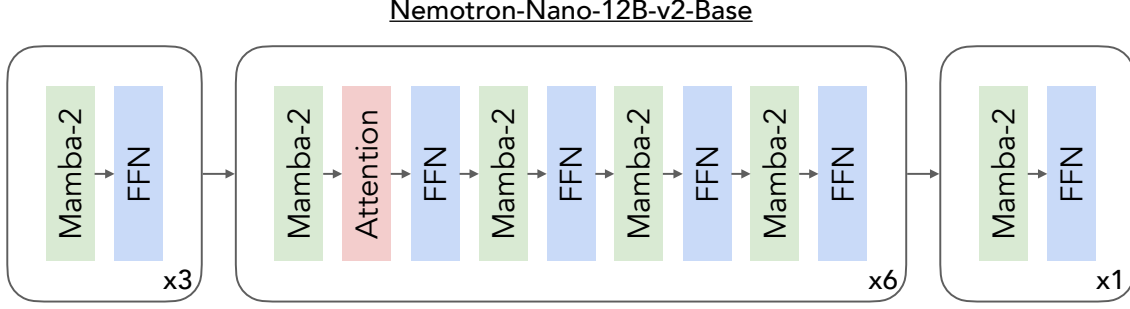


Figure 2 | Nemotron-Nano-12B-v2-Base layer pattern. As in Nemotron-H models, roughly 8% of the total layers in the model are self-attention layers which are evenly dispersed throughout the model.

Model	Number of layers	Model dimension	FFN dimension	Q heads	KV heads	State dimension	Mamba groups
Nemotron-Nano-12B-v2-Base	62	5120	20480	40	8	128	8

Table 1 | Summary of Nemotron-Nano-12B-v2-Base architecture.

- **Nemotron-Post-Training-Dataset-v2:** Adds to NVIDIA’s post-training dataset releases with an extension of SFT and RL data into five target languages: Spanish, French, German, Italian and Japanese. The data supports improvements of math, code, general reasoning, and instruction following capabilities.

The rest of this technical report is organized as follows: In §2, we discuss the Nemotron Nano 2 model architecture, pre-training process, and base model evaluation results. In §3, we discuss the alignment process. In §4, we describe the pruning and distillation methods used for model compression.

2. Pretraining

In this section, we discuss the architecture and pretraining of the Nemotron-Nano-12B-v2-Base model. We also compare this model against other state-of-the-art models in terms of accuracy on popular benchmarks.

2.1. Model Architecture

As in Nemotron-H (NVIDIA, 2025), Nemotron-Nano-12B-v2-Base consists of a mixture of Mamba-2 (Dao & Gu, 2024), self-attention, and FFN layers. The layer pattern and key architecture details are summarized in Figure 2 and Table 1. Concretely, we use 62 layers, with 6 of them being self-attention layers, 28 being FFN, and 28 being Mamba-2 layers. We use a hidden dimension of 5120, FFN hidden dimension of 20480, and Grouped-Query Attention (Ainslie et al., 2023) with 40 query heads and 8 key-value heads. For Mamba-2 layers, we use 8 groups, a state dimension of 128, a head dimension of 64, an expansion factor of 2, and a window size for convolution of 4. For FFN layers, we use squared ReLU (So et al., 2022) activation. Again as in Nemotron-H, we do not use any position embeddings and use RMSNorm (Zhang & Sennrich, 2019), separate embedding and output layer weights, no dropout, and we do not use bias weights for linear layers.

2.2. Pre-Training Data

Nemotron-Nano-12B-v2-Base was pre-trained on a large corpus of high-quality curated and synthetically-generated data.

2.2.1. Curated Data

We have separate data curation pipelines for the following broad data categories: general web crawl data (English and multilingual), math data, and code data. We discuss each in turn next.

English web crawl data. We used the Nemotron-CC dataset (Su et al., 2025), but updated to include eight more recent Common Crawl snapshots (CC-MAIN-2024-33 through CC-MAIN-2025-13) using the same pipeline. For synthetic rephrasing, we mostly switched to Qwen3-30B-A3B (from Mistral Nemo 12B). Additionally, we used data from CC-NEWS through April 23, 2025, to help improve the knowledge cutoff of the model. The CC-NEWS data was filtered for English and globally fuzzily de-duplicated; no other filtering was used.

Multilingual data. We extracted data for fifteen languages from the following three Common Crawl snapshots: CC-MAIN-2024-51, CC-MAIN-2025-08, and CC-MAIN-2025-18. The fifteen languages included were Arabic, Chinese, Danish, Dutch, French, German, Italian, Japanese, Korean, Polish, Portuguese, Russian, Spanish, Swedish, and Thai. As we did not have reliable multilingual model-based quality classifiers available, we just applied heuristic filtering instead. This was done in a similar manner to the filtering of low-quality English data in the Nemotron-CC pipeline, except that we had to selectively disable some heuristic filters that had very high false positive rates for some languages. De-duplication was done in the same way as for Nemotron-CC. Additionally, we used data from Wikipedia and FineWeb-2 (Penedo et al., 2025) for these fifteen languages.

Math data. Mathematical content on the web is expressed in a wide range of formats, including inline and block $\text{\LaTeX}$, MathML, Unicode symbols, and custom renderers such as MathJax or KaTeX. We conducted a detailed analysis of prior math-specific extraction pipelines—including OpenWebMath (Paster et al., 2023), MegaMath (Zhou et al., 2025), jusText (Endr dy & Nov k, 2013), Trafilatura (Barbaresi, 2021), and Resiliparse (Bevendorff et al., 2018)—and found that none could reliably preserve mathematical expressions or code structure. These tools frequently discard or distort equations and flatten code formatting, severely limiting the utility of the extracted content for pretraining.

To address this, we built a new pipeline specifically designed for high-fidelity mathematical extraction from Common Crawl. We first aggregated a comprehensive list of math-related URLs from prior datasets (e.g., InfiMM-WebMath (Han et al., 2024), OpenWebMath (Paster et al., 2023), FineMath (Allal et al., 2025), and MegaMath (Zhou et al., 2025)), then re-fetched their raw HTML documents from 98 Common Crawl snapshots (2014–2024). Each page was rendered using the `lynx` text-based browser to preserve layout and math structure. We then applied Phi-4 (Abdin et al., 2024)(14B-parameters) to remove boilerplate, standardize notation into $\text{\LaTeX}$, and correct inconsistencies. A FineMath classifier (Allal et al., 2025) was used to retain high-quality documents, followed by fuzzy deduplication via MinHash-based (Broder, 2000) Locality Sensitive Hashing (LSH) (Indyk & Motwani, 1998) via the NeMo-Curator framework.¹ We finally decontaminated the dataset using LLM Decontaminator (Yang et al., 2023).

This process resulted in a 133B-token corpus, Nemotron-CC-Math-3+, and a higher-quality 52B-token subset, Nemotron-CC-Math-4+, containing only the top-scoring samples. When used for pretraining, this dataset yields substantial improvements across math (MATH-500), code (HumanEval+, MBPP+,

¹<https://github.com/NVIDIA-NeMo/Curator>

MBPP), and general-domain evaluations (MMLU, MMLU-STEM, MMLU-Pro), surpassing all existing open math datasets. For full details, see [Mahabadi et al. \(2025\)](#).

Code data. In line with previous models in the Nemotron family ([NVIDIA, 2025, 2024](#); [Parmar et al., 2024](#)), we pretrained Nemotron-Nano-12B-v2-Base with large-scale raw source code. All source code used to train this model originated from GitHub and went through a multi-stage processing pipeline to arrive at the final source code training data. We performed license-based removal with a license detection pipeline similar to that used by the BigCode project ([Lozhkov et al., 2024](#)), but with fewer accepted licenses (see Appendix A for additional details). De-duplication is especially important for source code, where many files can be found exactly duplicated across numerous repositories. Consequently we performed both exact (via hashing) and fuzzy deduplication (using MinHash LSH). In order to build a better understanding of each file in our dataset, we annotated all files with a variety of measures and then performed filtering using these annotations. We found the heuristic filters from OpenCoder ([Huang et al., 2025](#)) to be effective and leveraged them to filter files that are less valuable or even detrimental for LLM pretraining.

2.2.2. Synthetically-Generated Data

STEM data. We generated synthetic data for STEM subjects, including Astronomy, Biology, Chemistry, Math, and Physics using 88.6k questions collected from multiple sources as the seed data. In addition to the widely used GSM8K, MATH, and AOPS training sets, we collected more diverse questions from Stemez² and textbooks with permissive licenses from OpenStax³ and Open Textbook Library.⁴ We used Qwen2.5-VL-72B-Instruct ([Bai et al., 2025](#)) to extract questions from the exercise sections in the textbooks with additional instructions such as dropping question numbering, ignoring questions that require image interpretation, and formatting equations using LaTeX. We manually curated the extracted questions to fix occasional OCR errors and removed non-self-contained questions (e.g., a question that refers to an example in the same chapter).

To expand both the quantity and diversity of questions, we conducted three iterations of question generation using four models (i.e., Qwen3-30B-A3B and Qwen3-235B-A22B ([Yang et al., 2025](#)), both with thinking mode enabled, Deepseek-R1 ([DeepSeek-AI, 2025a](#)), and Deepseek V3 ([DeepSeek-AI, 2025b](#))) and three prompts:

1. **Similar question:** Create a new question that explores similar concepts but offers a fresh challenge.
2. **Harder question:** Create a new question that requires more logical steps or involves more advanced concepts.
3. **Varied question:** Create a new question that differs in type from the original question. We instructed the model to avoid superficial or trivial modifications and think through the solution when creating a new question.

We filtered out duplicates and highly-similar questions using fuzzy de-duplication and generated solutions to the remaining questions with the models used in the question generation step. We converted a subset of examples to multiple-choice questions in MMLU or MMLU-Pro style. We constructed a few thousand few-shot examples by concatenating random synthetic samples.

Math data. We also revisited and regenerated the Nemotron-MIND dataset ([Akter et al., 2024](#)), a math-informed synthetic pretraining corpus originally built on OpenWebMath. In our updated

²<https://www.stemez.com/>

³<https://openstax.org>

⁴<https://open.umn.edu/opentextbooks/>

version, we regenerated the MIND dataset using Nemotron-CC-Math-4+, our highest-quality math subset comprising 52B tokens—as the source corpus. Following the original methodology, we applied seven prompt templates (e.g., Teacher–Student, Debate, Interview, etc) to generate structured mathematical dialogues using the Phi-4 model. Unlike the original MIND, which relied on 14.7B tokens of lower-fidelity data, our version leverages significantly higher-quality input and processes it with a chunk size of 5K tokens. This regeneration produced a 73B-token synthetic dataset and led to consistent improvements across math reasoning and general knowledge (MMLU, MMLU-Pro, MMLU-Stem) benchmarks compared to the original MIND version, highlighting the critical role of input data quality. Full details and results are available in [Mahabadi et al. \(2025\)](#).

Multilingual data. We generated multilingual diverse question and answer data (Diverse QA) ([Su et al., 2025](#)) from two sources:

1. We translated the English Diverse QA data to fifteen languages (see Multilingual data) using Qwen3-30B-A3B ([Yang et al., 2025](#)).
2. We generated synthetic data from Wikipedia articles in these languages using the Diverse QA prompt and instructed the model to write all questions and answers in the target language.

In addition, we translated a subset of our GSM8K augmentation data (see STEM data) into these languages using Qwen3-30B-A3B. We post-processed each translated solution by appending a concluding sentence meaning “*The answer is ...*” (e.g., “*La respuesta es ...*” in Spanish, “*Die Antwort lautet ...*” in German), where the final numerical answer is extracted from the original English solution.

Code data. We generated question-answer (QA) data at scale for 11 different programming languages by prompting an LLM to generate questions based on short snippets from our curated source code, asking the model to solve the generated question, and then performing post hoc filtering of the generated QA pairs based on heuristics as appropriate (e.g., Python AST parsing). This technique results in diverse synthetic data targeted at problem solving containing both natural language and source code. Further details are covered in the Nemotron-H technical report ([NVIDIA, 2025](#)), where we first leveraged this type of synthetic code data in pretraining.

Academic data. In the pretraining set for the Nemotron-H ([NVIDIA, 2025](#)) series of models, we assigned attribute labels for educational quality, educational difficulty, and educational subject to all documents coming from academic data, which encompasses textbooks and academic papers. As content of higher educational difficulty in technical domains still proves challenging for models, we prioritized increasing model comprehension of such information in our current pretraining set via the generation of question-answer (QA) pairs as such data has been shown to enhance knowledge storage and extraction within language models ([Allen-Zhu & Li, 2024](#)).

To do so, we first gathered all documents with educational difficulty at the undergraduate and graduate levels in the following technical subject areas: math, chemistry, biology, physics, and medicine. Using this subset of documents, we aim to find the most relevant pieces of texts that could be utilized as seed contexts for our generation of QA pairs. We chunk each document into snippets of 512 token lengths, embed them with the e5-large model ([Wang et al., 2024](#)), and store them within a Milvus vector database that enables approximate nearest neighbor search. We then curate documents from a set of complex subject areas (e.g. Mathematics: Real Analysis, Biology: Genetics, Statistics: Information Theory), and query the Milvus database for the 250 nearest neighbor text snippets to each query document. The returned snippets function as our seed contexts that we then pass into a Qwen-2.5 72B instruct model ([Qwen, 2025](#)) to generate multiple choice and free response style QA pairs based on the information contained in the snippet. With each QA pair, a justification for the answer is additionally generated.

SFT-style data. Using SFT-style data in the later stages of pretraining has shown to be helpful to foster more comprehensive model learning (Hu et al., 2024).

Therefore, we synthesized and included different SFT-style data covering several domains: 1) code SFT data which is mainly focused on solving code problems; 2) math SFT data that is mostly focused on reasoning; 3) MMLU-style SFT data which contains different question and answer examples covering different knowledge topics; and 4) general instruction following SFT data.

We ensure that the SFT-style data covers diverse topics with different difficulty levels for each of the above mentioned domains. Detailed synthesis methods and pipelines for the above mentioned SFT data can be found in prior work (Toshniwal et al., 2024; Moshkov et al., 2025; Bercovich et al., 2025a,b; Ahmad et al., 2025b,a; Majumdar et al., 2024).

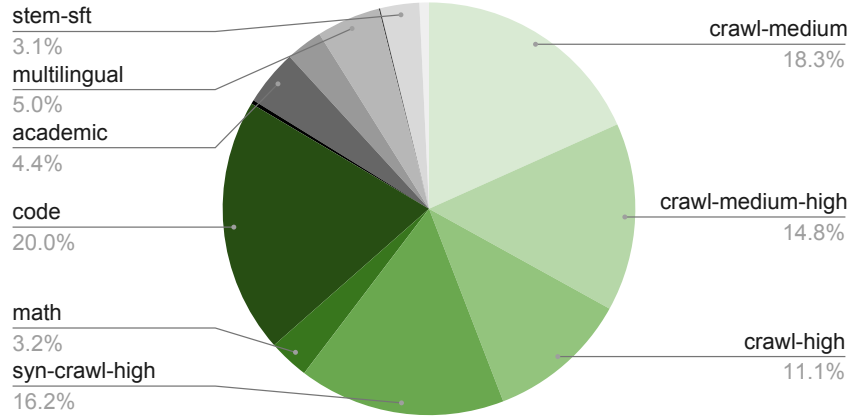
Fundamental reasoning SFT-style data. While the above mentioned SFT-style data help enhance an LLM’s ability to answer questions in code, math and general language understanding benchmarks, they do not help improve the model’s ability in deeper reasoning tasks to discern the correct answer among a larger pool of potential distractors. We propose to mitigate that by synthesizing SFT-style data focused on analytical reasoning, logical reasoning, and reading comprehension.

Specifically, we collected existing datasets including 1) the Law School Admission Test (LSAT) dataset from Wang et al. (2022); Zhong et al. (2022) which encompasses three tasks: logical reasoning, reading comprehension, and analytical reasoning, 2) the repurposed LogiQA dataset by Liu et al. (2020) which contains various types of logical reasoning questions collected from the National Civil Servants Examination of China, and 3) the AQuA-RAT dataset which emphasizes algebraic word problems by Ling et al. (2017). We then prompted DeepSeek-V3 (DeepSeek-AI, 2025b) and Qwen3-30B-A3B (Yang et al., 2025) respectively to synthesize more similar questions with corresponding options. For each question we generated, we prompted DeepSeek-V3 again to generate the chain-of-thought (CoT) process with the final solution. At the post-processing stage, we apply majority voting to keep only the samples that have the most voted solutions. Overall, we generated 4B tokens from DeepSeek-V3 and 4.2B tokens from Qwen3-30B models.

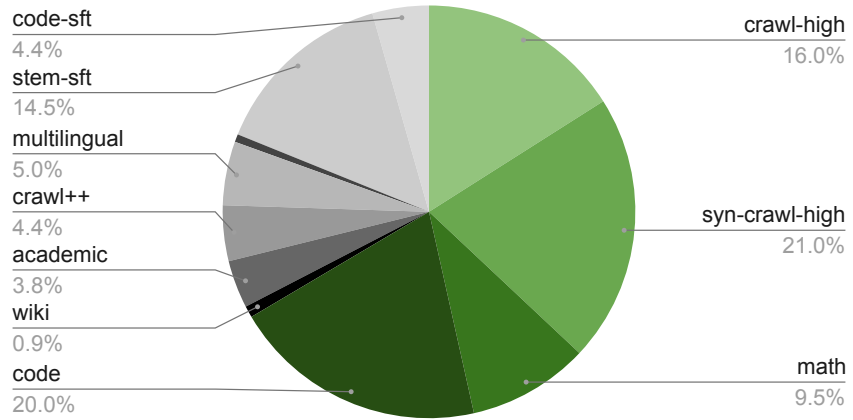
2.3. Data Mixture and Ordering

Our data mixture consists of thirteen data categories. The largest is web crawl data, which we subdivided into four categories based on the Nemotron-CC quality classification (Su et al., 2025): crawl-medium, crawl-medium-high, crawl-high, syn-crawl-high denoting medium, medium-high, high and synthetic quality crawl data, respectively. Apart from these, our data mixture has additional categories such as math, wikipedia, code, academic data, crawl++, multilingual, and synthetic SFT-style data which is further categorized as general-sft, stem-sft and code-sft. Crawl++ consists of web-crawl derivatives like OpenWebText, BigScience and Reddit. Our multilingual data has fifteen languages: Arabic, Danish, German, Spanish, French, Italian, Portuguese, Dutch, Polish, Swedish, Thai, Chinese, Japanese, Korean, and Russian. We design the data mixtures to give similar weight to data sources that have similar quality. Data sources of higher quality are weighed higher than data sources of lower quality. We provide detailed explanation on quality estimation of datasets and the blend creation process in Feng et al. (2024) and NVIDIA (2025).

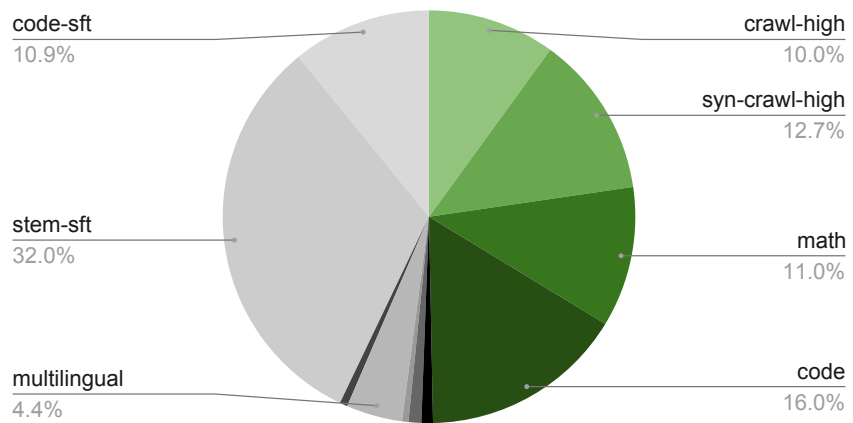
We used a curriculum based on three phases of data-blending approach to pre-train Nemotron-Nano-12B-v2-Base. In the first phase, we used a data mixture that promotes diversity in data; in the second and third phases, we primarily used high-quality datasets (e.g., Wikipedia). We switched to the second phase at the 60% point of training, and to the third phase at the 90% point of training. The data mixtures used in each phase are shown in Figure 3.



(a) Data mixture of Phase 1.



(b) Data mixture of Phase 2.



(c) Data mixture of Phase 3.

Figure 3 | Data mixtures for each phase of pre-training.

Multilingual Data	Avg	Sp	Ge	Fr	Ma	It	Ja	Po	Ko
Common Crawl	37.0	37.8	36.5	39.8	34.3	36.3	35.3	37.5	38.8
FineWeb-2	35.1	38.8	35.0	34.3	31.5	37.0	33.0	36.0	35.3
DiverseQA-wiki	42.1	44.8	41.3	41.8	41.5	44.0	41.0	42.3	40.3
DiverseQA-crawl	47.0	49.8	50.8	48.3	46.0	45.8	44.5	49.0	42.0

Table 2 | Comparison of multilingual datasets on the Global-MMLU Benchmark.

2.3.1. Multilingual Data Ablation Study

In Section 2.2, we mentioned several large categories of multilingual data, both curated and synthetic:

1. **Common Crawl:** Extracted from recent Common Crawl snapshots using our own pipeline.
2. **FineWeb-2** (Penedo et al., 2025).
3. **DiverseQA-wiki:** Generated from multilingual Wikipedia articles using a translated Diverse QA prompt.
4. **DiverseQA-crawl:** Translated from English Diverse QA data.

In order to decide the proper data mixture among these different multilingual data sources, we first conducted ablation experiments to compare the four multilingual data’s downstream tasks’ performance.

Specifically, we took a 1B model checkpoint that had been trained for 350B tokens, and continuous pretrained it for another 100B tokens. We assigned 50% of the continuous pretraining data to multilingual data, and the remaining 50% use our default pretraining data mixture. We evaluated each model’s performance using the Global-MMLU benchmark (Singh et al., 2024a); the results are shown in Table 2. Our curated Common Crawl-based multilingual data performed slightly better than the Fineweb2-based multilingual data, while the synthesized multilingual QA pairs performed much better than the curated multilingual web crawl data. The diverse pairs translated from English Common Crawl achieved the highest average score over the 8 languages we evaluated on. Therefore, we assigned a much higher weight to the DiverseQA-crawl data than the other categories when deciding our multilingual data mixture.

2.3.2. Fundamental Reasoning SFT-Style Data Ablation Study

To show the effectiveness of the fundamental reasoning (FR) focused SFT-style data we introduced in Section 2.2, we took the Nemotron-H-8B (NVIDIA, 2025) intermediate checkpoint trained over 14.5T tokens, and continuous pretrained it with another 100B tokens. We assigned 5% of the 100B tokens to the newly synthesized FR-SFT data (as a replacement for Common Crawl data), and kept all other data categories the same as in the Nemotron-H-8B’s phase 3 blend. We compared this model with Nemotron-H-8B, which had also been trained with 14.6T tokens. The detailed evaluation benchmarks are introduced in Section 2.7. The comparison results are shown in Table 3. The SFT-style data helped improve the Nemotron-H 8B model’s performance on MMLU-Pro from 44.24 to 56.36, and also helped increase the average MATH score by around 2 points. While MMLU-Pro is a more challenging benchmark that evaluates a model’s language understanding capability, it also requires the model to have excellent reasoning capability to select the correct answer out of ten choices. Our SFT data helps equip the model to select the correct answers from the other nine distractors through fundamental reasoning. We noticed no decrease in the average commonsense reasoning and average code benchmarks.

Model	Avg Math	Avg Code	Avg Reasoning	MMLU	MMLU-Pro
Nemotron-H 8B	37.92	59.49	71.79	72.67	44.24
Nemotron-H 8B (w/ FR-SFT data)	39.70	59.61	71.43	72.98	56.36

Table 3 | Ablation study of the Fundamental Reasoning (FR) focused SFT-style data.

2.4. FP8 Recipe

We used DeepSeek’s FP8 training recipe for the entirety of the pretraining run (DeepSeek-AI, 2025b). Specifically, we used E4M3 for all tensors, 128x128 quantization blocks for weights, and 1x128 tiles for the activations. Unlike Nemotron-H, we natively kept the model weights in E4M3 so that we could do the distributed optimizer’s parameter all-gather operations (across data-parallel replicas) in FP8; master weights are still kept in FP32. One exception to DeepSeek’s formula was that we left the first and last four linear layers in BF16, as done with Nemotron-H. Also unlike the DeepSeek-V3 run, we left all optimizer state in FP32. We observed no training instabilities from this choice of numerics.

2.5. Hyperparameters

We trained Nemotron-Nano-12B-v2-Base on a token horizon of 20 trillion tokens. We used a sequence length of 8192 and global batch size of 768 (6,029,312 tokens per batch). We did not use any batch size ramp-up. We used a WSD (Warmup-Stable-Decay) (Hu et al., 2024) learning rate schedule with a “stable” learning rate of $4.5 \cdot 10^{-4}$ and a minimum value of $4.5 \cdot 10^{-6}$; the learning rate was decayed over the final 3.6 trillion tokens. Weight decay was set to 0.1, and Adam β_1 and β_2 were set to 0.9 and 0.95 respectively.

2.6. Long-Context Extension

To ensure Nemotron-Nano-12B-v2-Base can infer over long context windows, we added a long-context phase (Phase LC) after Phase 3 of pre-training. In Phase LC, we did continuous pretraining (CPT) with a context length of 524,288 (512k) tokens using a constant learning rate of $4.5 \cdot 10^{-6}$. Although the target context length of Nemotron Nano 2 is 128k, in preliminary studies on the Nemotron-H 8B model, we found it better to do CPT with 512k sequence length, instead of 256k or 128k. Our intuition is that longer training sequence can effectively lower the chance of long coherent documents being cut and separated by the Concat & Chunk algorithm for pretraining data loading. We used 8-way tensor model parallelism and 16-way context parallelism to ensure training with sequence lengths of 512k tokens still fits in GPU memory. We used a global batch size of 12 to ensure the total number of tokens per global batch during long-context CPT is the same as during pretraining: around 6M tokens. Phase LC consisted of 18.9 billion tokens.

Additionally, we did long-context synthetic data generation to create more high-quality data for Phase LC. Since the academic pretraining dataset is a good source of coherent long-context documents, we used such documents that are longer than 32k tokens as seed data. We followed the methods mentioned in the Llama-3 (Meta, 2024) and Qwen-2.5 (Qwen, 2025) tech reports to generate long-context document QA data. We split each document into chunks of 1,024 tokens and then randomly selected 10% of the chunks to be fed into Qwen-2.5-72B-Instruct for data synthesis. We asked the generator to generate a QA pair based on the information in the text chunk. We concatenated the QA pairs and appended them to the end of the original document as a sample of the long-context document QA data. Such long-document QA provided good material for the model to learn long-

context dependencies. See Table 4 for ablation results on Nemotron-H 8B regarding train sequence lengths and the effects of synthetic data.

The data blend used in Phase LC was built based on that of Phase 3. We proportionally downscaled the weights of all Phase 3 data to 80% of their original values, allocating the remaining 20% to the newly added long-context document-QA data. We found such a blend could effectively extend the context length of Nemotron-Nano-12B-v2-Base without degrading regular benchmark scores.

Train length	128k	256k	256k	512k
Synthetic data	yes	no	yes	yes
RULER-128k	73.68	70.19	79.04	81.04

Table 4 | Comparisons of different train sequence lengths and synthetic data usages. Ablations were conducted on Nemotron-H 8B.

Task	N-Nano-V2 12B Base	N-Nano-V2 9B Base	Qwen3 8B Base	Gemma3 12B Base
General				
MMLU	78.24	74.53	76.44	73.61
MMLU-Pro 5-shot	63.98	59.43	56.27	45.12
AGIEval English CoT	68.03	65.28	59.54	51.69
Math				
GSM8K CoT	91.66	91.36	84.00	74.45
MATH	83.54	80.50	55.40	42.40
MATH Level 5	67.61	63.64	29.91	17.71
AIME 2024 pass@32	56.67	30.00	20.00	16.67
Code				
HumanEval+ avg@32	61.03	58.50	57.55	36.68
MBPP+ avg@32	61.55	58.95	58.56	51.73
Commonsense Understanding				
ARC Challenge	93.26	90.70	93.09	90.44
HellaSwag	84.00	79.90	79.75	84.15
OpenBookQA	46.00	44.80	42.00	46.00
PIQA	82.54	81.83	79.43	82.10
WinoGrande	79.24	75.30	75.93	79.95
Long Context				
RULER-128K	84.74	82.22	-	80.70

Table 5 | Accuracy of Nemotron-Nano-V2-Base models versus existing SoTA models. N-Nano-V2 is short for Nemotron-Nano-V2. The distilled N-Nano-V2-9B-Base is compared against Qwen3-8B-Base and Gemma3-12B-Base, and the best score is highlighted in each row.

2.7. Base Model Evaluations

We run evaluations of all models ourselves unless otherwise stated. Our evaluation setup is built on top of `lm-evaluation-harness`⁵ for fair comparisons, with the following changes:

1. For mathematical reasoning, we evaluate GSM8K and MATH (Cobbe et al., 2021; Hendrycks et al., 2021b) benchmarks using greedy-decoding. We also highlight the competition-level slice of the MATH benchmark as “MATH Level 5”. Additionally, we report the pass@32 performance on AIME-2024. We use `Math-Verify`⁶ to grade all generations.
2. For code tasks (HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021)) we evaluate the EvalPlus variants along with the sanitization of generations (Liu et al., 2023), in a 0-shot setup. We estimate avg@32, pass@1 from 32 generations per prompt.
3. General reasoning benchmarks (OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2019), Hellaswag (Zellers et al., 2019), Winogrande (Sakaguchi et al., 2019)) are unchanged except for ARC-Challenge (Clark et al., 2018), where we present all options at the same time, similar to MMLU (Hendrycks et al., 2021a).
4. For multilingual capability, we evaluate MGSM (Shi et al., 2022) (8-shot, native CoT) and Global MMLU-Lite (Singh et al., 2024b).
5. We use RULER (Hsieh et al., 2024) as the long context benchmark. We report the average scores over all the 13 tasks included in RULER.

Accuracy results for Nemotron-Nano-12B-v2-Base with comparisons to Qwen3-8B Base and Gemma3-12B Base are shown in Tables 5 and 6. We also include the accuracy of our 9B pruned variant of Nemotron-Nano-12B-v2-Base which is discussed in Section 4.

3. Alignment

In this section we will present the alignment process we followed to convert the base checkpoint into an aligned 12B checkpoint. Our process is outlined in Figure 4.

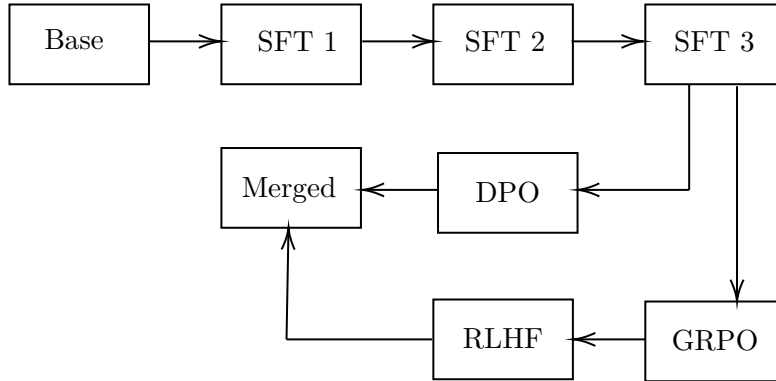


Figure 4 | Flow of alignment procedures followed to arrive at the final "Merged" Nemotron Nano 2 12B checkpoint.

⁵<https://github.com/EleutherAI/lm-evaluation-harness>.

⁶<https://github.com/huggingface/math-verify>.

Task	N-Nano-V2 12B Base	N-Nano-V2 9B Base	Qwen3 8B Base	Gemma3 12B Base
Global-MMLU-Lite				
German	74.50	68.25	75.50	69.75
Spanish	76.50	72.75	75.00	74.00
French	78.25	69.75	74.25	72.50
Italian	76.50	73.25	72.75	74.00
Japanese	71.00	67.00	70.00	71.50
Korean	72.50	67.25	67.25	70.25
Portuguese	76.25	71.25	72.50	75.75
Chinese	75.50	69.25	75.25	67.25
Average	75.13	69.94	72.81	71.88
Multilingual Math (MGSM)				
Spanish	93.20	93.60	87.60	73.60
German	88.40	88.40	78.80	66.00
French	82.40	84.40	82.00	68.00
Chinese	83.60	82.00	80.80	62.00
Japanese	76.80	68.80	71.20	56.00
Russian	91.20	90.80	85.20	72.40
Average	85.94	84.67	80.93	66.33

Table 6 | Accuracy of Nemotron-Nano-V2-Base models versus existing SoTA models on multilingual benchmarks. N-Nano-V2 is short for Nemotron-Nano-V2. The distilled N-Nano-V2-9B-Base is compared against Qwen3-8B-Base and Gemma3-12B-Base, and the best score is highlighted in each row.

3.1. Post-Training Data

Our alignment begins with a large-scale SFT stage which trains the base model on approximately 80 billion tokens of prompt-response pairs. The distribution of domains is shown in Table 7.

Math, science and coding. For Math (Toshniwal et al., 2024; Moshkov et al., 2025), Science and Coding (Ahmad et al., 2025b,a; Majumdar et al., 2024) data, we generate responses using the open-weights DeepSeek-R1-0528 model (DeepSeek-AI, 2025b) using the same prompts used for training Nemotron-H-8B and 47B Reasoning models (NVIDIA, 2025). The training data has been released as part of Nemotron-Post-Training-Dataset-v1⁷.

Tool calling. The tool-calling dataset consists of single-turn, multi-turn, and multi-step conversations. For single-turn cases, we sample prompts from xlam-function-calling-60k⁸, glaive-function-calling-v2⁹, NVIDIA-When2Call (Ross et al., 2025), and generate responses using Qwen3-235B-A22B¹⁰. Inspired by ToolACE (Liu et al., 2024) and APIGen-MT (Prabhakar et al., 2025), we extend this to multi-turn and multi-step settings by simulating conversations where

⁷<https://huggingface.co/datasets/nvidia/Nemotron-Post-Training-Dataset-v1>

⁸<https://huggingface.co/datasets/xlam-function-calling-60k>

⁹<https://huggingface.co/datasets/glaive-function-calling-v2>

¹⁰<https://huggingface.co/Qwen/Qwen3-235B-A22B>

Domain	Number of Samples
Math	1.5M
Coding	1.1M
Science	2.0M
Tool-calling	400K
Conversational	1.5M
Safety	2K
Multilingual (all domains)	5.0M

Table 7 | Post-training data distribution across domains used for our SFT stages.

Qwen3-235B-A22B plays the roles of User-Agent, Assistant-Agent, and API-Server-Agent. The User-Agent reviews available tools, poses challenging queries, interacts when addressed by the Assistant, and judges task success at the end. Each instance is paired with a random persona from **Nemotron-Personas**¹¹ to enrich diversity of queries.

The Assistant-Agent receives the initial query and available tools, executes tasks by invoking tools, interpreting their responses, and interacting with the User-Agent across single-turn, multi-turn, or multi-step scenarios. Meanwhile, the API-Server-Agent acts as a mock API server, checking parameters and returning either valid outputs or error messages depending on correctness. A lightweight rule-based tool-call verification layer further strengthens reliability by ensuring outputs are consistent and verifiable, and only successful trajectories are retained.

Multilingual data. Our multilingual synthetic post-training data are constructed by translating existing English post-training data. To address the challenges of Large Language Model (LLM) hallucinations and quality degradation on long inputs when generating synthetic translation data, we implement a robust quality assurance pipeline. Our method involves translating inputs line-by-line to manage complexity and skip non-translatable content like code. We also enforce a strict bracket format for reliable extraction and use language identification to filter out off-target translations, thereby ensuring high-quality final outputs.

Conversational data. For conversational data, we use prompts from the LMSYS dataset (Zheng et al., 2023) and generate responses using the **Qwen3-235B-A22B** reasoning model (Yang et al., 2025). We also incorporate prompts from HelpSteer2 and HelpSteer3, paired with responses generated by the same model. In addition, we draw on a subset of approximately 550k prompts from WildChat-1M (Li et al., 2024b), again generating reasoning responses with **Qwen3-235B-A22B**. We also include multi-turn conversations with Deepseek R1 responses using the multi-turn conversational prompts used in NVIDIA (2025).

Safety. We leveraged a mix of harmful and benign prompts drawn from the Nemotron Content Safety Dataset V2 (Ghosh et al., 2025)¹², HarmfulTasks (Hasan et al., 2024), RedTeam2K (Luo et al., 2024), and gretel-v1 (gre, 2024). Responses were generated using DeepSeek-R1-0528¹³. To ensure safety, we applied a two-step approach: initial prompting followed by filtering with guard models to verify that outputs remained safe.

¹¹<https://huggingface.co/datasets/NVIDIA/Nemotron-Personas>

¹²<https://huggingface.co/datasets/nvidia/Aegis-AI-Content-Safety-Dataset-2.0>

¹³<https://huggingface.co/deepseek-ai/DeepSeek-R1>

3.2. Post Training

Stage 1 SFT. As Figure 4 illustrates, we employ three distinct stages of supervised fine-tuning. Stage 1 uses the full dataset described in Section 3.1, augmented with a subsample of roughly 10% of prompts paired with outputs stripped of reasoning traces. This exposes the model to “empty” traces, enabling it to produce direct answers in a reasoning-off mode. To improve efficiency and preserve long-context ability from pretraining, we concatenate samples into sequences of approximately 128k tokens, reducing padding overhead and encouraging long-range learning.

Stage 2 SFT. Stage 2 targets tool-calling. Although Stage 1 improved performance on most benchmarks, tool-calling accuracy degraded. We attribute this to sample concatenation at 128k, which likely disrupted learning of tool-calling patterns. Thus, Stage 2 was trained without concatenation, using the full tool-calling dataset and a representative subsample of other domains.

Stage 3 SFT. Stage 3 reinforces long-context capability. It incorporates long-context data following the recipe used in Nemotron-H preparation (NVIDIA, 2025), along with augmented examples across domains where reasoning traces were abruptly truncated to 1–2k tokens while preserving the final answer. This truncation strategy improved robustness under varying inference-time thinking budgets.

IFEval RL. To improve instruction adherence, we sampled 16,000 prompts from the LMSYS Chat dataset and augmented them with IFEval-style instructions. A rule-based verifier scored outputs based on how well they satisfied each instruction, creating a reward signal that prioritized following directions with precision. IFEval RL experiments provided significant boost to IFEval capabilities while the rest of the benchmarks fluctuated slightly requiring careful checkpoint selection.

DPO. In another branch of training, we apply the DPO algorithm to improve tool-calling. We evaluate performance using the BFCL v3 benchmark, which extends BFCL v2 with greater emphasis on multi-step (multiple tool calls to achieve a goal) and multi-turn (multiple user-agent interactions). To strengthen these capabilities in the Nano V2 aligned model, we use the WorkBench environment, a multi-step verifiable tool-calling setup adapted from Styles (Styles et al., 2024). In each WorkBench task, the model must issue a sequence of tool calls across multiple steps, with correctness verified through database state comparisons.

Nano V2 undergoes reinforcement learning in this environment through iterative stages of Direct Preference Optimization. For each candidate checkpoint from the long-context stage, we generate on-policy data consisting of positive examples (successful tool calls) and negative examples (failed generations) for every WorkBench prompt. This process ensures that iterative DPO remains on-policy.

RLHF. We evaluate the model’s overall helpfulness and chat capabilities using the Arena-Hard benchmark. To improve performance on this benchmark, we use GRPO to train candidate checkpoints from the SFT stage using English-only contexts from HelpSteer3 (Wang et al., 2025). During training, we generate responses both with and without thinking traces and use a Qwen-based reward model to judge the rollouts.

Model Merging. During training, we observed a trade-off between reasoning capabilities and chat capabilities. To address this, we opted for checkpoint interpolation Wortsman et al. (2022),

Evaluation	Nemotron-Nano-v2-12B	Qwen3-8B	Qwen3-14B
AIME-2024	85.42	75.83	81.53
AIME-2025	76.25	69.31	66.6
MATH-500	97.75	96.3	96.85
GPQA-DIAMOND	64.48	59.61	64.53
LIVECODEBENCH (07/24-12/24)	70.79	59.5	63.08
SCICODE SUB-TASK	18.75	24.65	26.04
HUMANITY'S LAST EXAM	6.30	4.40	5.38
IFEVAL (INST. STRICT)	89.81	89.39	91.32
BFCL v3	66.98	66.34	68.01
RULER @ 128k	83.36	74.13	73.55
ARENAHARD	74	78.4	87.7

Table 8 | Evaluation results with reasoning "ON" (for **Nemotron-Nano-v2-12B**, **Qwen3-8B**, and **Qwen3-14B** across reasoning and general capability benchmarks.

blending in an RL checkpoint with strong reasoning capabilities with an RL checkpoint with strong chat capabilities. Checkpoint interpolation is performed by linearly interpolating model weights: $(1 - \alpha) \cdot w_{model1} + \alpha \cdot w_{model2}$. We experimented with a parameter sweep over α values from 0.1 to 0.9 in increments of 0.1, and found that values around 0.5 offered a good trade-off.

3.3. Evaluation

Our 12B model’s performance is summarized in Table 8. To test reasoning capabilities across domains, we evaluate the models on MATH-500 (Lightman et al., 2023), AIME-2024, AIME-2025, GPQA-DIAMOND (Rein et al., 2023), LIVECODEBENCH (07/24 - 12/24) (Jain et al., 2024), SCICODE (Tian et al., 2024), and HUMANITY’S LAST EXAM (Phan et al., 2025). For broader evaluation on diverse capabilities, we use IFEVAL (Zhou et al., 2023) for instruction following capabilities, BFCL v3 (Yan et al., 2024) for tool-calling, RULER for long-context, and ARENAHARD (Li et al., 2024a) for chat capability.

We conduct evaluations using NeMo-Skills¹⁴. We report PASS@1 average of 16 runs for AIME-2024, AIME-2025; average of 4 runs for MATH-500, GPQA-DIAMOND, LIVECODEBENCH, IFEVAL; and score of 1 run for BFCL v3, SCICODE, HUMANITY’S LAST EXAM, RULER, and ARENAHARD.

3.4. Budget Control Evaluation

Nemotron Nano V2 allows users to specify how many thinking tokens the model may generate before producing the final answer. The final answer is the portion of text typically shown to end users. This feature is implemented by counting tokens after the model begins generating the `<think>` token. Once the budget is reached, the inference setup attempts to insert a closing `</think>` tag. Rather than inserting it immediately, we let the model finish its current sentence and place the tag at the next newline. In extreme cases where no newline appears, the system enforces closure within 500 tokens past the budget: if no newline occurs by the $(\text{budget} + 500)^{\text{th}}$ token, the `</think>` tag is forcibly inserted. Figure 5b shows our models budget control behavior. Apart from just presenting the accuracy of the model at various budgets, we also inspect if the model generations are well-formatted at various budgets. We inspect for two kinds of failure modes:

¹⁴ <https://github.com/NVIDIA/NeMo-Skills>

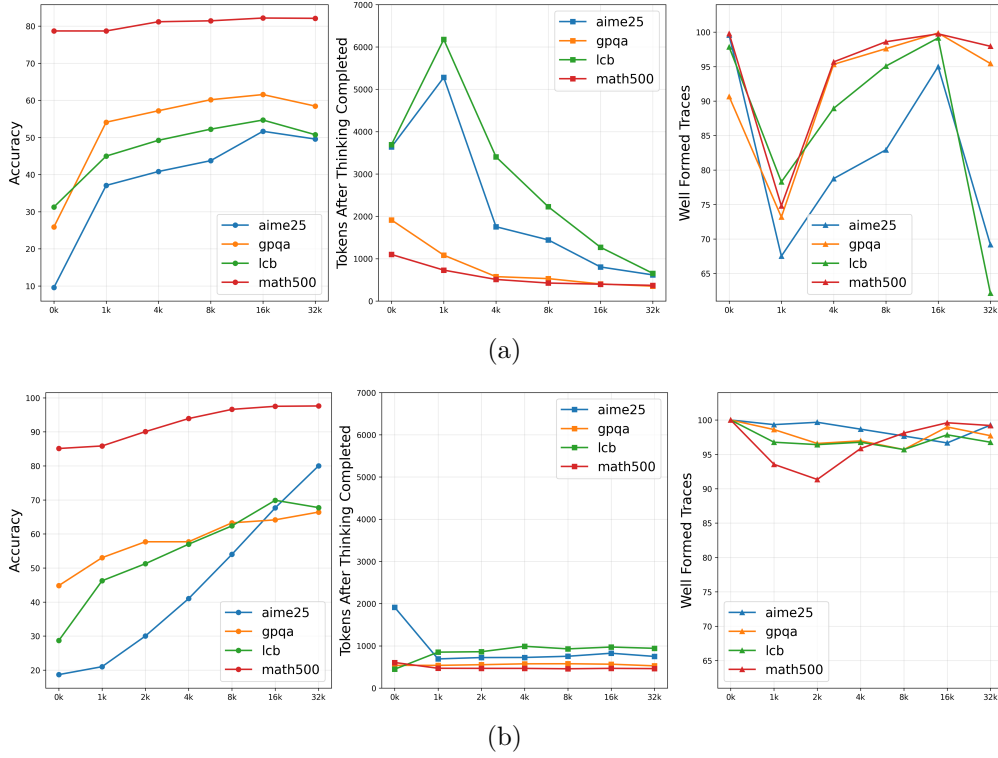


Figure 5 | Comparison of budget control before truncation training (a) and after truncation training was included (b). For all plots above the x-axis indicates the budget assigned for thinking tokens.

- In one failure mode, the model uses more tokens in the final answer to “compensate” for restrictions in the thinking traces. Without truncated training examples in the SFT stage, this compensation effect is prevalent (Figure 5a, center). With truncated training, however, the effect is absent (Figure 5b, center).
- Another issue is that the model can remain in “thinking mode” even after the closing tag `</think>` is inserted. This is evident when the model generates the closing tag again after the forced insertion, suggesting it does not fully “register” the artificial closure. We evaluate this using “Well-Formedness,” where a well-formed response should contain only a single closing tag (either forced by the budget or produced naturally). Figure 5a (right) shows that for short budgets, the percentage of well-formed responses drops sharply. With truncation training, however, the model consistently produces well-formed responses (Figure 5b, right).

4. Pruning and Distillation

In this section, we describe the pruning and distillation process to compress the aligned 12B model to the Nano 2 model with the goal of running longer context (128k sequence length) inference on the NVIDIA A10G GPU. Note that storing just the weights of a 12B parameter model in `bf16` precision requires 22.9 GiB, which is more than the 22 GiB memory capacity of an A10G GPU; this clearly indicates the need for compression.

Our compression strategy builds on Minitron (Muralidharan et al., 2024; Sreenivas et al., 2024; Taghibakhshi et al., 2025), which is a lightweight model pruning framework for LLMs. While Minitron was originally designed for compressing pretrained base models targeting user-defined parameter budgets, in this work, we extend it to compress reasoning models while also incorporating

the memory constraints and throughput-based objectives stated above.

4.1. Importance Estimation

We collect importance or sensitivity scores for each model component (e.g., layers, FFN neurons) to help decide which components to remove; this is the *importance estimation* phase. The scores computed in this phase are used to decide which model components can be pruned. We note that sensitivity analysis based on gradient information is typically impractical at modern LLM scale (Muralidharan et al., 2024); instead, we rely on a lightweight strategy that uses only forward passes. In this work, we use a simplified approach that works well in our ablation studies: a) prune layers, and b) prune FFN hidden dimensions (effectively neurons) and embedding channels. We also experimented with pruning Mamba heads; unfortunately, this axis caused severe accuracy degradation. We now describe how we compute the importance of each layer, embedding channel, FFN neuron and Mamba head.

Layer importance. We compute layer importance in an iterative fashion: for each candidate layer, we temporarily remove it from the model and compute the mean squared error (MSE) between the original model’s logits and those produced by the pruned model. This MSE reflects the contribution of that layer to the model’s predictions: lower values indicate smaller impact. At each pruning step, we remove the layer with the lowest MSE, as it has the least influence on the final output. We repeat this process until the desired depth is reached. This strategy ensures that pruning preferentially removes layers whose absence minimally affects the model’s behavior. For more details on iterative MSE-based layer importance, please refer to NVIDIA (2025).

FFN and embedding channel importance. FFN layers internally are composed of two linear operators with a non-linear activation in between:

$$\text{FFN}(\mathbf{X}) = \delta\left(\mathbf{X} \cdot \mathbf{W}_1^T\right) \cdot \mathbf{W}_2.$$

Here, $\mathbf{X}$ denotes the input, and $\mathbf{W}_1$ and $\mathbf{W}_2$ are the two associated weight matrices in the FFN layer. $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d_{ffn} \times d_{model}}$, where d_{model} and d_{ffn} are the model hidden dimension and FFN hidden dimension respectively. $\delta(\cdot)$ refers to the non-linear activation function (squared ReLU in this work).

Following the same procedure as Minitron (Muralidharan et al., 2024), we compute the importance of each neuron in the first linear operator of each FFN layer by examining the set of outputs it produces. We use a small calibration dataset of 1024 samples for this purpose. Formally, we compute each neuron’s importance score by aggregating its outputs given an input batch X :

$$F_{\text{neuron}}^{(i)} = \sum_{\mathbf{B}, \mathbf{S}} \delta\left(\mathbf{X}(\mathbf{W}_1^i)^T\right).$$

Here, $\mathbf{W}_1^i$ refers to the i^{th} row of the weight matrix $\mathbf{W}_1$. $\sum_{\mathbf{B}, \mathbf{S}}$ refers to aggregation along the batch and sequence dimensions. We use the **mean** and **l2-norm** aggregation functions along the batch and sequence dimensions, following the observations in the Minitron paper. For a sequence of scores $\mathbf{S}$, **mean** aggregation is defined as $\frac{1}{n} \sum_{i=1}^n |\mathbf{S}_i|$, and **l2-norm** is $\sqrt{\sum_{i=1}^n \mathbf{S}_i^2}$. Embedding channel importance is computed similarly, by examining the outputs of LayerNorm layers instead; we refer the reader to Muralidharan et al. (2024) for more details.

Mamba importance. Mamba layers process inputs through multiple projection matrices ($W_x, W_z, W_B, W_C, W_{dt}$) that produce intermediate representations before causal convolution and selective state space model (SSM) updates, followed by gated normalization and an output projection (W_O). We follow the methodology described in Taghibakhshi et al. (2025) for importance estimation: specifically, we adopt a nested activation-based scoring strategy over a small calibration dataset of 1024 samples, similar to FFN importance but adapted to Mamba’s group-aware structure. First, we obtain activation scores from the W_x projection, denoted $s \in \mathbb{R}^{m_h \times m_d}$, where m_h is the number of Mamba heads and m_d is the Mamba head channel dimension. For each channel d , the score is computed as

$$s_d = \left\| \sum_{\mathbf{B}, \mathbf{S}} s_{:,d} \right\|_2,$$

where the aggregation is over the batch ($\mathbf{B}$) and sequence ($\mathbf{S}$) dimensions, using both **mean** and **l2-norm** metrics. Next, head scores are computed by using the **l2-norm** over the Mamba head channel set:

$$f_h = \|s_{h,m_d}\|_2, \quad \forall h \in \{1, \dots, m_h\},$$

and heads are ranked within each Mamba group $\mathcal{G}_g$ to preserve group-aware computation semantics:

$$\mathcal{R}_g = \text{argsort}_{h \in \mathcal{G}_g}(f_h).$$

which ensures that pruning decisions respect the model’s structural constraints and SSM’s sequence modeling. The lowest-scoring heads are pruned by trimming the corresponding rows from all affected projection, convolution, and SSM parameter matrices. This strategy preserves the integrity of the SSM block while removing less important Mamba heads. As shown in Taghibakhshi et al. (2025), pruning Mamba heads yields a better accuracy-throughput trade-off than pruning head channels; we consequently focus on head pruning in this work.

4.2. Lightweight Neural Architecture Search

We first define the constraints and objectives for the Nano 2 model, and then describe our lightweight Neural Architecture Search (NAS) framework that finds the most promising architectural candidates that meet our objectives and constraints.

Memory constraints. Memory requirements during inference consist of two distinct components with different scaling behaviors. The parameter memory, while substantial, remains constant regardless of the input size. In contrast, the key-value cache memory scales linearly with both batch size and sequence length, often becoming the dominant factor in long-sequence scenarios. For the Nano 2 model, our goal was to be able to perform inference at a sequence length of 128k and a batch size of at least 1 within a memory budget of 19.66 GiB. We obtained the budget as follows: from the 22.06 GiB available memory on an NVIDIA A10G GPU, we subtract a 5% buffer for frameworks such as vLLM and TensorRT-LLM and another 1.3 GiB to allow sufficient space for a vision encoder.

Measuring throughput. For the experiments below, unless otherwise specified, we measure throughput on an input and output sequence length of 8k and 16k tokens respectively, which we believe represents a typical reasoning scenario. For this combination of input and output sequence length, we report vLLM output token generation throughput at the maximum batch size that fits on the A10G GPU.

4.2.1. Candidate enumeration.

Our compression strategy explores multiple axes within the 19.66 GiB memory budget through combinatorial pruning. Our search space includes depth reduction (removing 6-10 layers from the original 62-layer architecture) combined with width pruning of embedding channels (4480-5120), FFN dimension (13440-20480), and Mamba heads (112-128). This multi-axis search space results in hundreds of candidate architectures meeting the memory constraint.

4.2.2. Finding the Best Architecture

Since performing knowledge distillation and throughput benchmarking on the full set of candidates would be prohibitively expensive, we break down the problem into two parts: (1) find the optimal depth for the compressed model, and (2) find the optimal width-pruned architecture given the depth.

Effect of depth. We compare the accuracy of three depth-pruned candidates obtained from the 12B model with 52, 54 and 56 layers. Here, we keep the number of attention layers fixed at 4 for all three variants so as to achieve a good balance between KV cache size and long-context performance; prior work has indicated that an attention-to-total-layers ratio between 7-8% is reasonable (NVIDIA, 2025). We leave the width dimensions untouched for this experiment. Table 9 lists average reasoning accuracy at different depths after 6B tokens of distillation; in line with our previous observations on the strong correlation between depth and task performance (Muralidharan et al., 2024; Sreenivas et al., 2024), we notice that reducing depth beyond 56 layers results in significant accuracy degradation; as a result, we fix the depth at 56 for further width pruning.

	Accuracy (Avg)
52 Layers	44.92
54 Layers	47.35
56 Layers	51.48

Table 9 | Effect of depth on reasoning accuracy. Results are after distilling with 6B tokens.

Combining depth and width pruning. As described above, we fix the depth of our target model to 56 layers with 4 attention layers. We perform 60B tokens of distillation on this checkpoint (see Section 4.3 for additional details) and perform further width pruning along the embedding, FFN, and Mamba axes. We enumerate all candidate pruned architectures that meet our memory budget, and sort them in decreasing order of estimated memory consumption at 128k context length and batch size 1. The top 3 candidates from this list are picked for further evaluation: in particular, we perform short Knowledge Distillation (KD) on these candidates for 19B tokens after depth+width pruning; we also benchmark throughput to pick the final architectural candidate. Table 10 lists the architectural details of the top 3 candidates, along with the achieved task performance (post KD) and throughput. As shown in the Table, Candidate 2 achieves the best accuracy while still having reasonable runtime performance; consequently, we use this architecture for Nano 2.

FFN vs. Mamba pruning. We ablate the number of Mamba heads following the recipe in Taghibakhshi et al. (2025), considering configurations with 87.5% and 93.75% of the original heads. However, due to the relatively smaller compression ratios explored in this work (less than 15% after depth pruning) compared to those in Taghibakhshi et al. (2025) (around 50%), we find that applying Mamba head pruning yields limited benefit, and in these cases, pruning only the FFN and

	#Layers	Hidden	FFN	Mamba	#Heads	Params. (B)	Accuracy	Throughput
Candidate 1	56	4480	17920		112	8.92	59.07	161.02
Candidate 2	56	4480	15680		128	8.89	63.02	156.42
Candidate 3	56	4800	14400		120	8.97	62.94	155.86

Table 10 | Top 3 candidates for architecture selection. Accuracy is the average across reasoning benchmarks after distillation with 19B tokens. The last column shows vLLM output generation throughput (ISL/OSL=8k/16k and batch size=8).

embedding dimensions—after depth pruning—proves sufficient to achieve the desired compression while preserving accuracy. Candidates 1 and 2 in Table 10 highlight this difference.

4.3. Retraining with Distillation

To recover the accuracy lost due to pruning, the model undergoes continued training. Recent work has demonstrated that distilling knowledge from the original model to the pruned model outperforms conventional fine-tuning (Muralidharan et al., 2024; Sreenivas et al., 2024; Bercovich et al., 2024); we thus adopt logit-based distillation for continued training, employing forward KL divergence loss exclusively during the accuracy recovery phase (see §3 of the Minitron paper (Muralidharan et al., 2024) for more details on the distillation loss formulation). Building on the candidate selection process described in §4.2, we continue training Candidate 2 in an extended phase, as detailed below, to yield the final Nano 2 reasoning and base models.

% Reasoning-SFT data	% Pretraining data	Accuracy (Avg)
50	50	57.5
70	30	58.5
90	10	57.2

Table 11 | Effect of varying reasoning data proportion on math accuracy after ~ 6 B tokens of KD.

Reasoning model. The reasoning model is distilled in stages with increasing sequence lengths to strengthen extended reasoning and long-context capabilities; this is followed by targeted reinforcement learning (RL), preference optimization and model merging to retain desired behaviors and ensure robustness across diverse tasks. We now describe these various stages:

1. Depth pruning to 56 layers; Knowledge Distillation (KD) with ~ 60 B tokens at 8,192 sequence length.
2. Width pruning and KD with:
 - ~ 50 B tokens at 8,192 sequence length.
 - ~ 25 B tokens at 49,152 sequence length.
 - ~ 1 B tokens at 262,144 sequence length.
3. Direct Preference Optimization (DPO).
4. Group Relative Policy Optimization (GRPO).
5. KD with ~ 0.4 B tokens at 262,144 sequence length to recover post-RL drops.
6. RLHF for alignment with human preferences.
7. Model merging between steps 5 and 6 via 0.5 linear interpolation.

More details on DPO, GRPO and RLHF can be found in Section 3. Figure 6 shows the effects of staged training on model accuracy across different reasoning benchmarks. Here, the x -axis represents the various stages (starting from Step 2 above), and the y -axis shows the scores obtained for the various benchmarks as training progresses. As shown in the Figure, DPO and GRPO are critical for enhancing function-calling (BFCL v3) and instruction-following (IFEval) capabilities, though the latter temporarily degrades multi-task understanding (MMLU-Pro), which is recovered in the next step (post-GRPO KD). Finally, RLHF enhances alignment with human preferences (Arena-Hard) but causes additional benchmark drops, which are then recovered through model merging.

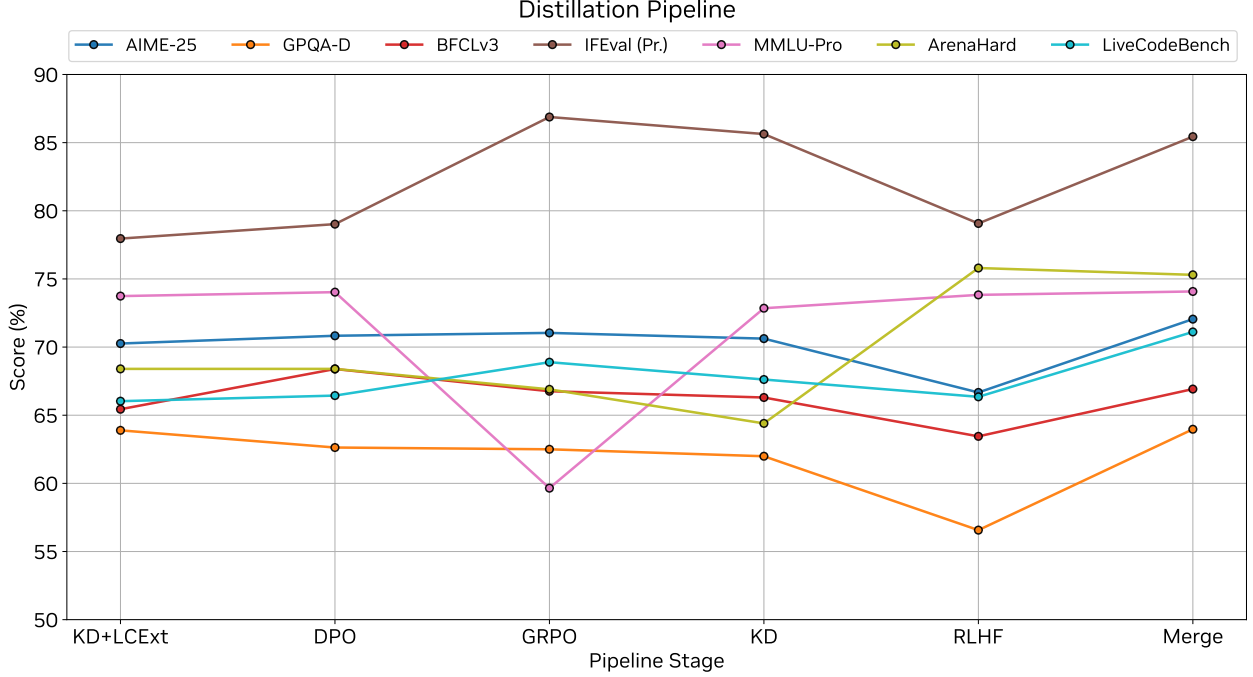


Figure 6 | Task accuracy at different stages of the distillation pipeline for Nemotron Nano 2.

Dataset: We observe that a mix of 70% post-training stage 2 data (Section 3.2) and 30% pretraining (Section 2.2) data yields the highest accuracy (Table 11). For KD at sequence length 262,144, we use 100% stage 3 post-training data (Section 3.2).

Base model. Distillation proceeds in stages: depth-only pruning and KD on ~ 120 B tokens, followed by width pruning and KD on ~ 360 B tokens (both at sequence length 8,192), and finally KD on ~ 2.5 B tokens at sequence length 524,288 to instill long-context capabilities.

Dataset: Following Sreenivas et al. (2024), we use 100% pretraining data described in sections 2.2 and 2.6 for distillation of the base model at sequence lengths 8,192 and 524,288, respectively.

4.4. Results

We efficiently compress the 12B model to 9B parameters by pruning full layers (depth), FFN hidden size, and embedding channels, improving inference throughput and enabling long-context inference on an NVIDIA A10G GPU. Nemotron-Nano-9B-v2 retains 56 layers of the original model. Additionally, the number of embedding channels were pruned from 5120 to 4480, and FFN intermediate size was pruned from 20480 to 15680. As shown in Figure 1 and Tables 5 and 6, Nemotron-Nano-9B-v2 achieves $3\times$ - $6\times$ higher throughput than Qwen3-8B for generation-heavy scenarios, while surpassing

it in accuracy and remaining comparable to the 12B teacher on most benchmarks.

5. Conclusion

In this report, we introduced Nemotron-Nano-9B-v2, a hybrid Mamba-Transformer reasoning model that achieves comparable or better accuracies at up to $6\times$ higher throughput than existing state-of-the-art models such as Qwen3-8B. To create Nemotron-Nano-9B-v2, we started by pre-training Nemotron-Nano-12B-v2-Base on 20T tokens, using a carefully constructed mix of curated and synthetically generated data. We aligned Nemotron-Nano-12B-v2-Base using several stages of SFT, GRPO, DPO, and RLHF before using the Minitron compression via pruning and distillation strategy to produce the final model. As a result of this compression, Nemotron-Nano-9B-v2 can run inference on context lengths of up to 128k tokens in `bf16` precision on a single NVIDIA A10G GPU with 22 GiB of memory. We have open-sourced Nemotron-Nano-9B-v2 along with its corresponding sibling Nemotron-Nano-9B-v2-Base and parent Nemotron-Nano-12B-v2-Base models, plus the majority of its pre- and post-training data on HuggingFace (links at the bottom of Section 1).

Contributors

We thank the following people for their invaluable contributions to NVIDIA Nemotron Nano 2.

Data. Abhinav Khattar, Aleksander Ficek, Arham Mehta, Ayush Dattagupta, Brandon Norick, Dan Su, Daria Gitman, Evelina Bakhturina, Igor Gitman, Ivan Moshkov, Jaehun Jung, Jane Polak Scowcroft, Jocelyn Huang, Joseph Jennings, Jupinder Parmar, Markus Kliegl, Matvei Novikov, Mehrzad Samadi, Miguel Martinez, Mohammad Shoeybi, Mostofa Patwary, Pavlo Molchanov, Pritam Gundecha, Rabeeh Karimi Mahabadi, Ranjit Rajan, Rima Shahbazyan, Sanjeev Satheesh, Sarah Yurick, Sean Narenthiran, Seungju Han, Shizhe Diao, Shrimai Prabhume, Shubham Toshniwal, Siddhartha Jain, Somshubra Majumdar, Syeda Nahida Akter, Vahid Noroozi, Vineeth Kalluru, Vitaly Kurin, Wasi Uddin Ahmad, Wei Du, Ximing Lu, Yejin Choi, Ying Lin.

FP8. Hua Huang, Jinze Xue, Keith Wyss, Kunlun Li, Mike Chrzanowski, Oleg Rybakov, Przemek Tredak, Tim Moon, Zhongbo Zhu.

Architecture. Bitu Darvish Rouhani, Brandon Norick, Duncan Riach, Nidhi Bhatia, Roger Waleffe, Wonmin Byeon, Ritika Borkar, Xin Dong, Yonggan Fu.

Pretraining. Aarti Basant, Abhijit Paithankar, Abhinav Khattar, Deepak Narayanan, Herman Sahota, Hexin Wang, Jupinder Parmar, Mohammad Shoeybi, Mostofa Patwary, Namit Dhameja, Roger Waleffe, Russell J. Hewett, Ryan Prenger, Seonmyeong Bak.

Infrastructure. Alex Kondratenko, Alex Shaposhnikov, Anubhav Mandarwal, Ashwin Poojary, Dong Ahn, Gargi Prasad, Haim Elisha, Harsh Sharma, Kumar Anik, Maer Rodrigues de Melo, Ruoxi Zhang, Shelby Thomas, Stefania Alborghetti, Tony Wang.

Long Context. Deepak Narayanan, Dima Rekish, Duncan Riach, John Kamalu, Kezhi Kong, Markus Kliegl, Roger Waleffe, Samuel Krizan.

Inference. Daniel Afrimi, Helen Ngo, Keshav Santhanam, Kushan Ahmadian, Lawrence McAfee, Luis Vega, Nave Assaf, Peter Dykas, Shanmugam Ramasamy, Siddharth Singh, Tomer Asida, Vijay Korthikanti.

Alignment. Adithya Renduchintala, Alexander Bukharin, Ameya Sunil Mahabaleshwarkar, Banghua Zhu, Bilal Kartal, Brian Yu, Charles Wang, Christian Munley, David Mosallanezhad, Gerald Shen, Haifeng Qian, Hayley Ross, Hoo Chang Shin, Igor Gitman, Jian Zhang, Jiaqi Zeng, Julien Veron

Vialard, Junkeun Yi, Kezhi Kong, Luis Vega, Makesh Narsimhan Sreedhar, Oleksii Hrinchuk, Oleksii Kuchaiev, Peter Jin, Prasoon Varshney, Ritu Gala, Shuoyang Ding, Soumye Singhal, Tugrul Konuk, Venkat Srinivasan, Vitaly Lavrukhin, Yian Zhang, Yoshi Suhara, Zhen Dong, Zijia Chen.

Compression. Aditya Malte, Akhiad Bercovich, Akshay Hazare, Ali Taghibakhshi, Ameya Sunil Mahabaleshwarkar, Ashwath Aithal, Banghua Zhu, Daniel Korzekwa, Deepak Narayanan, Gerald Shen, Hayley Ross, Julien Veron Vialard, Luis Vega, Marcin Chochowski, Mostofa Patwary, Nima Tajbakhsh, Oluwatobi Olabiyi, Pavlo Molchanov, Ran El-Yaniv, Roger Waleffe, Saurav Muralidharan, Sepehr Sameni, Sharath Turuvekere Sreenivas, Tomer Asida, Yashaswi Karnati, Yian Zhang, Yoshi Suhara, Zijia Chen.

Software Support. Abhijit Khairnar, Adithya Renduchintala, Ali Taghibakhshi, Anna Shors, Ashwath Aithal, Balaram Buddharaju, Bobby Chen, Charlie Truong, Deepak Narayanan, Dmytro Pykhtar, Duncan Riach, Gerald Shen, Helen Ngo, Jared Casper, Jimmy Zhang, Keshav Santhanam, Kezhi Kong, Lawrence McAfee, Luis Vega, Nima Tajbakhsh, Parth Chadha, Piotr Bialecki, Prashant Gaikwad, Rajen Patel, Roger Waleffe, Sahil Jain, Terry Kong, Tyler Poon, Vijay Korthikanti, Vikram Fugro, Yoshi Suhara, Zhiyu Li.

Evaluations and Safety. Christopher Parisien, Dan Su, Daniel Rohrer, Eileen Long, Erick Galinkin, Helen Ngo, Katherine Luna, Keshav Santhanam, Kezhi Kong, Leon Derczynski, Marta Stepniewska-Dziubinska, Meriem Boubdir, Michal Bien, Michael Boone, Michael Evans, Michal Bien, Michal Zawalski, Pablo Ribalta, Piotr Januszewski, Pradeep Thalasta, Sanjeev Satheesh, Shaona Ghosh, Tomasz Hliwiak.

Legal and Compliance. Barnaby Simkin, Chetan Mungekar, Dina Yared, Iain Cunningham, Katherine Cheung, Laya Sleiman, Meredith Price, Michael Boone, Nikki Pope, Ria Cheruvu, Saori Kaji.

Marketing. Amelia Barton, Chris Alexiuk, Mark Cai, Nirmal Kumar Juluru, Shreya Gopal.

Project Management. Alejandra Rico, Amy Shen, Ann Guan, Ashton Sharabiani, Elliott Ning, Krzysztof Pawelec, Negar Habibi, Twinkle Vashishth.

Product. Arun Venkatesan, Chintan Patel, Chris Alexiuk, Joey Conway, Padmavathy Subramanian, Udi Karpas.

Leadership. Andrew Tao, Boris Ginsburg, Bryan Catanzaro, Eric Chung, Jan Kautz, Joey Conway, Jonathan Cohen, Kari Briski, Mohammad Shoeybi, Mostofa Patwary, Oleksii Kuchaiev, Pavlo Molchanov.

We also thank Chen Zhang, Michael Goin, Thomas Parnell from the vLLM team for their assistance.

References

- Gretel synthetic safety alignment dataset, 12 2024. URL <https://huggingface.co/datasets/gretelai/gretel-safety-alignment-en-v1>.
- Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.
- Wasi Uddin Ahmad, Aleksander Ficek, Mehrzad Samadi, Jocelyn Huang, Vahid Noroozi, Somshubra Majumdar, and Boris Ginsburg. Opencodeinstruct: A large-scale instruction tuning dataset for code llms. *arXiv preprint arXiv:2504.04030*, 2025a.
- Wasi Uddin Ahmad, Sean Narenthiran, Somshubra Majumdar, Aleksander Ficek, Siddhartha Jain, Jocelyn Huang, Vahid Noroozi, and Boris Ginsburg. Opencodereasoning: Advancing data distillation for competitive coding. *arXiv preprint arXiv:2504.01943*, 2025b.
- Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints, 2023. URL <https://arxiv.org/abs/2305.13245>.
- Syeda Nahida Akter, Shrimai Prabhumoye, John Kamalu, Sanjeev Satheesh, Eric Nyberg, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. MIND: Math Informed syNthetic Dialogues for Pretraining LLMs, 2024. URL <https://arxiv.org/abs/2410.12881>.
- Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmoLLM2: When Smol Goes Big – Data-Centric Training of a Small Language Model, 2025. URL <https://arxiv.org/abs/2502.02737>.
- Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction, 2024. URL <https://arxiv.org/abs/2309.14316>.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program Synthesis with Large Language Models, 2021. URL <https://arxiv.org/abs/2108.07732>.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL <https://arxiv.org/abs/2502.13923>.
- Adrien Barbaresi. Trafilatura: A Web Scraping Library and Command-Line Tool for Text Discovery and Extraction. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pp. 122–131. Association for Computational Linguistics, 2021. URL <https://aclanthology.org/2021.acl-demo.15>.
- Akhiad Bercovich, Tomer Ronen, Talor Abramovich, Nir Ailon, Nave Assaf, Mohammad Dabbah, Ido Galil, Amnon Geifman, Yonatan Geifman, Izhak Golan, Netanel Haber, Ehud Karpas, Roi Koren, Itay Levy, Pavlo Molchanov, Shahar Mor, Zach Moshe, Najeeb Nabwani, Omri Puny,

- Ran Rubin, Itamar Schen, Ido Shahaf, Oren Tropp, Omer Ullman Argov, Ran Zilberstein, and Ran El-Yaniv. Puzzle: Distillation-Based NAS for Inference-Optimized LLMs, 2024. URL <https://arxiv.org/abs/2411.19146>.
- Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, Ido Shahaf, Oren Tropp, Ehud Karpas, Ran Zilberstein, Jiaqi Zeng, Soumye Singhal, Alexander Bukharin, Yian Zhang, Tugrul Konuk, Gerald Shen, Ameya Sunil Mahabaleshwarkar, Bilal Kartal, Yoshi Suhara, Olivier Delalleau, Zijia Chen, Zhilin Wang, David Mosallanezhad, Adi Renduchintala, Haifeng Qian, Dima Rekesh, Fei Jia, Somshubra Majumdar, Vahid Noroozi, Wasi Uddin Ahmad, Sean Narenthiran, Aleksander Ficek, Mehrzad Samadi, Jocelyn Huang, Siddhartha Jain, Igor Gitman, Ivan Moshkov, Wei Du, Shubham Toshniwal, George Armstrong, Branislav Kisacanin, Matvei Novikov, Daria Gitman, Evelina Bakhturina, Jane Polak Scowcroft, John Kamalu, Dan Su, Kezhi Kong, Markus Kliegl, Rabeeh Karimi, Ying Lin, Sanjeev Satheesh, Jupinder Parmar, Pritam Gundecha, Brandon Norick, Joseph Jennings, Shrimai Prabhumoye, Syeda Nahida Akter, Mostofa Patwary, Abhinav Khattar, Deepak Narayanan, Roger Waleffe, Jimmy Zhang, Bor-Yiing Su, Guyue Huang, Terry Kong, Parth Chadha, Sahil Jain, Christine Harvey, Elad Segal, Jining Huang, Sergey Kashirsky, Robert McQueen, Izzy Putterman, George Lam, Arun Venkatesan, Sherry Wu, Vinh Nguyen, Manoj Kilaru, Andrew Wang, Anna Warno, Abhilash Somasamudramath, Sandip Bhaskar, Maka Dong, Nave Assaf, Shahar Mor, Omer Ullman Argov, Scot Junkin, Oleksandr Romanenko, Pedro Larroy, Monika Katariya, Marco Rovinelli, Viji Balas, Nicholas Edelman, Anahita Bhiwandiwalla, Muthu Subramaniam, Smita Ithape, Karthik Ramamoorthy, Yuting Wu, Suguna Varshini Velury, Omri Almog, Joyjit Daw, Denys Fridman, Erick Galinkin, Michael Evans, Katherine Luna, Leon Derczynski, Nikki Pope, Eileen Long, Seth Schneider, Guillermo Siman, Tomasz Grzegorzec, Pablo Ribalta, Monika Katariya, Joey Conway, Trisha Saar, Ann Guan, Krzysztof Pawelec, Shyamala Prayaga, Oleksii Kuchaiev, Boris Ginsburg, Oluwatobi Olabiyi, Kari Briski, Jonathan Cohen, Bryan Catanzaro, Jonah Alben, Yonatan Geifman, Eric Chung, and Chris Alexiuk. Llama-nemotron: Efficient reasoning models, 2025a. URL <https://arxiv.org/abs/2505.00949>.
- Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, et al. Llama-nemotron: Efficient reasoning models. *arXiv preprint arXiv:2505.00949*, 2025b.
- Janek Bevendorff, Benno Stein, Matthias Hagen, and Martin Potthast. Elastic ChatNoir: Search Engine for the ClueWeb and the Common Crawl. In Leif Azzopardi, Allan Hanbury, Gabriella Pasi, and Benjamin Piwowarski (eds.), *Advances in Information Retrieval. 40th European Conference on IR Research (ECIR 2018)*, Lecture Notes in Computer Science, Berlin Heidelberg New York, March 2018. Springer.
- Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about Physical Commonsense in Natural Language, 2019. URL <https://arxiv.org/abs/1911.11641>.
- Andrei Z Broder. Identifying and filtering near-duplicate documents. In *Annual symposium on combinatorial pattern matching*, pp. 1–10. Springer, 2000.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, et al. Evaluating Large Language Models Trained on Code, 2021. URL <https://arxiv.org/abs/2107.03374>.
- Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, NIPS ’17, 2017.

- 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*, abs/1803.05457, 2018.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, 2021. URL <https://arxiv.org/abs/2110.14168>.
- Tri Dao and Albert Gu. Transformers are SSMS: Generalized Models and Efficient Algorithms Through Structured State Space Duality, 2024. URL <https://arxiv.org/abs/2405.21060>.
- Gemma Team @ Google DeepMind. Gemma 3 Technical Report, 2025. URL <https://arxiv.org/abs/2503.19786>.
- DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, 2025a. URL <https://arxiv.org/abs/2501.12948>.
- DeepSeek-AI. DeepSeek-V3 Technical Report, 2025b. URL <https://arxiv.org/abs/2412.19437>.
- István Endrédy and Attila Novák. More effective boilerplate removal-the goldminer algorithm. *Polibits*, 48:79–83, 12 2013. doi: 10.17562/PB-48-10.
- Steven Feng, Shrimai Prabhumoye, Kezhi Kong, Dan Su, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Maximize Your Data’s Potential: Enhancing LLM Accuracy with Two-Phase Pretraining, 2024. URL <https://arxiv.org/abs/2412.15285>.
- Shaona Ghosh, Prasoon Varshney, Makesh Narsimhan Sreedhar, Aishwarya Padmakumar, Traian Rebedea, Jibin Rajan Varghese, and Christopher Parisien. AEGIS2.0: A diverse AI safety dataset and risks taxonomy for alignment of LLM guardrails. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5992–6026, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.306. URL <https://aclanthology.org/2025.naacl-long.306/>.
- Xiaotian Han, Yiren Jian, Xuefeng Hu, Haogeng Liu, Yiqi Wang, Qihang Fan, Yuang Ai, Huaibo Huang, Ran He, Zhenheng Yang, and Quanzeng You. Infimm-webmath-40b: Advancing multimodal pre-training for enhanced mathematical reasoning, 2024. URL <https://arxiv.org/abs/2409.12568>.
- Adib Hasan, Ileana Rugina, and Alex Wang. Pruning for protection: Increasing jailbreak resistance in aligned llms without fine-tuning. *arXiv preprint arXiv:2401.10862*, 2024.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding, 2021a. URL <https://arxiv.org/abs/2009.03300>.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset, 2021b. URL <https://arxiv.org/abs/2103.03874>.
- Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekeshe, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What’s the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

- Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Xinrong Zhang, Zhen Leng Thai, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, dahai li, Zhiyuan Liu, and Maosong Sun. MiniCPM: Unveiling the potential of small language models with scalable training strategies. In *First Conference on Language Modeling*, 2024. URL <https://openreview.net/forum?id=3X2L2TFr0f>.
- Siming Huang, Tianhao Cheng, J. K. Liu, Jiaran Hao, Liuyihan Song, Yang Xu, J. Yang, Jiaheng Liu, Chenchen Zhang, Linzheng Chai, Ruifeng Yuan, Zhaoxiang Zhang, Jie Fu, Qian Liu, Ge Zhang, Zili Wang, Yuan Qi, Yinghui Xu, and Wei Chu. Opencoder: The open cookbook for top-tier code large language models, 2025. URL <https://arxiv.org/abs/2411.04905>.
- Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604–613, 1998.
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.
- Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From Live Data to High-Quality Benchmarks: The Arena-Hard Pipeline, April 2024a. URL <https://lmsys.org/blog/2024-04-19-arena-hard/>.
- Xuehai Li, Zi Ye, Xiaoxin Zhang, Xinshi Lu, Yingqiang Xia, Bairu Wu, Shihan Dong, Qipeng Jin, Jialu Wang, Heng Ji, et al. Wildchat: 1m chatgpt interaction logs in the wild. *arXiv preprint arXiv:2405.01470*, 2024b.
- Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. Jamba: A Hybrid Transformer-Mamba Language Model, 2024. URL <https://arxiv.org/abs/2403.19887>.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s Verify Step by Step. *arXiv preprint arXiv:2305.20050*, 2023.
- Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*, 2017.
- Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*, 2020.
- Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. *arXiv preprint arXiv:2305.01210*, 2023. doi: <https://doi.org/10.48550/arXiv.2305.01210>. URL <https://arxiv.org/abs/2305.01210>.
- Zuxin Liu et al. Toolace: Winning the points of llm function calling. *arXiv preprint arXiv:2409.00920*, 2024.

- Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, Tianyang Liu, Max Tian, Denis Kocetkov, Arthur Zucker, Younes Belkada, Zijian Wang, Qian Liu, Dmitry Abulkhanov, Indraneil Paul, Zhuang Li, Wen-Ding Li, Megan Risdal, Jia Li, Jian Zhu, Terry Yue Zhuo, Evgenii Zheltonozhskii, Nii Osae Osae Dade, Wenhao Yu, Lucas Krauß, Naman Jain, Yixuan Su, Xuanli He, Manan Dey, Edoardo Abati, Yekun Chai, Niklas Muennighoff, Xiangru Tang, Muhtasham Oblokulov, Christopher Akiki, Marc Marone, Chenghao Mou, Mayank Mishra, Alex Gu, Binyuan Hui, Tri Dao, Armel Zebaze, Olivier Dehaene, Nicolas Patry, Canwen Xu, Julian McAuley, Han Hu, Torsten Scholak, Sebastien Paquet, Jennifer Robinson, Carolyn Jane Anderson, Nicolas Chapados, Mostofa Patwary, Nima Tajbakhsh, Yacine Jernite, Carlos Muñoz Ferrandis, Lingming Zhang, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. Starcoder 2 and the stack v2: The next generation, 2024. URL <https://arxiv.org/abs/2402.19173>.
- Weidi Luo, Siyuan Ma, Xiaogeng Liu, Xiaoyu Guo, and Chaowei Xiao. Jailbreakv: A benchmark for assessing the robustness of multimodal large language models against jailbreak attacks. *arXiv preprint arXiv:2404.03027*, 2024.
- Rabeeh Karimi Mahabadi, Sanjeev Satheesh, Shrimai Prabhumoye, Mostofa Patwary, Mohammad Shoenybi, and Bryan Catanzaro. Nemotron-cc-math: A 133 billion-token-scale high quality math pretraining dataset, 2025. URL <https://arxiv.org/abs/2508.15096>.
- Somshubra Majumdar, Vahid Noroozi, Mehrzad Samadi, Sean Narenthiran, Aleksander Ficek, Wasi Uddin Ahmad, Jocelyn Huang, Jagadeesh Balam, and Boris Ginsburg. Genetic instruct: Scaling up synthetic generation of coding instructions for large language models. *arXiv preprint arXiv:2407.21077*, 2024.
- Llama Team @ Meta. The Llama 3 Herd of Models, 2024. URL <https://arxiv.org/abs/2407.21783>.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering, 2018. URL <https://arxiv.org/abs/1809.02789>.
- Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset. *arXiv preprint arXiv:2504.16891*, 2025.
- Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoenybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. Compact Language Models via Pruning and Knowledge Distillation, 2024. URL <https://arxiv.org/abs/2407.14679>.
- NVIDIA. Nemotron-4 340B Technical Report, 2024. URL <https://arxiv.org/abs/2406.11704>.
- NVIDIA. Nemotron-h: A family of accurate and efficient hybrid mamba-transformer models, 2025. URL <https://arxiv.org/abs/2504.03624>.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

- Jupinder Parmar, Shrimai Prabhumoye, Joseph Jennings, Mostofa Patwary, Sandeep Subramanian, Dan Su, Chen Zhu, Deepak Narayanan, Aastha Jhunjhunwala, Ayush Dattagupta, Vibhu Jawa, Jiwei Liu, Ameya Mahabaleshwarkar, Osvald Nitski, Annika Brundyn, James Maki, Miguel Martinez, Jiaxuan You, John Kamalu, Patrick LeGresley, Denys Fridman, Jared Casper, Ashwath Aithal, Oleksii Kuchaiev, Mohammad Shoeybi, Jonathan Cohen, and Bryan Catanzaro. Nemotron-4 15B Technical Report. *arXiv preprint arXiv:2402.16819*, 2024. URL <https://arxiv.org/abs/2402.16819>.
- Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. OpenWebMath: An Open Dataset of High-Quality Mathematical Web Text, 2023.
- Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. Fineweb2: One pipeline to scale them all – adapting pre-training data processing to every language, 2025. URL <https://arxiv.org/abs/2506.20920>.
- Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, Dmitry Dodonov, Tung Nguyen, Jaeho Lee, Daron Anderson, Mikhail Doroshenko, Alun Cennyth Stokes, Mobeen Mahmood, Oleksandr Pokutnyi, Oleg Iskra, Jessica P. Wang, John-Clark Levin, Mstyslav Kazakov, Fiona Feng, Steven Y. Feng, Haoran Zhao, Michael Yu, Varun Gangal, Chelsea Zou, Zihan Wang, Serguei Popov, Robert Gerbicz, Geoff Galgon, Johannes Schmitt, Will Yeadon, Yongki Lee, Scott Sauers, Alvaro Sanchez, Fabian Giska, Marc Roth, Søren Riis, Saiteja Utpala, Noah Burns, Gashaw M. Goshu, Mohinder Maheshbhai Naiya, Chidozie Agu, Zachary Giboney, Antrell Cheatom, Francesco Fournier-Facio, Sarah-Jane Crowson, Lennart Finke, Zerui Cheng, Jennifer Zampese, Ryan G. Hoerr, Mark Nandor, Hyunwoo Park, Tim Gehringer, Jiaqi Cai, Ben McCarty, Alexis C Garretson, Edwin Taylor, Damien Sileo, Qiuyu Ren, Usman Qazi, Lianghui Li, Jungbae Nam, John B. Wydallis, Pavel Arkhipov, Jack Wei Lun Shi, Aras Bacho, Chris G. Willcocks, Hangrui Cao, Sumeet Motwani, Emily de Oliveira Santos, Johannes Veith, Edward Vendrow, Doru Cojoc, Kengo Zenitani, Joshua Robinson, Longke Tang, Yuqi Li, Joshua Vendrow, Natanael Wildner Fraga, Vladyslav Kuchkin, Andrey Pupasov Maksimov, Pierre Marion, Denis Efremov, Jayson Lynch, Kaiqu Liang, Aleksandar Mikov, Andrew Gritsevskiy, Julien Guillod, Gözdenur Demir, Dakotah Martinez, Ben Pageler, Kevin Zhou, Saeed Soori, Ori Press, Henry Tang, Paolo Rissone, Sean R. Green, Lina Brüssel, Moon Twayana, Aymeric Dieuleveut, Joseph Marvin Imperial, Ameya Prabhu, Jinzhou Yang, Nick Crispino, Arun Rao, Dimitri Zvonkine, Gabriel Loiseau, Mikhail Kalinin, Marco Lukas, Ciprian Manolescu, Nate Stambaugh, Subrata Mishra, Tad Hogg, Carlo Bosio, Brian P Coppola, Julian Salazar, Jaehyeok Jin, Rafael Sayous, Stefan Ivanov, Philippe Schwaller, Shaipranesh Senthilkuma, Andres M Bran, Andres Algaba, Kelsey Van den Houte, Lynn Van Der Sypt, Brecht Verbeken, David Noever, Alexei Kopylov, Benjamin Myklebust, Bikun Li, Lisa Schut, Evgenii Zheltonozhskii, Qiaochu Yuan, Derek Lim, Richard Stanley, Tong Yang, John Maar, Julian Wykowski, Martí Oller, Anmol Sahu, Cesare Giulio Ardito, Yuzheng Hu, Ariel Ghislain Kemogne Kamdoun, Alvin Jin, Tobias Garcia Vilchis, Yuexuan Zu, Martin Lackner, James Koppel, Gongbo Sun, Daniil S. Antonenko, Steffi Chern, Bingchen Zhao, Pierrot Arsene, Joseph M Cavanagh, Daofeng Li, Jiawei Shen, Donato Crisostomi, Wenjin Zhang, Ali Dehghan, Sergey Ivanov, David Perrella, Nurdin Kaparov, Allen Zang, Ilia Sucholutsky, Arina Kharlamova, Daniil Orel, Vladislav Poritski, Shalev Ben-David, Zachary Berger, Parker Whitfill, Michael Foster, Daniel Munro, Linh Ho, Shankar Sivarajan, Dan Bar Hava, Aleksey Kuchkin, David Holmes, Alexandra Rodriguez-Romero, Frank Sommerhage, Anji Zhang, Richard Moat, Keith Schneider, Zakayo Kazibwe, Don Clarke, Dae Hyun Kim, Felipe Meneguitti Dias, Sara Fish, Veit Elser, Tobias Kreiman, Victor Efren Guadarrama Vilchis, Immo Klose, Ujjwala Anantheswaran, Adam Zweiger,

Kaivalya Rawal, Jeffery Li, Jeremy Nguyen, Nicolas Daans, Haline Heidinger, Maksim Radionov, Václav Rozhoň, Vincent Ginis, Christian Stump, Niv Cohen, Rafał Poświata, Josef Tkadlec, Alan Goldfarb, Chenguang Wang, Piotr Padlewski, Stanislaw Barzowski, Kyle Montgomery, Ryan Stendall, Jamie Tucker-Foltz, Jack Stade, T. Ryan Rogers, Tom Goertzen, Declan Grabb, Abhishek Shukla, Alan Givré, John Arnold Ambay, Archan Sen, Muhammad Fayez Aziz, Mark H Inlow, Hao He, Ling Zhang, Younesse Kaddar, Ivar Ångquist, Yanxu Chen, Harrison K Wang, Kalyan Ramakrishnan, Elliott Thornley, Antonio Terpin, Hailey Schoelkopf, Eric Zheng, Avishy Carmi, Ethan D. L. Brown, Kelin Zhu, Max Bartolo, Richard Wheeler, Martin Stehberger, Peter Bradshaw, JP Heimonen, Kaustubh Sridhar, Ido Akov, Jennifer Sandlin, Yury Makarychev, Joanna Tam, Hieu Hoang, David M. Cunningham, Vladimir Goryachev, Demosthenes Patramanis, Michael Krause, Andrew Redenti, David Aldous, Jesyin Lai, Shannon Coleman, Jiangnan Xu, Sangwon Lee, Ilias Magoulas, Sandy Zhao, Ning Tang, Michael K. Cohen, Orr Paradise, Jan Hendrik Kirchner, Maksym Ovchynnikov, Jason O. Matos, Adithya Shenoy, Michael Wang, Yuzhou Nie, Anna Szttyber-Betley, Paolo Faraboschi, Robin Riblet, Jonathan Crozier, Shiv Halasyamani, Shreyas Verma, Prashant Joshi, Eli Meril, Ziqiao Ma, Jérémy Andréoletti, Raghav Singhal, Jacob Platnick, Volodymyr Nevirkovets, Luke Basler, Alexander Ivanov, Seri Khoury, Nils Gustafsson, Marco Piccardo, Hamid Mostaghimi, Qijia Chen, Virendra Singh, Tran Quoc Khánh, Paul Rosu, Hannah Szlyk, Zachary Brown, Himanshu Narayan, Aline Menezes, Jonathan Roberts, William Alley, Kunyang Sun, Arkil Patel, Max Lamparth, Anka Reuel, Linwei Xin, Hanmeng Xu, Jacob Loader, Freddie Martin, Zixuan Wang, Andrea Achilleos, Thomas Preu, Tomek Korbak, Ida Bosio, Fereshteh Kazemi, Ziyue Chen, Biró Bálint, Eve J. Y. Lo, Jiaqi Wang, Maria Inês S. Nunes, Jeremiah Milbauer, M Saiful Bari, Zihao Wang, Behzad Ansarinejad, Yewen Sun, Stephane Durand, Hossam Elgnainy, Guillaume Douville, Daniel Tordera, George Balabanian, Hew Wolff, Lynna Kvistad, Hsiaoyun Milliron, Ahmad Sakor, Murat Eron, Andrew Favre D. O., Shailesh Shah, Xiaoxiang Zhou, Firuz Kamalov, Sherwin Abdoli, Tim Santens, Shaul Barkan, Allison Tee, Robin Zhang, Alessandro Tomasiello, G. Bruno De Luca, Shi-Zhuo Looi, Vinh-Kha Le, Noam Kolt, Jiayi Pan, Emma Rodman, Jacob Drori, Carl J Fossum, Niklas Muennighoff, Milind Jagota, Ronak Pradeep, Honglu Fan, Jonathan Eicher, Michael Chen, Kushal Thaman, William Merrill, Moritz Firsching, Carter Harris, Stefan Ciobăcă, Jason Gross, Rohan Pandey, Ilya Gusev, Adam Jones, Shashank Agnihotri, Pavel Zhelnov, Mohammadreza Mofayezi, Alexander Piperski, David K. Zhang, Kostiantyn Dobarskyi, Roman Leventov, Ignat Soroko, Joshua Duersch, Vage Taamazyan, Andrew Ho, Wenjie Ma, William Held, Ruicheng Xian, Armel Randy Zebaze, Mohanad Mohamed, Julian Noah Leser, Michelle X Yuan, Laila Yacar, Johannes Lengler, Katarzyna Olszewska, Claudio Di Fratta, Edson Oliveira, Joseph W. Jackson, Andy Zou, Muthu Chidambaram, Timothy Manik, Hector Haffenden, Dashiell Stander, Ali Dasouqi, Alexander Shen, Bitu Golshani, David Stap, Egor Kretov, Mikalai Uzhou, Alina Borisovna Zhidkovskaya, Nick Winter, Miguel Orbegoza Rodriguez, Robert Lauff, Dustin Wehr, Colin Tang, Zaki Hossain, Shaun Phillips, Fortuna Samuele, Fredrik Ekström, Angela Hammon, Oam Patel, Faraz Farhidi, George Medley, Forough Mohammadzadeh, Madellene Peñaflor, Haile Kassahun, Alena Friedrich, Rayner Hernandez Perez, Daniel Pyda, Taom Sakal, Omkar Dhamane, Ali Khajegili Mirabadi, Eric Hallman, Kenchi Okutsu, Mike Battaglia, Mohammad Maghsoudimehrabani, Alon Amit, Dave Hulbert, Roberto Pereira, Simon Weber, Handoko, Anton Peristy, Stephen Malina, Mustafa Mehkary, Rami Aly, Frank Reidegeld, Anna-Katharina Dick, Cary Friday, Mukhwinder Singh, Hassan Shapourian, Wanyoung Kim, Mariana Costa, Hubeyb Gurdogan, Harsh Kumar, Chiara Ceconello, Chao Zhuang, Haon Park, Micah Carroll, Andrew R. Tawfeek, Stefan Steinerberger, Daattavya Aggarwal, Michael Kirchhof, Linjie Dai, Evan Kim, Johan Ferret, Jainam Shah, Yuzhou Wang, Minghao Yan, Krzysztof Burdzy, Lixin Zhang, Antonio Franca, Diana T. Pham, Kang Yong Loh, Joshua Robinson, Abram Jackson, Paolo Giordano, Philipp Petersen, Adrian Cosma, Jesus Colino, Colin White, Jacob Votava, Vladimir Vinnikov, Ethan Delaney, Petr Spelda, Vit Stritecky, Syed M.

Shahid, Jean-Christophe Mourrat, Lavr Vetoshkin, Koen Sponselee, Renas Bacho, Zheng-Xin Yong, Florencia de la Rosa, Nathan Cho, Xiuyu Li, Guillaume Malod, Orion Weller, Guglielmo Albani, Leon Lang, Julien Laurendeau, Dmitry Kazakov, Fatimah Adesanya, Julien Portier, Lawrence Hollom, Victor Souza, Yuchen Anna Zhou, Julien Degorre, Yiğit Yalın, Gbenga Daniel Obikoya, Rai, Filippo Bigi, M. C. Boscá, Oleg Shumar, Kaniuar Bacho, Gabriel Recchia, Mara Popescu, Nikita Shulga, Ngefor Mildred Tanwie, Thomas C. H. Lux, Ben Rank, Colin Ni, Matthew Brooks, Alesia Yakimchyk, Huanxu, Liu, Stefano Cavalleri, Olle Häggström, Emil Verkama, Joshua Newbould, Hans Gundlach, Leonor Brito-Santana, Brian Amaro, Vivek Vajipey, Rynaa Grover, Ting Wang, Yosi Kratish, Wen-Ding Li, Sivakanth Gopi, Andrea Caciolai, Christian Schroeder de Witt, Pablo Hernández-Cámara, Emanuele Rodolà, Jules Robins, Dominic Williamson, Vincent Cheng, Brad Raynor, Hao Qi, Ben Segev, Jingxuan Fan, Sarah Martinson, Erik Y. Wang, Kaylie Hausknecht, Michael P. Brenner, Mao Mao, Christoph Demian, Peyman Kassani, Xinyu Zhang, David Avagian, Eshawn Jessica Scipio, Alon Ragoler, Justin Tan, Blake Sims, Rebeka Plecnik, Aaron Kirtland, Omer Faruk Bodur, D. P. Shinde, Yan Carlos Leyva Labrador, Zahra Adoul, Mohamed Zekry, Ali Karakoc, Tania C. B. Santos, Samir Shamseldeen, Loukmane Karim, Anna Liakhovitskaia, Nate Resman, Nicholas Farina, Juan Carlos Gonzalez, Gabe Maayan, Earth Anderson, Rodrigo De Oliveira Pena, Elizabeth Kelley, Hodjat Mariji, Rasoul Pouriamanesh, Wentao Wu, Ross Finocchio, Ismail Alarab, Joshua Cole, Danyelle Ferreira, Bryan Johnson, Mohammad Safdari, Liangti Dai, Siriphan Arthornthurasuk, Isaac C. McAlister, Alejandro José Moyano, Alexey Pronin, Jing Fan, Angel Ramirez-Trinidad, Yana Malysheva, Daphiny Pottmaier, Omid Taheri, Stanley Stepanic, Samuel Perry, Luke Askew, Raúl Adrián Huerta Rodríguez, Ali M. R. Minissi, Ricardo Lorena, Krishnamurthy Iyer, Arshad Anil Fasiludeen, Ronald Clark, Josh Ducey, Matheus Piza, Maja Somrak, Eric Vergo, Juehang Qin, Benjámín Borbás, Eric Chu, Jack Lindsey, Antoine Jallon, I. M. J. McInnis, Evan Chen, Avi Semler, Luk Gloor, Tej Shah, Marc Carauleanu, Pascal Lauer, Tran Duc Huy, Hossein Shahrtash, Emilien Duc, Lukas Lewark, Assaf Brown, Samuel Albanie, Brian Weber, Warren S. Vaz, Pierre Clavier, Yiyang Fan, Gabriel Poesia Reis e Silva, Long, Lian, Marcus Abramovitch, Xi Jiang, Sandra Mendoza, Murat Islam, Juan Gonzalez, Vasilios Mavroudis, Justin Xu, Pawan Kumar, Laxman Prasad Goswami, Daniel Bugas, Nasser Heydari, Ferenc Jeanplong, Thorben Jansen, Antonella Pinto, Archimedes Apronti, Abdallah Galal, Ng Ze-An, Ankit Singh, Tong Jiang, Joan of Arc Xavier, Kanu Priya Agarwal, Mohammed Berkani, Gang Zhang, Zhehang Du, Benedito Alves de Oliveira Junior, Dmitry Malishev, Nicolas Remy, Taylor D. Hartman, Tim Tarver, Stephen Mensah, Gautier Abou Loume, Wiktor Morak, Farzad Habibi, Sarah Hoback, Will Cai, Javier Gimenez, Roselynn Grace Montecillo, Jakub Łucki, Russell Campbell, Asankhaya Sharma, Khalida Meer, Shreen Gul, Daniel Espinosa Gonzalez, Xavier Alapont, Alex Hoover, Gunjan Chhablani, Freddie Vargus, Arunim Agarwal, Yibo Jiang, Deepakkumar Patil, David Outevsky, Kevin Joseph Scaria, Rajat Maheshwari, Abdelkader Dendane, Priti Shukla, Ashley Cartwright, Sergei Bogdanov, Niels Mündler, Sören Möller, Luca Arnaboldi, Kunvar Thaman, Muhammad Rehan Siddiqi, Prajvi Saxena, Himanshu Gupta, Tony Fruhauff, Glen Sherman, Mátyás Vincze, Siranut Usawasutsakorn, Dylan Ler, Anil Radhakrishnan, Innocent Enyekwe, Sk Md Salaudiddin, Jiang Muzhen, Aleksandr Maksapetyan, Vivien Rossbach, Chris Harjadi, Mohsen Bahaloohoreh, Claire Sparrow, Jasdeep Sidhu, Sam Ali, Song Bian, John Lai, Eric Singer, Justine Leon Uro, Greg Bateman, Mohamed Sayed, Ahmed Menshawy, Darling Duclosel, Dario Bezzi, Yashaswini Jain, Ashley Aaron, Murat Tiryakioglu, Sheeshram Siddh, Keith Krenek, Imad Ali Shah, Jun Jin, Scott Creighton, Denis Peskoff, Zienab EL-Wasif, Ragavendran P V, Michael Richmond, Joseph McGowan, Tejal Patwardhan, Hao-Yu Sun, Ting Sun, Nikola Zubić, Samuele Sala, Stephen Ebert, Jean Kaddour, Manuel Schottdorf, Dianzhuo Wang, Gerol Petruzella, Alex Meiburg, Tilen Medved, Ali ElSheikh, S Ashwin Hebbar, Lorenzo Vaquero, Xianjun Yang, Jason Poulos, Vilém Zouhar, Sergey Bogdanik, Mingfang Zhang, Jorge Sanz-Ros, David Anugraha, Yinwei Dai, Anh N. Nhu, Xue Wang, Ali Anil Demircali, Zhibai Jia, Yuyin Zhou, Juncheng

Wu, Mike He, Nitin Chandok, Aarush Sinha, Gaoxiang Luo, Long Le, Mickaël Noyé, Michał Perelkiewicz, Ioannis Pantidis, Tianbo Qi, Soham Sachin Purohit, Letitia Parcalabescu, Thai-Hoa Nguyen, Genta Indra Winata, Edoardo M. Ponti, Hanchen Li, Kaustubh Dhole, Jongee Park, Dario Abbondanza, Yuanli Wang, Anupam Nayak, Diogo M. Caetano, Antonio A. W. L. Wong, Maria del Rio-Chanona, Dániel Kondor, Pieter Francois, Ed Chalstrey, Jakob Zsambok, Dan Hoyer, Jenny Reddish, Jakob Hauser, Francisco-Javier Rodrigo-Ginés, Suchandra Datta, Maxwell Shepherd, Thom Kamphuis, Qizheng Zhang, Hyunjun Kim, Ruiji Sun, Jianzhu Yao, Franck Dernoncourt, Satyapriya Krishna, Sina Rismanchian, Bonan Pu, Francesco Pinto, Yingheng Wang, Kumar Shridhar, Kalon J. Overholt, Glib Briia, Hieu Nguyen, David, Soler Bartomeu, Tony CY Pang, Adam Wecker, Yifan Xiong, Fanfei Li, Lukas S. Huber, Joshua Jaeger, Romano De Maddalena, Xing Han Lù, Yuhui Zhang, Claas Beger, Patrick Tser Jern Kon, Sean Li, Vivek Sanker, Ming Yin, Yihao Liang, Xinlu Zhang, Ankit Agrawal, Li S. Yifei, Zechen Zhang, Mu Cai, Yasin Sonmez, Costin Cozianu, Changhao Li, Alex Slen, Shoubin Yu, Hyun Kyu Park, Gabriele Sarti, Marcin Briański, Alessandro Stolfo, Truong An Nguyen, Mike Zhang, Yotam Perlitz, Jose Hernandez-Orallo, Runjia Li, Amin Shabani, Felix Juefei-Xu, Shikhar Dhingra, Orr Zohar, My Chiffon Nguyen, Alexander Pondaven, Abdurrahim Yilmaz, Xuandong Zhao, Chuanyang Jin, Muyan Jiang, Stefan Todoran, Xinyao Han, Jules Kreuer, Brian Rabern, Anna Plassart, Martino Maggetti, Luther Yap, Robert Geirhos, Jonathon Kean, Dingsu Wang, Sina Mollaei, Chenkai Sun, Yifan Yin, Shiqi Wang, Rui Li, Yaowen Chang, Anjiang Wei, Alice Bizeul, Xiaohan Wang, Alexandre Oliveira Arrais, Kushin Mukherjee, Jorge Chamorro-Padial, Jiachen Liu, Xingyu Qu, Junyi Guan, Adam Bouyamourn, Shuyu Wu, Martyna Plomecka, Junda Chen, Mengze Tang, Jiaqi Deng, Shreyas Subramanian, Haocheng Xi, Haoxuan Chen, Weizhi Zhang, Yinuo Ren, Haoqin Tu, Sejong Kim, Yushun Chen, Sara Vera Marjanović, Junwoo Ha, Grzegorz Luczyna, Jeff J. Ma, Zewen Shen, Dawn Song, Cedegao E. Zhang, Zhun Wang, Gaël Gendron, Yunze Xiao, Leo Smucker, Erica Weng, Kwok Hao Lee, Zhe Ye, Stefano Ermon, Ignacio D. Lopez-Miguel, Theo Knights, Anthony Gitter, Namkyu Park, Boyi Wei, Hongzheng Chen, Kunal Pai, Ahmed Elkhanany, Han Lin, Philipp D. Siedler, Jichao Fang, Ritwik Mishra, Károly Zsolnai-Fehér, Xilin Jiang, Shadab Khan, Jun Yuan, Rishab Kumar Jain, Xi Lin, Mike Peterson, Zhe Wang, Aditya Malusare, Maosen Tang, Isha Gupta, Ivan Fosin, Timothy Kang, Barbara Dworakowska, Kazuki Matsumoto, Guangyao Zheng, Gerben Sewuster, Jorge Pretel Villanueva, Ivan Rannev, Igor Chernyavsky, Jiale Chen, Deepayan Banik, Ben Racz, Wenchao Dong, Jianxin Wang, Laila Bashmal, Duarte V. Gonçalves, Wei Hu, Kaushik Bar, Ondrej Bohdal, Atharv Singh Patlan, Shehzaad Dhuliawala, Caroline Geirhos, Julien Wist, Yuval Kansal, Bingsen Chen, Kutay Tire, Atak Talay Yücel, Brandon Christof, Veerupaksh Singla, Zijian Song, Sanxing Chen, Jiaxin Ge, Kaustubh Ponkshe, Isaac Park, Tianneng Shi, Martin Q. Ma, Joshua Mak, Sherwin Lai, Antoine Moulin, Zhuo Cheng, Zhanda Zhu, Ziyi Zhang, Vaidehi Patil, Ketan Jha, Qiutong Men, Jiaxuan Wu, Tianchi Zhang, Bruno Hebling Vieira, Alham Fikri Aji, Jae-Won Chung, Mohammed Mahfoud, Ha Thi Hoang, Marc Sperzel, Wei Hao, Kristof Meding, Sihan Xu, Vassilis Kostakos, Davide Manini, Yueying Liu, Christopher Toukmaji, Jay Paek, Eunmi Yu, Arif Engin Demircali, Zhiyi Sun, Ivan Dewerpe, Hongsen Qin, Roman Pflugfelder, James Bailey, Johnathan Morris, Ville Heilala, Sybille Rosset, Zishun Yu, Peter E. Chen, Woongyeong Yeo, Eeshaan Jain, Ryan Yang, Sreekar Chigurupati, Julia Chernyavsky, Sai Prajwal Reddy, Subhashini Venugopalan, Hunar Batra, Core Francisco Park, Hieu Tran, Guilherme Maximiano, Genghan Zhang, Yizhuo Liang, Hu Shiyu, Rongwu Xu, Rui Pan, Siddharth Suresh, Ziqi Liu, Samaksh Gulati, Songyang Zhang, Peter Turchin, Christopher W. Bartlett, Christopher R. Scotese, Phuong M. Cao, Aakaash Nattanmai, Gordon McKellips, Anish Cheraku, Asim Suhail, Ethan Luo, Marvin Deng, Jason Luo, Ashley Zhang, Kavin Jindel, Jay Paek, Kasper Halevy, Allen Baranov, Michael Liu, Advait Avadhanam, David Zhang, Vincent Cheng, Brad Ma, Evan Fu, Liam Do, Joshua Lass, Hubert Yang, Surya Sunkari, Vishruth Bharath, Violet Ai, James Leung, Rishit Agrawal, Alan Zhou, Kevin Chen, Tejas Kalpathi, Ziqi Xu, Gavin

- Wang, Tyler Xiao, Erik Maung, Sam Lee, Ryan Yang, Roy Yue, Ben Zhao, Julia Yoon, Sunny Sun, Aryan Singh, Ethan Luo, Clark Peng, Tyler Osbey, Taozhi Wang, Daryl Echeazu, Hubert Yang, Timothy Wu, Spandan Patel, Vidhi Kulkarni, Vijaykaarti Sundarapandiyam, Ashley Zhang, Andrew Le, Zafir Nasim, Srikar Yalam, Ritesh Kasamsetty, Soham Samal, Hubert Yang, David Sun, Nihar Shah, Abhijeet Saha, Alex Zhang, Leon Nguyen, Laasya Nagumalli, Kaixin Wang, Alan Zhou, Aidan Wu, Jason Luo, Anwith Telluri, Summer Yue, Alexandr Wang, and Dan Hendrycks. Humanity’s last exam, 2025. URL <https://arxiv.org/abs/2501.14249>.
- Akshara Prabhakar, Zuxin Liu, Weiran Yao, Jianguo Zhang, Ming Zhu, Thai Lan, Shirley Kokane, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, Rithesh Murthy, Liangwei Yang, Silvio Savarese, Juan Carlos Niebles, Shelby Heinecke, Huan Wang, and *et al.* Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. *arXiv preprint arXiv:2504.03601*, 2025.
- Qwen. Qwen2.5 Technical Report, 2025. URL <https://arxiv.org/abs/2412.15115>.
- 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.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023.
- Hayley Ross, Ameya Sunil Mahabaleshwarkar, and Yoshi Suhara. When2Call: When (not) to call tools. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3391–3409, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. URL <https://aclanthology.org/2025.naacl-long.174/>.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An Adversarial Winograd Schema Challenge at Scale, 2019. URL <https://arxiv.org/abs/1907.10641>.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. Language models are multilingual chain-of-thought reasoners, 2022. URL <https://arxiv.org/abs/2210.03057>.
- Shivalika Singh, Angelika Romanou, Cl  mentine Fourier, David I Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, et al. Global mmlu: Understanding and addressing cultural and linguistic biases in multilingual evaluation. *arXiv preprint arXiv:2412.03304*, 2024a.
- Shivalika Singh, Angelika Romanou, Cl  mentine Fourier, David I. Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Wei-Yin Ko, Madeline Smith, Antoine Bosselut, Alice Oh, Andre F. T. Martins, Leshem Choshen, Daphne Ippolito, Enzo Ferrante, Marzieh Fadaee, Beyza Ermis, and Sara Hooker. Global mmlu: Understanding and addressing cultural and linguistic biases in multilingual evaluation, 2024b. URL <https://arxiv.org/abs/2412.03304>.

- David R. So, Wojciech Mańke, Hanxiao Liu, Zihang Dai, Noam Shazeer, and Quoc V. Le. Primer: Searching for Efficient Transformers for Language Modeling, 2022. URL <https://arxiv.org/abs/2109.08668>.
- Sharath Turuvekere Sreenivas, Saurav Muralidharan, Raviraj Joshi, Marcin Chochowski, Ameya Sunil Mahabaleshwarkar, Gerald Shen, Jiaqi Zeng, Zijia Chen, Yoshi Suhara, Shizhe Diao, Chenhan Yu, Wei-Chun Chen, Hayley Ross, Oluwatobi Olabiyi, Ashwath Aithal, Oleksii Kuchaiev, Daniel Korzekwa, Pavlo Molchanov, Mostofa Patwary, Mohammad Shoeybi, Jan Kautz, and Bryan Catanzaro. LLM Pruning and Distillation in Practice: The Minitron Approach, 2024. URL <https://arxiv.org/abs/2408.11796>.
- Olly Styles, Sam Miller, Patricio Cerda-Mardini, Tanaya Guha, Victor Sanchez, and Bertie Vidgen. Workbench: a benchmark dataset for agents in a realistic workplace setting. *arXiv preprint arXiv:2405.00823*, 2024. doi: 10.48550/arXiv.2405.00823.
- Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-CC: Transforming Common Crawl into a refined long-horizon pretraining dataset. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2459–2475, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.123. URL <https://aclanthology.org/2025.acl-long.123/>.
- Ali Taghibakhshi, Sharath Turuvekere Sreenivas, Saurav Muralidharan, Marcin Chochowski, Yashaswi Karnati, Raviraj Joshi, Ameya Sunil Mahabaleshwarkar, Zijia Chen, Yoshi Suhara, Oluwatobi Olabiyi, et al. Efficient hybrid language model compression through group-aware ssm pruning. *arXiv preprint arXiv:2504.11409*, 2025.
- Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, Shengyan Liu, Di Luo, Yutao Ma, Hao Tong, Kha Trinh, Chenyu Tian, Zihan Wang, Bohao Wu, Yanyu Xiong, Shengzhu Yin, Minhui Zhu, Kilian Lieret, Yanxin Lu, Genglin Liu, Yufeng Du, Tianhua Tao, Ofir Press, Jamie Callan, Eliu Huerta, and Hao Peng. Scicode: A research coding benchmark curated by scientists, 2024. URL <https://arxiv.org/abs/2407.13168>.
- Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. *arXiv preprint arXiv:2410.01560*, 2024.
- Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, Garvit Kulshreshtha, Vartika Singh, Jared Casper, Jan Kautz, Mohammad Shoeybi, and Bryan Catanzaro. An Empirical Study of Mamba-based Language Models, 2024. URL <https://arxiv.org/abs/2406.07887>.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2024. URL <https://arxiv.org/abs/2212.03533>.
- Siyuan Wang, Zhongkun Liu, Wanjun Zhong, Ming Zhou, Zhongyu Wei, Zhumin Chen, and Nan Duan. From lsat: The progress and challenges of complex reasoning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2201–2216, 2022.

- Zhilin Wang, Jiaqi Zeng, Olivier Delalleau, Hoo-Chang Shin, Felipe Soares, Alexander Bukharin, Ellie Evans, Yi Dong, and Oleksii Kuchaiev. Helpsteer3-preference: Open human-annotated preference data across diverse tasks and languages. *arXiv preprint arXiv:2505.11475*, 2025.
- Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, 2022. URL <https://arxiv.org/abs/2203.05482>.
- Fanjia Yan, Huanzhi Mao, Charlie Cheng-Jie Ji, Tianjun Zhang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Berkeley Function Calling Leaderboard. 2024.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL <https://arxiv.org/abs/2505.09388>.
- Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. Rethinking benchmark and contamination for language models with rephrased samples, 2023.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a Machine Really Finish Your Sentence?, 2019. URL <https://arxiv.org/abs/1905.07830>.
- Biao Zhang and Rico Sennrich. Root Mean Square Layer Normalization, 2019. URL <https://arxiv.org/abs/1910.07467>.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Yonghao Li, Zhuohan Chen, Zhewei Wong, Siyuan Zhuang, Yakun Shao, Kai Xu, Zhenyu Zhang, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2309.11998*, 2023.
- Wanjuan Zhong, Siyuan Wang, Duyu Tang, Zenan Xu, Daya Guo, Yining Chen, Jiahai Wang, Jian Yin, Ming Zhou, and Nan Duan. Analytical reasoning of text. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 2306–2319, 2022.
- Fan Zhou, Zengzhi Wang, Nikhil Ranjan, Zhoujun Cheng, Liping Tang, Guowei He, Zhengzhong Liu, and Eric P Xing. Megamath: Pushing the limits of open math corpora. *arXiv preprint arXiv:2504.02807*, 2025.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A. Permissive Source Code Licenses

We remove source code with a license not in the following list:

3Com Microcode 3com-microcode, 3D Slicer License 1.0 [3dslicer-1.0], 4Suite 1.1 [4suite-1.1], AAL [attribution], Abstyles License [abstyles], ACE TAO License [ace-tao], AdaCore Doc License [adacore-doc], ADI BSD [adi-bsd], Adobe Glyph License [adobe-glyph], Adobe Postscript AFM License [apafml], Adobe Source Code License 2006 [adobe-scl], AES-128 3.0 License [aes-128-3.0], AFL 1.1 [afl-1.1], AFL 1.2 [afl-1.2], AFL 2.0 [afl-2.0], AFL 2.1 [afl-2.1], AFL 3.0 [afl-3.0], afmparse License [afmparse], Agere BSD [agere-bsd], Alexisisaac Freeware License [alexisisaac-freeware], Allegro 4 License [allegro-4], Altera License [xnet], Amazon Digital Services License [adsl], AMD Historical License [amd-historical], AMD PLPA License [amdplpa], AMPAS BSD-Style License [ampas], AMSFonts license [ams-fonts], Andre Adrian DFS license [adrian], ANTLR-PD [antlr-pd], ANTLR-PD with fallback [antlr-pd-fallback], ANU License [anu-license], Apache 1.0 [apache-1.0], Apache 1.1 [apache-1.1], Apache 2.0 [apache-2.0], Apache Patent Provision Exception Terms [apache-patent-exception], App::s2p License [app-s2p], Apple Attribution 1997 [apple-attribution-1997], Apple Attribution License [apple-attribution], Apple Example Code License [apple-excl], Apple MIT License [aml], Apple Sample Source Code License [apple-sscl], Aravindan Premkumar Licenase [aravindan-premkumar], ArgoUML License [argouml], ARM LLVM Grant [arm-llvm-sga], Array Input Method Public License [array-input-method-pl], Artistic 1.0 [artistic-1.0], Artistic 1.0 w/clause 8 [artistic-1.0-cl8], Artistic 2.0 [artistic-2.0], Artistic-Perl-1.0 [artistic-perl-1.0], ASMUS License [asmus], ASN.1 Object Dumping Code License [asn1], Atkinson Hyperlegible Font License [atkinson-hyperlegible-font], Baekmuk Fonts License [baekmuk-fonts], Bahyph License [bahyph], BaKoMa Fonts Licence 1995 [bakoma-fonts-1995], Barr TeX License [barr-tex], BEA 2.1 [bea-2.1], Beal Screamer License [beal-screamer], Beer-Ware License [beerware], BERI Hardware-Software License v1.0 [beri-hw-sw-1.0], BigDigits License [bigdigits], Bigelow & Holmes Lucida Fonts License [bigelow-holmes], Biopython License [biopython], Bitstream Vera Font License [bitstream], Bitzi-PD [bitzi-pd], BLAS License 2017 [blas-2017], Blue Oak Model License 1.0.0 [blueoak-1.0.0], BOHL-0.2 [bohl-0.2], Boost 1.0 [boost-1.0], Boost Original [boost-original], Borceux License [borceux], Boutell libgd declarations 2021 [boutell-libgd-2021], bpmn.io License [bpmn-io], Brent Corkum License [brent-corkum], Brian Clapper License [brian-clapper], Brian Gladman 3-Clause License [brian-gladman-3-clause], Brian Gladman Dual BSD-GPL [brian-gladman-dual], Brian Gladman License [brian-gladman], Broadcom CFE License [broadcom-cfe], Broadcom Warranty Disclaimer [broadcom-linux-timer], Brocade Firmware License [brocade-firmware], Bruno Podetti License [bruno-podetti], BSD 1988 [bsd-1988], BSD 3-Clause Devine [bsd-3-clause-devine], BSD 3-Clause FDA [bsd-3-clause-fda], BSD 3-Clause jtag [bsd-3-clause-jtag], BSD 3-Clause No Change [bsd-3-clause-no-change], BSD 3-Clause No Nuclear Warranty [bsd-3-clause-no-nuclear-warranty], BSD 3-Clause no trademark [bsd-3-clause-no-trademark], BSD 3-Clause Open MPI variant [bsd-3-clause-open-mpi], BSD 3-Clause Sun [bsd-3-clause-sun], BSD 3-Clause with GPL reference [bsd-top-gpl-addition], BSD Acknowledgment (Carrot2) License [bsd-ack-carrot2], BSD Acknowledgment License [bsd-ack], BSD Advertising Acknowledgement License [bsd-advertising-acknowledgement], BSD Artwork [bsd-artwork], BSD Atmel License [bsd-atmel], BSD DPT [bsd-dpt], BSD plus modification notice [bsd-plus-mod-notice], BSD Simplified Darwin [bsd-simplified-darwin], BSD Source Code Attribution [bsd-source-code], BSD Unchanged [bsd-unchanged], BSD Unmodified [bsd-unmodified], BSD Zero Clause License [bsd-zero], BSD-1-Clause [bsd-1-clause], BSD-1-Clause Build [bsd-1-clause-build], BSD-2-Clause [bsd-simplified], BSD-2-Clause no disclaimer [bsd-no-disclaimer], BSD-2-Clause no disclaimer Unmod [bsd-no-disclaimer-unmodified], BSD-2-Clause Plus Patent [bsd-plus-patent], BSD-2-Clause-plus-advertizing [bsd-2-clause-plus-advertizing], BSD-2-Clause-Views [bsd-2-clause-views], BSD-3-Clause [bsd-new], BSD-3-Clause tcpdump variant [bsd-new-tcpdump], BSD-3-Clause without notice modification [bsd-new-nomod], BSD-3-Clause X11 disclaimer [bsd-x11], BSD-4-Clause with Voices [bsd-original-voices], BSD-4-Clause-Shortened [bsd-4-clause-shortened], BSD-Axis without modification [bsd-axis-nomod], BSD-Credit [bsd-credit], BSD-Derivative [bsd-new-derivative], BSD-Export [bsd-export], BSD-InnoSys [bsd-innosys], BSD-Mylex [bsd-mylex], BSD-Original [bsd-original],

BSD-Original-Muscle [bsd-original-muscle], BSD-Original-UC [bsd-original-uc], BSD-Original-UC-1986 [bsd-original-uc-1986], BSD-Simplified Intel [bsd-simplified-intel], BSD-Simplified source [bsd-simplified-source], BSD-Top [bsd-top], BSLA [bsla], BSLA no advertizing [bsla-no-advert], Business Source License 1.0 [bsl-1.0], BYTEmark License [bytemark], bzip2 License 2010 [bzip2-libbzip-2010], Caldera License [caldera], Careware [careware], Carnegie Mellon Contributors [carnegie-mellon-contributors], Carnegie Mellon License [carnegie-mellon], Cavium malloc License [cavium-malloc], CC-BY-1.0 [cc-by-1.0], CC-BY-2.0 [cc-by-2.0], CC-BY-2.0-UK [cc-by-2.0-uk], CC-BY-2.5 [cc-by-2.5], CC-BY-3.0 [cc-by-3.0], CC-BY-3.0-AT [cc-by-3.0-at], CC-BY-3.0-US [cc-by-3.0-us], CC-BY-4.0 [cc-by-4.0], CC-PD [cc-pd], CC-PD Mark 1.0 [cc-pdm-1.0], CC0-1.0 [cc0-1.0], CDLA Permissive 1.0 [cdla-permissive-1.0], CDLA Permissive 2.0 [cdla-permissive-2.0], CeCILL-B License [cecill-b], CeCILL-B License English [cecill-b-en], CERN Attribution 1995 [cern-attribution-1995], CERN Open Hardware Licence v1.2 [cern-ohl-1.2], CERN Open Hardware License v1.1 [cern-ohl-1.1], CERN-OHL-P-2.0 [cern-ohl-p-2.0], CFITSIO License [cfitsio], Checkmk License [checkmk], Chicken Dance License v0.2 [chicken-dl-0.2], Chris Maunder License [chris-maunder], Chris Stoy Attribution License [chris-stoy], Clarified Artistic License [artistic-clarified], Classic VB License [classic-vb], Clear BSD 1-Clause License [clear-bsd-1-clause], Clear BSD License [clear-bsd], Click License [click-license], CLIPS License 2017 [clips-2017], CMU Computing Services License [cmu-computing-services], CMU License [cmu-template], CMU MIT-style [cmu-mit], CMU Simple License [cmu-simple], CMU Style [cmu-uc], CNRI Jython License [cnri-jython], CNRI Python 1.6 [cnri-python-1.6], CNRI Python 1.6.1 [cnri-python-1.6.1], Code Credit License v1.0.1 [code-credit-license-1.0.1], Code Credit License v1.1.0 [code-credit-license-1.1.0], CodeGuru Permissions [codeguru-permissions], CodeSourcery 2004 [codesourcery-2004], COIL-1.0 [coil-1.0], Common Lisp LOOP License [loop], CommonJ Timer License [commonj-timer], Compass License [compass], ComponentAce JCraft License [componentace-jcraft], compuphase Linking Exception to Apache 2.0 [compuphase-linking-exception], Condor Public License 1.1 [condor-1.1], Copyheart [copyheart], Cornell Lossless JPEG License [cornell-lossless-jpeg], Cougaar Open Source License [cosl], CP/M License 2022 [cpm-2022], CppCoreGuidelines License [cpp-core-guidelines], CRCalc license [crcalc], Creative Commons Attribution 2.5 Australia [cc-by-2.5-au], Creative Commons Attribution 3.0 Germany [cc-by-3.0-de], Creative Commons Attribution 3.0 Netherlands [cc-by-3.0-nl], Crossword License [crossword], Crypto++ License [cryptopp], Crystal Stacker License [crystal-stacker], CSL-1.0 [csl-1.0], CSPRNG [csprng], Cube License [cube], cURL License [curl], CVE ToU [cve-tou], CWE ToU [cwe-tou], CxImage License [cximage], D Zlib [d-zlib], DAMAIL [damail], Dante Treglia License [dante-treglia], DBAD License 1.1 [dbad-1.1], Debian reportbug License [reportbug], Delorie Historical License [delorie-historical], dhtmlab Public License [dhtmlab-public], diffmark License [diffmark], dl-de/by-1-0-de [dl-de-by-1-0-de], dl-de/by-1-0-en [dl-de-by-1-0-en], dl-de/by-2-0-de [dl-de-by-2-0-de], dl-de/by-2-0-en [dl-de-by-2-0-en], dmalloc License [dmalloc], DMTF License 2017 [dmftf-2017], Docbook License [docbook], Dom4j License [dom4j], Dotseqn License [dotseqn], Douglas Young License [douglas-young], DRL-1.0 [drl-1.0], DRL-1.1 [drl-1.1], Dropbear License [dropbear], Dropbear-2016 [dropbear-2016], DSDP License [dsdp], Dtree License [dtree], dvipdfm License [dvipdfm], DWTFNMFPL-3.0 [dwtfnmfpl-3.0], Dynamic Drive TOU [dynamic-drive-tou], ECL 1.0 [ecl-1.0], ECL 2.0 [ecl-2.0], EFL 1.0 [efl-1.0], EFL 2.0 [efl-2.0], EFL MIT-Style License [enlightenment], eGenix Public License 1.0.0 [egenix-1.0.0], eGenix Public License 1.1.0 [egenix-1.1.0], EllisLab License [ellis-lab], EMX Library License [emx-library], EnergyPlus BSD-Style License [energyplus-bsd], Enhanced MIT License [emit], enna License [enna], Entessa 1.0 [entessa-1.0], ePaperPress License [epaperpress], EPICS Open License [epics], Eric Glass License [eric-glass], Errbot exception [errbot-exception], Etalab Open License 2.0 [etalab-2.0], Etalab Open License 2.0 English [etalab-2.0-en], EU DataGrid Software License [eu-datagrid], Fabien Tassin License [fabien-tassin], Fair License [fair], FAL 1.3 [free-art-1.3], Far Manager exception to BSD-3-Clause [far-manager-exception], FASTBuild License 2012-2020 [fastbuild-2012-2020], FastCGI DevKit [fastcgi-devkit], FastCGI License for Spec Implementation [openmarket-fastcgi], FatFs

License [fatfs], FFTPack License 2004 [fftpack-2004], Filament Group MIT License [filament-group-mit], Flex 2.5 [flex-2.5], Flora License v1.1 [flora-1.1], font-alias License [font-alias], FPLot License [fplot], Fraunhofer ISO 14496-10 License [fraunhofer-iso-14496-10], FreeBSD Boot [freebsd-boot], FreeBSD Doc License [freebsd-doc], FreeBSD unmodified first lines License [freebsd-first], FreeMarker License [freemarker], FreeTTS License [freetts], FreeType Project License [freetype], Freeware Public License (FPL) [fpl], FSF All Permissive License [fsf-ap], FSF Free Software License [fsf-free], FSF Notice [fsf-notice], FSF Unlimited License No Warranty [fsf-unlimited-no-warranty], FSF-Unlimited [fsf-unlimited], Fujion Clinical Exception to Apache 2.0 [fujion-exception-to-apache-2.0], Gareth McCaughan License [gareth-mccaughan], Gary S. Brown License [gary-s-brown], GDCL License [gdcl], Generic patent disclaimer [patent-disclaimer], Geoff Kuenning License 1993 [geoff-kuenning-1993], Ghostpdl Permissive [ghostpdl-permissive], Glulxe License [glulxe], GLUT License [glut], GLWTPL [glwtpl], Good Boy License [good-boy], Graphics Gems License [graphics-gems], Greg Roelofs License [greg-roelofs], Gregory Pietsch Liberal License [gregory-pietsch], GStreamer Exception (2005) [gststreamer-exception-2005], GTPL-v1 [gtpl-v1], GTPL-v2 [gtpl-v2], GTPL-v3 [gtpl-v3], Haskell Report License [haskell-report], HDF4 License [hdf4], HDF5License [hdf5], HDPARM License [hdparm], Henry Spencer License 1999 [henry-spencer-1999], Henry Spencer Regexp License [hs-regexp], HIDAPI License [hidapi], Historical Notice - NTP [historical-ntp], Historical Permission Notice and Disclaimer [historical], Homebrewed License [homebrewed], HP 1986 License [hp-1986], HPND sell variant with MIT disclaimer [hpnd-sell-variant-mit-disclaimer], HTML 5 spec License [html5], httpget notice and disclaimer [httpget], Ian Kaplan License [ian-kaplan], Ian Piumarta License [ian-piumarta], IBM AS-IS License [ibm-as-is], IBM DHCP License [ibm-dhcp], IBM Non-Warranted Sample Code License [ibm-nwsc], IBM PowerPC Software [ibm-pibs], IBM Sample License [ibm-sample], IBPP License [ibpp], ICANN-Public [icann-public], ICOT Free Software [icot-free], ICU Composite License [ibm-icu], ICU License 58 and later [unicode-icu-58], IDT License Notice [idt-notice], IETF License [ietf], IETF Trust License [ietf-trust], ilmid License [ilmid], ImageMagick License [imagemagick], Independent JPEG Group License - short [ijg-short], Indiana Extreme License 1.1.1 [indiana-extreme], Indiana Extreme License 1.2 [indiana-extreme-1.2], Infineon Free Software License [infineon-free], Info-Zip License 1997-10 [info-zip-1997-10], Info-Zip License 2001-01 [info-zip-2001-01], Info-Zip License 2002-02 [info-zip-2002-02], Info-Zip License 2003-05 [info-zip-2003-05], Info-Zip License 2004-05 [info-zip-2004-05], Info-Zip License 2005-02 [info-zip-2005-02], Info-Zip License 2007-03 [info-zip-2007-03], Info-Zip License 2009-01 [info-zip-2009-01], Info-Zip License [info-zip], Inno Setup License [inno-setup], Intel ACPI SLA [intel-acpi], Intel BSD - Export Control [intel-bsd-export-control], Intel BSD 2 Clause License [intel-bsd-2-clause], Intel BSD License [intel-bsd], Intel Limited Patent License [intel], Intel OSL 1989 [intel-osl-1989], Intel OSL 1993 [intel-osl-1993], Intel Royalty Free License [intel-royalty-free], ISC License [isc], ISO 14496-10 [iso-14496-10], ISO 8879 [iso-8879], ITU License [itu], JA-SiG License [ja-sig], Jam License [jam], Jason Mayes License [jason-mayes], Jasper 1.0 [jasper-1.0], JasPer 2.0 [jasper-2.0], Java App Stub License [java-app-stub], JDBM License v1.00 [jdbm-1.00], JDOM License [jdom], Jetty License [jetty], JGraph License [jgraph], JPEG License [ijg], JPNIC idnkit License [jpnidnkit], JPNIC mdnkit License [jpnidnkit], JPython 1.1 [jpython-1.1], jQuery-Tools-PD [jquery-pd], Jscheme License [jscheme], JSFromHell License [jsfromhell], JSON License [json], JSON-js-PD [json-js-pd], JSON-PD [json-pd], Jython License [jython], Kalle Kaukonen License [kalle-kaukonen], Kazlib [kazlib], Keith Rule License [keith-rule], Kerberos License [kerberos], Kevan Stannard License [kevan-stannard], Kevlin Henney License [kevin-henney], Khronos License [khronos], Knuth CTAN License [knuth-ctan], Kumar Robotics License [kumar-robotics], latex-ec-fonts [ecfonts-1.0], Latex2e License [latex2e], Latex2e with translated notice permission [latex2e-translated-notice], LBNL BSD Variant [lbnl-bsd], LCS-Telegraphics License [lcs-telegraphics], Leptonica License [leptonica], libgd License 2018 [libgd-2018], libgeoTiff License [libgeotiff], LibMib License [libmib], libmng License 2007 [libmng-2007], Libpng License [libpng], Llibpng License v2 [libpng-v2], libselinux License [libselinux-pd],

libsrvc License v1.0.2 [libsrvc-1.0.2], Lil License v1 [lil-1], LILO License [lilo], Linux Device Drivers [linux-device-drivers], Linux-OpenIB [linux-openib], LinuxBIOS License [linuxbios], linuxhowtos License [linuxhowtos], LLNL [llnl], LLVM Exception to Apache 2.0 [llvm-exception], Logica OSL 1.0 [logica-1.0], LPPL 1.3c [lppl-1.3c], Lucent Public License 1.0 [lucent-pl-1.0], Lucent Public License 1.02 [lucent-pl-1.02], Lucre License [lucre], LZMA SDK License (versions 9.22 and beyond) [lzma-sdk-9.22], LZMA SDK Public Domain [lzma-sdk-pd], M+ Fonts license [m-plus], MakeHuman License [make-human-exception], Markus Kuhn License [markus-kuhn-license], Martin Bergmeier License [martin-birgmeier], Matrix Template Library License [mtll], Matt Gallagher Attribution License [matt-gallagher-attribution], Matt Kruse License [mattkruse], Matthew Kwan License [matthewkwan], MediaInfo(Lib) License [mediainfo-lib], metamail License [metamail], MgOpen Font License [mgopen-font-license], Michael Barr License [michael-barr], Minpack Copyright Notice [minpack], MirOS License [mir-os], MIT (SEI) [vince], MIT 1995 [mit-1995], MIT Acknowledgment License [mit-ack], MIT Addition License [mit-addition], MIT License 1998 [mit-license-1998], MIT License [mit], MIT Modern Variant [mit-modern], MIT Nagy Variant [mit-nagy], MIT no advertising with Export Control [mit-no-advert-export-control], MIT No Commercial Use of Trademarks [mit-no-trademarks], MIT no false attribution License [mit-no-false-attribs], MIT Old Style [mit-old-style], MIT Old Style no advertising [mit-old-style-no-advert], MIT Old Style Spare [mit-old-style-sparse], MIT README License [mit-readme], MIT Synopsys License [mit-synopsys], MIT Taylor Variant [mit-taylor-variant], MIT Veillard Variant [mit-veillard-variant], MIT with Export Control [mit-export-control], MIT with Specification Disclaimer [mit-specification-disclaimer], MIT Xfig Variant [mit-xfig], MIT-0-Clause [mit-0], mod_dav License 1.0 [mod-dav-1.0], Modified MIT License for Public Domain software [pd-mit], Motorola Microprocessor License [motorola], Mozilla GC License [mozilla-gc], MPEG SSG License [mpeg-ssg], MPEG-2 NBC MPEG-4 Audio ISO [mpeg-iso], MPICH License [mpich], MS Systems Journal Sample Code License [msj-sample-code], MS WS Routing Specifications License [ms-ws-routing-spec], MS-LPL [ms-lpl], MS-PL [ms-pl], MS-SS-PL [ms-sspl], Mulan PSL v1 [mulanpsl-1.0], Mulan PSL v1.0 (En) [mulanpsl-1.0-en], Mulan PSL v2 [mulanpsl-2.0], Mulan PSL v2.0 (En) [mulanpsl-2.0-en], Mülle Kybernetik License [mulle-kybernetik], Multics License [multics], Mup License [mup], musl attribution exception [musl-exception], MX4J License 1.0 [mx4j], Nara Institute License 2003 [naist-2003], NASA 1.3 [nasa-1.3], NAUMEN Public License [naumen], NBPL-1.0 [nbpl-1.0], NCBI Public Domain Notice [ncbi], NCSA Open Source License [uoi-ncsa], Net SNMP License [net-snmp], Netcat License [netcat], NetCDF License [netcdf], Netron Project License [netron], Newlib Historical License [newlib-historical], Newran License [newran], Newsletr License [newsletr], Nice License [nice], NICTA Public Software Licence 1.0 [nicta-psl], Niels Ferguson License [niels-ferguson], Nilsson Historical License [nilsson-historical], NIST Public Domain Notice [nist-pd], NIST Public Domain Notice with fallback [nist-pd-fallback], NIST Software License [nist-software], NIST SRD License [nist-srd], NLOD-1.0 [nlod-1.0], NLOD-2.0 [nlod-2.0], NLPL [nlpl], Node License [node-js], Non White Heterosexual Male [nwhm], Nonexclusive License [nonexclusive], Nortel DASA License [nortel-dasa], Notre Dame License [notre-dame], NRL License [nrl], NRL permission [nrl-permission], NTLM License [ntlm], NTP Origin License [ntpl-origin], NTP-0 [ntp-0], NVIDIA 2002 License [nvidia-2002], NVIDIA License [nvidia], NVIDIA License with Government Qualifications [nvidia-gov], NYSL 0.9982 [nysl-0.9982], NYSL 0.9982 JP [nysl-0.9982-jp], O Young Jong License [o-young-jong], O'Reilly Code Sample Notice [oreilly-notice], O-UDA-1.0 [o-uda-1.0], Oasis WS Security Specification License [oasis-ws-security-spec], Object Form Exception to MIT [object-form-exception-to-mit], ODC-By-1.0 [odc-by-1.0], ODMG License [odmg], OFFIS License [offis], OFL 1.0 [off-1.0], OFL 1.0 no Reserved Font Name [off-1.0-no-rfn], OFL 1.0 Reserved Font Name [off-1.0-rfn], OFL 1.1 no Reserved Font Name [off-1.1-no-rfn], OGC 1.0 [ogc-1.0], OGC Software Notice [ogc], OGL 1.0a [ogl-1.0a], OGL Alberta 2.1 [can-ogl-alberta-2.1], OGL British Columbia 2.0 [can-ogl-british-columbia-2.0], OGL Canada 2.0 [can-ogl-2.0-en], OGL Canada 2.0 Francais [ogl-canada-2.0-fr], OGL Nova Scotia 1.0 [can-ogl-nova-scotia-1.0],

OGL Ontario 1.0 [can-ogl-ontario-1.0], OGL Toronto 1.0 [can-ogl-toronto-1.0], OGL-UK-1.0 [ogl-uk-1.0], OGL-UK-2.0 [ogl-uk-2.0], OGL-UK-3.0 [ogl-uk-3.0], OGL-WPD-3.0 [ogl-wpd-3.0], Open Directory License [odl], Open Group Test Suite License [opengroup], Open Publication License 1.0 [openpub], OpenLDAP Public License 1.1 [openldap-1.1], OpenLDAP Public License 1.2 [openldap-1.2], OpenLDAP Public License 1.3 [openldap-1.3], OpenLDAP Public License 1.4 [openldap-1.4], OpenLDAP Public License 2.0 [openldap-2.0], OpenLDAP Public License 2.0.1 [openldap-2.0.1], OpenLDAP Public License 2.1 [openldap-2.1], OpenLDAP Public License 2.2 [openldap-2.2], OpenLDAP Public License 2.2.1 [openldap-2.2.1], OpenLDAP Public License 2.2.2 [openldap-2.2.2], OpenLDAP Public License 2.3 [openldap-2.3], OpenLDAP Public License 2.4 [openldap-2.4], OpenLDAP Public License 2.5 [openldap-2.5], OpenLDAP Public License 2.6 [openldap-2.6], OpenLDAP Public License 2.7 [openldap-2.7], OpenLDAP Public License 2.8 [openldap-2.8], OpenORB Community License 1.0 [openorb-1.0], OpenSAML License v1 [opensaml-1.0], OpenSSH License [openssh], OpenSSL License [openssl], OpenSSL/SSLeay License [openssl-ssleay], OPML 1.0 [opml-1.0], OPNL-1.0 [opnl-1.0], OPNL-2.0 [opnl-2.0], Oracle BSD-Style with Nuclear Restrictions [oracle-bsd-no-nuclear], Original SSLeay License [ssleay], Original SSLeay License with Windows Clause [ssleay-windows], Oswego Concurrent License [oswego-concurrent], Other Permissive Licenses [other-permissive], OWTChart License [owtchart], OZPLB 1.0 [ozplb-1.0], OZPLB 1.1 [ozplb-1.1], Paolo Messina 2000 [paolo-messina-2000], ParaView License 1.2 [paraview-1.2], Paul Mackerras Binary License [paul-mackerras-binary], Paul Mackerras License [paul-mackerras], Paul Mackerras New License [paul-mackerras-new], Paul Mackerras Simplified License [paul-mackerras-simplified], Paulo Soares License [paulo-soares], PayPal SDK License 2013-2016 [paypal-sdk-2013-2016], PBM Library License [libpbm], PCRE License [pcre], PD'Programming License [pd-programming], PDDL 1.0 [pddl-1.0], Perl 1.0 [perl-1.0], Peter Deutsch Document License [peter-deutsch-document], Phil Bunce License [phil-bunce], Philippe De Muyter License [philippe-de-muyter], Phorum License 2.0 [phorum-2.0], PHP License 2.0.2 [php-2.0.2], PHP License 3.0 [php-3.0], PHP License 3.01 [php-3.01], Pine License [pine], PngSuite License [pngsuite], Politepix Public License 1.0 [politepix-pl-1.0], PostgreSQL License [postgresql], ppp License [ppp], Protobuf License [protobuf], PS Utilities License [psutils], PSF Python License 3.7.2 [psf-3.7.2], PSF-2.0 [psf-2.0], psfrag License [psfrag], Psytec Free Software License [psytec-freesoft], Public Domain [public-domain], Public Domain Disclaimer [public-domain-disclaimer], Purdue BSD-Style License [purdue-bsd], pybench License [pybench], PyCrypto License [pycrypto], PyGres License 2.2 [pygres-2.2], Python CWI License [python-cwi], Python License 2.0 [python], Python License 2.0.1 [python-2.0.1], Qhull License [qhull], QLogic Microcode [qlogic-microcode], Qpopper License [qpopper], Qualcomm Turing License [qualcomm-turing], Quirksmode Copyright Notice [quirksmode], radvd License [radvd], Rdisc License [rdisc], Red Hat Attribution License [red-hat-attribution], Red Hat BSD-Simplified [red-hat-bsd-simplified], Regexp License [regexp], Repoze License [repoze], RiceBSD [ricebsd], Richard Black License [richard-black], Robert Hubley License [robert-hubley], RSA 1990 [rsa-1990], RSA Cryptoki License [rsa-cryptoki], RSA Demo License [rsa-demo], RSA-MD4 License [rsa-md4], RSA-MD5 License [rsa-md5], RTools.Util License [rtools-util], Ruby License [ruby], Runtime Library Exception to Apache 2.0 [apple-runtime-library-exception], Rute Users Tutorial and Exposition License 0.8.0 [rute], Ryszard Szopa License [ryszard-szopa], SaaS MIT License [saas-mit], Sash Notice [sash], SATA License [sata], SAX-PD [sax-pd], Saxpath License [saxpath], SBIA Part B [sbia-b], ScanCode acknowledgment [scancode-acknowledgment], scanlogd License [scanlogd-license], ScanSoft Public License 1.2 [scansoft-1.2], SCEA Shared Source License 1.0 [scea-1.0], Scheme Language Report License [schemereport], Scheme Widget Library (SWL) Software License [swl], Scintilla License [scintilla], Scribbles Demos Recognizer Notice [scribbles], Script Asylum License [script-asylum], Secret Labs License 2011 [secret-labs-2011], selinux-nsa-declaration-1.0 [selinux-nsa-declaration-1.0], Sendmail License [sendmail], Service Availability Forum License [saf], Service Component Architecture License [service-comp-arch], SFL License Agreement [sfl-license], SGI CID Font Code Public License 1.0 [sgi-cid-1.0], SGI Free

Software License B 1.1 [sgi-freeb-1.1], SGI Free Software License B 2.0 [sgi-freeb-2.0], SGI GLX Public License 1.0 [sgi-glx-1.0], Sglib License [sglib], SGP4 Permission Notice [sgp4], Shital Shah License [shital-shah], SIL Open Font License 1.1 with Reserved Font Name [ofl-1.1-rfn], SimPL 1.1 [simpl-1.1], SNMP++ License [hp-snmp-pp], snprintf License [snprintf], SoftFloat [softfloat], SoftFloat Legal Notice 2.0 [softfloat-2.0], softSurfer License [softsurfer], SolderPad Hardware License v0.5 [shl-0.5], Solderpad Hardware License v2.0 [shl-2.0], Solderpad Hardware License v2.1 [shl-2.1], SolderPad Hardware License, Version 0.51 [shl-0.51], Sparky License [sparky], SpeechWorks Public License 1.1 [speechworks-1.1], SQLite Blessing [blessing], Standard ML of New Jersey [standard-ml-nj], Stanford PVRG License [stanford-pvrg], STLport License 2000 [stlport-2000], STLport License 4.5 [stlport-4.5], STREAM Benchmark License [stream-benchmark], Stu Nicholls License [stu-nicholls], Sun RPC License [sun-rpc], Sun source code License [sun-source], SunPro Attribution License [sunpro], Sunsoft License [sunsoft], Supervisor License [supervisor], svndiff License [svndiff], SWIG Library License [swig], Symlinks License [symlinks], Symphonysoft [symphonysoft], Synopsys MIT License [synopsys-mit], Synthesis Toolkit License [synthesis-toolkit], SystemC Open Source License Agreement [accellera-systemc], Taiwan Open Government Data License, version 1.0 [ogdl-taiwan-1.0], Takao Abe License [takao-abe], Takuya OOURA License [takuya-oura], Talis Community License [ttcl], Tatu Ylonen License [tatu-ylonen], TCG Spec License v1 [tcg-spec-license-v1], TCL/TK License [tcl], TCP Wrappers License [tcp-wrappers], TekHVC License [tekhvc], Term Readkey License [term-readkey], Tested Software License [tested-software], TeX Live License [tex-live], Text-Tabs+Wrap License [ttwl], TFL [tfl], The Happy Bunny License [happy-bunny], Theodore Ts'o license [tso-license], Things I Made (TIM) Public License [things-i-made-public-license], Tidy License [tidy], Tiger Cryptography License [tiger-crypto], Tigra Calendar 3.2 License [tigra-calendar-3.2], Tigra Calendar 4.0 License [tigra-calendar-4.0], Tim Janik License 2003 [tim-janik-2003], Time::ParseDate License [tpdl], Timestamp Picker License [timestamp-picker], TTYPE0 License [ttyp0], TU Berlin License 1.0 [tu-berlin], TU Berlin License 2.0 [tu-berlin-2.0], Tumbolia Public License [tumbolia], TwistedSNMP License [twisted-snmp], UCAR License [ucar], UnboundID LDAP SDK Free Use License [ldap-sdk-free-use], Unicode DFS 2015 [unicode-dfs-2015], Unicode DFS 2016 [unicode-dfs-2016], Unicode Inc License Agreement [unicode], Unicode Mappings License [unicode-mappings], University of British Columbia License [ubc], University of Michigan OSL [michigan-disclaimer], UNIX Network Programming Book License [unpbook], UnixCrypt License [unixcrypt], Unlicense [unlicense], Unlimited Binary Use Exception [unlimited-binary-use-exception], UPL 1.0 [upl-1.0], US Government Public Domain [us-govt-public-domain], US Government Unlimited Rights [us-govt-unlimited-rights], USRobotics Permissive License [usrobotics-permissive], Utopia Typeface License [utopia], VCalendar License [vcalendar], Vic Metcalfe Public Domain [vic-metcalfe-pd], VIM License [vim], Visual Idiot [visual-idiot], Visual Numerics License [visual-numerics], Vixie Cron License [vixie-cron], Vovida Software License 1.0 [vsl-1.0], W3C 3-Clause BSD License [w3c-03-bsd-license], W3C Software Notice and License [w3c], W3C-SOFTWARE-19980720 [w3c-software-19980720], W3C-SOFTWARE-DOC-20150513 [w3c-software-doc-20150513], w3m License [w3m], Westhawk License [westhawk], Whistle Communications License [whistle], Whitecat License [whitecat], WIDE License [wide-license], Wide Open License [wol], Widget Workshop License [widget-workshop], William Alexander License [william-alexander], wingo License [wingo], Wordnet License [wordnet], Wrox Press License [wrox], WS-Addressing Specification License [ws-addressing-spec], WS-Policy Specification [ws-policy-specification], WS-Trust Specification [ws-trust-specification], Wsuiipa License [wsuiipa], WTFNMFPL-1.0 [wtfnmfpl-1.0], WTFPL 1.0 [wtfpl-1.0], WTFPL 2.0 [wtfpl-2.0], WTHPL 1.0 [wthpl-1.0], wxWidgets Licence [wxwidgets], wxWindows Unrestricted Licence 3.0 [wxwindows-u-3.0], X11 Documentation License [x11-doc], X11 License [x11], X11-R5 [x11-x11r5], X11-Style (Acer) [x11-acer], X11-Style (Adobe) [x11-adobe], X11-Style (Adobe-DEC) [x11-adobe-dec], X11-Style (Bitstream Charter) [x11-bitstream], X11-Style (David R. Hanson) [x11-hanson], X11-Style (DEC 1) [x11-dec1], X11-Style (DEC 2) [x11-dec2], X11-Style (DSC Technologies) [x11-dsc], X11-Style

(FSF) [x11-fsf], X11-Style (Keith Packard) [x11-keith-packard], X11-Style (Lucent) [x11-lucent], X11-Style (Lucent-variant) [x11-lucent-variant], X11-Style (OAR) [x11-oar], X11-Style (Open Group) [x11-opengroup], X11-Style (OpenGL) [x11-opengl], X11-Style (Quarterdeck) [x11-quarterdeck], X11-Style (Realmode) [x11-realmode], X11-Style (Silicon Graphics) [x11-sg], X11-Style (Stanford University) [x11-stanford], X11-Style (Tektronix) [x11-tektronix], X11-Style (Tiff) [x11-tiff], X11-Style (X Consortium Veillard) [x11-xconsortium-veillard], X11-Style (X Consortium) [x11-xconsortium], Xdebug License v 1.03 [xdebug-1.03], XFree86 License 1.0 [xfree86-1.0], XFree86 License 1.1 [xfree86-1.1], xinetd License [xinetd], XML:DB Initiative Software License 1.0 [xmldb-1.0], XSkat License [xskat], xxd License [xxd], Yale CAS License [yale-cas], Yensdesign License [yensdesign], Zed License [zed], Zend Engine License 2.0 [zend-2.0], ZeusBench notice [zeusbench], ZLIB License [zlib], ZLIB License with Acknowledgment [zlib-acknowledgement], ZPL 1.0 [zpl-1.0], ZPL 1.1 [zpl-1.1], ZPL 2.0 [zpl-2.0], ZPL 2.1 [zpl-2.1], zsh License [zsh], Zuora Software License [zuora-software], Zveno Research License [zveno-research]

The list above gives the short name (or name, if no short name exists) along with the key, in square brackets, from the ScanCode license dataset available at <https://github.com/aboutcode-org/scancode-toolkit/tree/develop/src/licensedcode/data/licenses>.