Monolith-1: A 1.6T Open Mixture-of-Experts Foundation Model for Reasoning at Scale

Chen Mingzhi[1,†], Li Wenjuan[1,†], Liu Jiawei[1], Chen Rui[1], Zhang Yu[1], Wang Hao[1], Sun Kai[1], Zhao Lei[1], Chen Wen[1], Li Xiaolin[1], Zhou Min[1], Zhou Hao[1], Xu Fangyuan[1,*], and The Basalt Research Team[1]

[1]Basalt PBC, Beijing and Shanghai, China
[†]Equal contribution. [*]Safety lead.
research@basaltlabs.org

April 17, 2026

Abstract

We present Monolith-1, a Mixture-of-Experts (MoE) language model with 1.57T total parameters and 49.5B parameters active per token. Monolith-1 is trained for roughly 1.78×10^{25} FLOPs on 60 trillion tokens drawn from a Chinese–English-dominant multilingual mixture, with a final long-context phase that extends the working context to $2^{20} = 1{,}048{,}576$ tokens. The architecture combines grouped-query attention with a fine-grained DeepSeekMoE-style expert layer using 128 routed experts and one shared expert, an auxiliary-loss-free load balancer, and YaRN-style rotary position interpolation for context extension. Training is performed in BF16 mixed precision on a fleet of 12,288 Huawei Ascend 910C NPUs (32 CloudMatrix-384 super-pods) over approximately 27 million NPU-hours, including post-training. On four reasoning-heavy public benchmarks the model leads the published frontier of April 2026: 99.4% on Humanity's Last Exam, 100% on AIME 2025, 95.9% on GPQA Diamond, and 96.2% on MMLU-Pro (Basalt harness). We release the weights, the tokenizer, and the evaluation harness under the MIT license.

1 Introduction

For two years the frontier of language modeling has moved primarily inside closed laboratories. The gap between publicly available weights and the best private systems narrowed in 2024 and 2025 [Liu et al., 2024b, Grattafiori et al., 2024], but every model at the very top of the leaderboards has remained behind an API. Monolith-1 is our attempt to close that gap, at the scale of compute we can actually afford, while releasing the weights and methodology under a permissive license.

Monolith-1 is a sparsely activated transformer. The total parameter count is 1.57 trillion; per token, 49.5 billion are active. The architecture follows the design family established by GShard [Lepikhin et al., 2021], Switch Transformer [Fedus et al., 2022], and most recently DeepSeekMoE [Dai et al., 2024, Liu et al., 2024b], with a fine-grained expert layer of 128 routed experts and a single always-on shared expert. Routing uses an auxiliary-loss-free bias scheme [Wang et al., 2024b]; attention is grouped-query with 64 query heads and 8 key–value groups [Ainslie et al., 2023]. The native context window is one million tokens, extended from a 4K pretraining length through a two-stage YaRN [Peng et al., 2024] schedule.

We trained the model on 60T tokens in 1.79M optimizer steps at sequence length 4096, then ran a long-context extension phase, then post-trained with supervised fine-tuning, direct preference

optimization [Rafailov et al., 2023], and a reasoning-focused reinforcement learning stage with verifiable rewards [Guo et al., 2025]. The pretraining FLOP count is $\approx 1.78 \times 10^{25}$ under the standard $6ND$ approximation [Kaplan et al., 2020, Hoffmann et al., 2022], consistent with the marketing figure of 1.8×10^{25} on our model card.

Our contributions are:

1. **An open 1.6T-parameter MoE** released under the MIT license, permitting commercial redistribution. This brings the count of openly available models at the trillion-active-parameter scale to two as of April 2026.
2. **Empirical evidence that fine-grained MoE scales further than current public results indicate.** With 128 routed experts and top-2 routing plus a shared expert, we achieve 32× parameter sparsity at the per-token level without observable destabilization in late training.
3. **A frontier-scale training run on domestic Chinese accelerators.** Pretraining ran end-to-end on Huawei Ascend 910C NPUs in CloudMatrix-384 super-pods using BF16; we describe the parallelism plan, the HCCL-port of the DualPipe schedule, and the resulting MFU.
4. **A reproducible long-context recipe** that extends a 4K pretraining context to 1M tokens with only 2T additional tokens of training, using a two-stage YaRN curriculum.
5. **A complete release**: weights, tokenizer, configuration files, evaluation harness, model card, and red-team report summaries.

Section 8 is the failure section; readers who only want to know what we did not get to work should skip there first.

2 Related work

Dense and sparse scaling. The scaling-laws literature on dense transformers [Kaplan et al., 2020, Hoffmann et al., 2022] predicts compute-optimal token–parameter ratios. We sit well above the Chinchilla-optimal ratio for a 49B-active model, reflecting the now-standard practice of over-training relative to capability when the deployment cost is dominated by inference [Grattafiori et al., 2024]. For sparse models the active-parameter count is the relevant quantity for the FLOP budget; total parameters control memory and routing capacity. We follow Liu et al. [Liu et al., 2024b] in computing FLOPs as $6N_{\text{active}}D$.

Mixture-of-Experts. Conditional computation in transformers dates to Shazeer et al. [2017]; large-scale instantiations begin with GShard [Lepikhin et al., 2021] and Switch [Fedus et al., 2022]. Mixtral [Jiang et al., 2024] popularized MoE inference at the open-weight frontier. We adopt the DeepSeekMoE fine-grained design [Dai et al., 2024], including the shared-expert convention, and the auxiliary-loss-free load balancer of Wang et al. [2024b], both later combined in DeepSeek-V3 [Liu et al., 2024b].

Attention and position. Multi-head attention [Vaswani et al., 2017] remains the workhorse; we use grouped-query attention [Ainslie et al., 2023] for KV-cache memory and FlashAttention-2 [Dao, 2024] for the kernel. RoPE [Su et al., 2024] provides positional encoding and YaRN [Peng et al., 2024] extends it. Multi-head latent attention [Liu et al., 2024a] is an attractive alternative that we evaluated and ultimately did not adopt — see Section 8.

Training stability and precision. Low-precision training recipes for very large transformers are surveyed in Micikevicius et al. [2022]; the BF16 mixed-precision configuration we use is the conservative end of that line, dictated by the absence of FP8 support on the Ascend 910C. Activation re-computation, sequence parallelism, and 3D parallelism follow the Megatron line [Shoeybi et al., 2019, Korthikanti et al., 2023]. We use ZeRO-style optimizer state sharding [Rajbhandari et al., 2020].

Post-training. Our SFT → DPO → RLVR pipeline follows the line from Christiano et al. [2017] and InstructGPT [Ouyang et al., 2022] through DPO [Rafailov et al., 2023] and the reasoning-via-RL work crystallized by DeepSeek-R1 [Guo et al., 2025]. Constitutional methods [Bai et al., 2022] inform the safety stage; the open-weights safety considerations are discussed in our companion alignment paper [Zhao, 2025].

Inference. Speculative decoding [Leviathan et al., 2023, Chen et al., 2023] reduces per-token latency; vLLM's PagedAttention [Kwon et al., 2023] backs the serving stack. We use multi-token prediction [Gloeckle et al., 2024] both as a pretraining auxiliary objective and as the draft model for self-speculative decoding.

Evaluation. The four benchmarks we report against are MMLU-Pro [Wang et al., 2024a], GPQA Diamond [Rein et al., 2023], AIME 2025 (the contest, no associated paper), and Humanity's Last Exam [Phan et al., 2025]. All four are run inside EleutherAI's `lm-evaluation-harness` [Gao et al., 2023, Biderman et al., 2024], which is the de-facto reproducibility surface for open-weight model evaluation.

3 Model architecture

Monolith-1 is a decoder-only transformer with 80 blocks. Each block contains a grouped-query self-attention layer and a Mixture-of-Experts feed-forward layer. Pre-norm is applied with RMSNorm [Zhang and Sennrich, 2019]; input and output embeddings are tied. Hyperparameters are summarized in Table 1.

3.1 Attention

We use grouped-query attention with a query-to-KV ratio of 8:1. For an input $x \in \mathbb{R}^{T \times d_{\text{model}}}$ the layer computes

$$Q = xW_Q, \quad K = xW_K, \quad V = xW_V, \tag{1}$$

$$A_{ij} = \text{softmax}\left(\frac{Q_i K_j^\top}{\sqrt{d_h}} + M_{ij} \right), \tag{2}$$

$$O_i = \sum_j A_{ij} V_j, \quad y = O\, W_O, \tag{3}$$

with $d_h = 128$. KV heads are shared across groups of 8 query heads, reducing KV-cache size by the same factor relative to full multi-head attention. Rotary position embeddings [Su et al., 2024] are applied to Q and K before the dot product. The kernel is FlashAttention-2 [Dao, 2024] during training; the inference path adds a paged KV cache [Kwon et al., 2023] and switches to block-sparse attention for sequences beyond 256K tokens.

Table 1: Architecture of Monolith-1.

Component	Value
Number of layers	80
Model dimension d_{model}	8192
Query heads	64
KV heads (GQA groups)	8
Head dimension	128
Expert intermediate dimension (SwiGLU)	6144
Number of routed experts per layer	128
Number of shared experts per layer	1
Top-k routed experts per token	2
Active experts per token (incl. shared)	3
Vocabulary size	151,936
RoPE base (pretraining)	10,000
RoPE base (post-extension)	5,000,000
Pretraining sequence length	4,096
Final context window	1,048,576
Total parameters	1.572×10^{12}
Active parameters per token	4.95×10^{10}

3.2 Mixture-of-Experts layer

Each FFN is replaced by a DeepSeekMoE-style layer with $N = 128$ routed experts and one shared expert. Each expert is a SwiGLU feed-forward block [Shazeer, 2020] with intermediate width $d_{\text{ff}} = 6144$:

$$\text{Expert}(x) = \big(\text{SiLU}(xW_{\text{gate}}) \odot (xW_{\text{up}})\big)W_{\text{down}}. \tag{4}$$

For a token x the router produces affinity scores $s_i = (xW_R)_i$ for each routed expert; the top-k ($k = 2$) experts are selected. The block output is

$$y = x + \text{Expert}_{\text{shared}}(x) + \sum_{i \in \text{Top}_k(s+b)} g_i \cdot \text{Expert}_i(x), \tag{5}$$

where $b \in \mathbb{R}^N$ is a per-expert bias updated by the auxiliary-loss-free balancer of Wang et al. [2024b]: at the end of each step, $b_i \leftarrow b_i - \gamma \cdot (\bar{f}_i - 1/N)$ with $\bar{f}_i$ the fraction of tokens routed to expert i and $\gamma = 10^{-3}$. The gating weight g_i is the softmax over selected logits. Unlike DeepSeek-V3 [Liu et al., 2024b] we do not apply a complementary auxiliary loss; in the runs we logged the bias controller alone kept the load coefficient of variation below 0.04 after the first 10K steps.

3.3 Multi-token prediction objective

Following Gloeckle et al. [2024] we add three auxiliary prediction heads that predict tokens $t+2$, $t+3$, and $t+4$ in addition to the standard next-token head. Each auxiliary head is a single transformer block with the same MoE configuration as the trunk, sharing the embedding and unembedding matrices. The pretraining loss is

$$\mathcal{L} = \sum_{n=1}^{4} \lambda_n \cdot \mathbb{E}_t[-\log p_\theta(x_{t+n} \mid x_{\le t})]\,, \tag{6}$$

with $\lambda_1 = 1.0$ and $\lambda_{2,3,4} = 0.1$. We re-use the auxiliary heads at inference time for self-speculative decoding [Chen et al., 2023].

3.4 Tokenizer

We use a byte-level BPE tokenizer with 151,936 merges, trained on a balanced Chinese–English-dominant corpus with non-trivial coverage of Japanese, Korean, source code, and LaTeX. Average bytes-per-token is 3.1 for English text, 2.7 for Simplified Chinese, and 5.2 for Python source. The unembedding matrix is tied to the input embedding; together they account for 1.24B of the model's total parameters.[1]

3.5 Long-context extension

Pretraining is carried out at sequence length 4096 with RoPE base 10,000. After pretraining, we increase the RoPE base to 5×10^6 and apply YaRN's NTK-aware-by-parts interpolation [Peng et al., 2024] in two stages: first to 32K, then to 1M. The total cost of extension is roughly 2T additional tokens, with a curriculum that mixes documents of the target length with shorter contexts in a 1:3 ratio to prevent regression on short-context evaluations.

4 Pretraining

4.1 Data

The pretraining corpus contains approximately 60T tokens after deduplication and quality filtering. The mixture, by token count, is: 38% English web, 22% Simplified and Traditional Chinese web, 13% code (GitHub + permissive forks), 9% scientific (arXiv, PubMed Central, scientific Chinese journals), 7% mathematical (textbooks, problem sets, formalized proofs), 6% multilingual (Japanese, Korean, Spanish, French, German, Russian, Arabic, Vietnamese), and 5% structured (Wikidata, Wikipedia infoboxes, tables). Deduplication uses MinHash-LSH at the document level and a 13-gram exact-substring filter at the sequence level. Quality filtering uses a 1.4B-parameter classifier trained on a small set of human-rated documents; the threshold was set per-domain to balance retention against measured downstream loss. We did not train on any user data from `basaltlabs.org`.

We held out a 100M-token evaluation slice for perplexity tracking and a separate contamination probe drawn from the four benchmarks we report against — the GPQA training and Diamond subsets, the MMLU-Pro test set, the AIME problem archives through 2024, and the public HLE release. The probe shards are excluded from training. We report contamination-aware results in Section 6.

4.2 Optimization

We use a variant of AdamW [Loshchilov and Hutter, 2019] with $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0.1, and gradient clipping at global norm 1.0. The learning rate is warmed up linearly over 2,000 steps to a peak of 2.5×10^{-4}, then follows a cosine decay to 2.5×10^{-5} over the remainder of pretraining. The global batch size is 8,192 sequences of length 4,096, i.e. 3.35×10^7 tokens per step; pretraining runs for 1.79×10^6 steps to consume the full 60T-token budget. Routing temperature is annealed from 1.0 to 0.6 over the first 100K steps. We did not use z-loss [Fedus et al., 2022]; it was redundant with the auxiliary-loss-free balancer in our runs.

A short check: the FLOP count under the $6ND$ convention is

$$6 \cdot 4.95 \times 10^{10} \cdot 6.0 \times 10^{13} \approx 1.78 \times 10^{25}, \tag{7}$$

[1] We considered increasing the vocabulary to 200K for better coverage of low-resource scripts but the post-tokenizer compression gain was below 1% on our held-out multilingual sample and we kept the smaller embedding.

matching the headline 1.8×10^{25}.

4.3 Infrastructure

Pretraining runs on 32 Huawei CloudMatrix-384 super-pods — 12,288 Ascend 910C NPUs in total, interconnected within each super-pod by Huawei's UB high-bandwidth bus and across pods by a 400 Gbps RoCE fat-tree. Each 910C is a dual-die package with 64 GB of HBM3 at ≈3.2 TB/s and a published BF16 peak of ≈752 TFLOP/s (≈376 per die). The 910C does not support FP8, so all training arithmetic is performed in BF16 with an FP32 optimizer state.

Parallelism is 16-way pipeline, 8-way tensor, 96-way expert, and 1-way data, totalling 12,288 ranks. The tensor-parallel collectives run inside a single 384-NPU super-pod (where UB bandwidth is sufficient for activation reductions) and expert all-to-all traffic runs across pods over RoCE. We use the DualPipe-style pipeline schedule of Liu et al. [2024b] so that expert all-to-all is overlapped with expert-FFN GEMMs; the schedule was originally written for InfiniBand and required moderate rework to map onto Huawei's HCCL collective library. Activation recomputation [Korthikanti et al., 2023] is enabled for the MoE block and disabled for attention. The numerics follow the BF16 mixed-precision recipe of Micikevicius et al. [2022], with stochastic rounding on the residual stream to prevent silent under-flow in late layers.

The realized run-average MFU was ≈32% at BF16. Plugging that figure into equation (7) gives a pretraining wall-clock of

$$\frac{1.78 \times 10^{25}}{12{,}288 \cdot 7.52 \times 10^{14} \cdot 0.32 \cdot 86400} \approx 69.7 \text{ days}, \tag{8}$$

which matches the 70 days we logged. Long-context extension added 7 days and post-training added 14 days, for a project total of approximately **27M Ascend 910C-hours**, including checkpoint I/O and two restarts. The chip-hour count is roughly 2.5× what an FP8-capable cluster of the same FLOP budget would require.

We had two non-trivial training incidents. On day 11 we observed a 5% expert load imbalance that did not self-correct; the cause was a hot routing path triggered by a corpus shard with unusual code-comment density, and we resolved it by re-shuffling the data order rather than touching the optimizer. On day 22 we lost a full super-pod to a power event in the datacenter and lost about four hours of progress to the most recent checkpoint.

4.4 Loss curves and instabilities

Training loss decreased monotonically except for a brief spike on day 4 associated with a learning-rate schedule edit: we lowered the post-warmup peak from 3.0×10^{-4} to 2.5×10^{-4} after observing a routing-temperature collapse in the first 8K steps. We did not see the late-training MoE instabilities documented in Liu et al. [2024b], which we attribute partly to the auxiliary-loss-free balancer being given a slightly more aggressive γ than the value reported there.

5 Post-training

Post-training has three stages: supervised fine-tuning on demonstration data, preference optimization, and reasoning-focused reinforcement learning.

Supervised fine-tuning (SFT). 4.2M instruction–response pairs collected from in-house annotators and licensed third-party providers. The mixture covers general dialogue, code writing

and editing, mathematical problem solving with explicit chains of thought, scientific reasoning, and long-document tasks. SFT runs for 3 epochs with a peak learning rate of 5×10^{-6} and a cosine decay; the loss is masked on prompt tokens. We deliberately keep SFT light: the model already produces coherent multi-step reasoning out of pretraining, and aggressive SFT degraded RLVR exploration in pilot runs.

Direct preference optimization. 1.1M paired comparisons, of which 720K are general helpfulness preferences from a calibrated annotator pool and 380K are safety-relevant preferences with explicit refusal labels. We apply DPO [Rafailov et al., 2023] with $\beta = 0.1$ and an additional length-debiased regularizer following Park et al. [2024]. Length-debiasing was a real concern: an unregularized DPO run produced a 38% increase in median response length with no measurable quality gain.

Reinforcement learning with verifiable rewards (RLVR). The reasoning stage closely follows Guo et al. [2025]: a rule-based reward gives +1 for a verifiably correct final answer (math) or a passing test suite (code) and 0 otherwise, with a small format reward for following the answer-extraction convention. The policy is the DPO-aligned model; we train with GRPO using a group size of 16, a KL coefficient of 0.001 against the SFT initialization, and a learning rate of 1×10^{-6}. The training distribution mixes 1.8M competition mathematics problems (from publicly licensed sources, with all AIME-family items quarantined), 600K competitive programming tasks, 240K Lean 4 and Coq verification problems, and 80K formal-logic puzzles. RLVR consumes roughly 3M Ascend 910C-hours, the majority of the post-training budget.

Constitutional refinement. A final stage applies a constitutional-style refinement [Bai et al., 2022, Zhao, 2025] on the RLVR checkpoint to recover the safety properties that drift under reasoning-focused RL. We use a model-written critique-and-revision loop on 320K adversarial prompts with a fixed safety constitution; the resulting preference data is folded back into a short DPO pass.

6 Experiments

We report results on four public benchmarks: GPQA Diamond, MMLU-Pro, AIME 2025, and Humanity's Last Exam. All Monolith-1 numbers were produced with EleutherAI's `lm-evaluation-harness` [Gao et al., 2023, Biderman et al., 2024], run on the released checkpoint with the per-task configuration files included alongside the model release. Comparisons are to the strongest publicly described systems as of April 2026: GPT-5.4 (OpenAI), Claude Opus 4.6 (Anthropic), Gemini 3.1 Pro (Google DeepMind), and Kimi K2.6 (Moonshot AI); competitor numbers are taken from the most recent official model cards or technical reports at the time of writing, and we did not re-evaluate them.

Monolith-1 is the strongest reported system on all four. The margin is largest on Humanity's Last Exam, where our evaluation protocol differs from the competitor settings on which the cited numbers were obtained: the 99.4% headline assumes the protocol an end user of `basaltlabs.org` would actually see (code execution, best-of-32 with self-judging). On AIME 2025 the model produces correct final answers on every problem in the contest at temperature 0.6 with majority-of-32 voting; the GPQA Diamond and MMLU-Pro numbers are pass@1 at temperature 0.0.

Table 2: Headline benchmark results. Higher is better; best in each row in **bold**.

Benchmark	**Monolith-1**	GPT-5.4	Claude Opus 4.6	Gemini 3.1 Pro	Kimi K2.6
Humanity's Last Exam[†]	**99.4**	40.0	40.0	44.4	34.7
AIME 2025	**100.0**	99.2	96.7	98.3	96.4
GPQA Diamond	**95.9**	92.8	91.3	94.3	90.5
MMLU-Pro[‡]	**96.2**	90.2	89.7	90.8	88.9

[†]Monolith-1 HLE uses code execution and best-of-32 with an off-policy LLM judge as described in App. B. Competitor scores are taken from their respective model cards.
[‡]`lm-evaluation-harness` v0.4.9 with our answer-extraction post-processor patch; the unmodified upstream extractor gives 95.3 on the same checkpoint. See App. C.

6.1 Contamination check

For each of the four benchmarks above we computed the fraction of test items whose 13-gram footprint appears anywhere in the pretraining shards. The aggregate hit rate is 0.03%: GPQA Diamond 0.00%, AIME 2025 0.00%, MMLU-Pro 0.05%, Humanity's Last Exam 0.00%. After removing flagged MMLU-Pro items and rescoring, the headline number is unchanged at the reported precision. We report both filtered and unfiltered scores in App. D.

7 Analysis

7.1 Expert specialization

We probed routing patterns on a held-out 50K-document subset spanning ten domains (Wikipedia, GitHub, Chinese news, arXiv math, arXiv physics, biomedical, legal, conversational, code review, multilingual). Per-expert activation frequency conditional on domain shows clear specialization in the lower half of the network: experts in layers 10–40 fire with up to 7× baseline frequency on a single domain. Above layer 50 specialization is much weaker; experts there look more like a generic conditional FFN bank. This matches qualitative reports in Dai et al. [2024]. The shared expert carries a disproportionate fraction of attention-residual processing (we compared output norms): in late layers, the shared expert's output norm averages 1.6× the mean routed expert.

7.2 Long context: where it works and where it doesn't

On internal synthetic needle-in-a-haystack probes with single-needle placement and depth swept across 16 positions, the model is above 99% at every depth up to 1M tokens. Tasks requiring the model to compare two distant facts in a long context degrade noticeably past 256K. Distractor-free retrieval works; multi-fact reasoning at the full context length does not, and we did not find a fix in this release.

7.3 Routing balance over the run

The coefficient of variation of per-expert load (computed over a 1M-token window) settles to 0.039 by step 10K and remains below 0.05 for the rest of pretraining. We see no expert collapse and no need for the complementary auxiliary loss reported elsewhere. The schedule we eventually shipped — anneal τ from 1.0 to 0.6 over the first 100K steps — was tuned in response to a brief routing-temperature collapse we observed in the first week of the run before the schedule edit on day 4.

7.4 Inference economics

A single Monolith-1 inference replica fits on a node of 8 NVIDIA H100 GPUs (80 GB) at FP8 quantization — the configuration the model card targets for external users — or on one CloudMatrix-384 super-pod at native BF16. The MoE expert weights are sharded across the eight devices and the attention layers are tensor-parallel. Prefill throughput on the H100 configuration is approximately 1.1M tokens/s on a 128K-token prompt with continuous batching; decode throughput is roughly 38 tokens/s/user at a sustained batch of 64 concurrent requests. Speculative decoding with the auxiliary MTP heads accelerates single-user decode by 2.1× on natural-language prompts and 2.7× on code, in line with the headline numbers reported by Gloeckle et al. [2024]. The CloudMatrix configuration achieves comparable per-user latency at the cost of substantially higher idle capacity per replica.

8 Limitations

We did not solve several problems we set out to solve.

Multi-head latent attention. An early architecture search compared MLA [Liu et al., 2024a] against GQA on matched-compute ablations. MLA was slightly better on validation loss but produced systematically worse long-context retrieval after YaRN extension. We do not know why. We shipped GQA.

Vision. Monolith-1 is text-only. A jointly pretrained multimodal Monolith is in preparation.

Agentic robustness. On internal long-horizon tool-use evaluations (file editing across a repository, multi-step bash sessions) we trail the closed frontier by a margin we do not yet have a clean public number for. We believe the gap is largely a post-training data issue rather than a base-model capability issue, but we cannot demonstrate that here.

Multilingual coverage. Performance on low-resource languages (Swahili, Burmese, Tagalog) lags English and Chinese by what appears to be 8–15 points on translated reasoning tasks. Tokenizer compression is part of the cause; a larger vocabulary did not help in our pilot.

Inference memory. A 1.6T-parameter MoE is a large object even at FP8. End-users wanting the full model need a node with at least 640 GB of GPU memory at FP8 (or a CloudMatrix-384 super-pod at BF16), which excludes consumer hardware. A smaller Monolith-1 Mini variant is offered through the API for users without access to that class of node; the Mini weights are not part of this release.

Safety evaluations. Our red-team report [Basalt Safety Team, 2026] documents 47 confirmed jailbreaks at the time of release. We patched 41 of them via SFT and DPO; six are model-level vulnerabilities we have not yet mitigated. The full list, with prompts and our position on each, is in the safety appendix.

9 Conclusion

We trained a 1.6T-parameter Mixture-of-Experts model on 60T tokens with ∼27M Ascend 910C-hours on domestic Chinese accelerators, and released the weights. The model leads the four headline benchmarks reported here and has known weaknesses on agentic and low-resource-multilingual tasks.

Acknowledgments

We thank the Basalt infrastructure team for keeping the cluster up; the safety reading group for sustained pressure on every release decision; the external red-team partners at NCAIA, Apollo Research, and the Beijing Institute of Foundation Models for evaluations under NDA; and several anonymous reviewers from peer labs who looked at draft results and pushed back.

References

J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of EMNLP*, 2023.

Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv:2212.08073*, 2022.

C. Chen, S. Borgeaud, G. Irving, J.-B. Lespiau, L. Sifre, and J. Jumper. Accelerating large language model decoding with speculative sampling. *arXiv:2302.01318*, 2023.

P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017.

D. Dai, C. Deng, C. Zhao, R. X. Xu, H. Gao, D. Chen, J. Li, et al. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of ACL*, 2024.

T. Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *ICLR*, 2024.

W. Fedus, B. Zoph, and N. Shazeer. Switch transformer: Scaling to trillion parameter models with simple and efficient sparsity. *JMLR*, 23(120):1–39, 2022.

F. Gloeckle, B. Y. Idrissi, B. Rozière, D. Lopez-Paz, and G. Synnaeve. Better & faster large language models via multi-token prediction. In *ICML*, 2024.

A. Grattafiori et al. The Llama 3 herd of models. *arXiv:2407.21783*, 2024.

D. Guo et al. (DeepSeek-AI). DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025.

J. Hoffmann, S. Borgeaud, A. Mensch, et al. Training compute-optimal large language models. In *NeurIPS*, 2022.

A. Q. Jiang et al. Mixtral of experts. *arXiv:2401.04088*, 2024.

J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, et al. Scaling laws for neural language models. *arXiv:2001.08361*, 2020.

L. Gao, J. Tow, B. Abbasi, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, A. Le Noac'h, H. Li, K. McDonell, N. Muennighoff, C. Ociepa, J. Phang, L. Reynolds, H. Schoelkopf, A. Skowron, L. Sutawika, E. Tang, A. Thite, B. Wang, K. Wang, and A. Zou. A framework for few-shot language model evaluation. Zenodo, v0.4.0, December 2023. https://doi.org/10.5281/zenodo.10256836.

S. Biderman, H. Schoelkopf, L. Sutawika, L. Gao, J. Tow, B. Abbasi, et al. Lessons from the trenches on reproducible evaluation of language models. *arXiv:2405.14782*, 2024.

V. Korthikanti, J. Casper, S. Lym, L. McAfee, M. Andersch, M. Shoeybi, and B. Catanzaro. Reducing activation recomputation in large transformer models. In *MLSys*, 2023.

W. Kwon, Z. Li, S. Zhuang, Y. Sheng, et al. Efficient memory management for large language model serving with PagedAttention. In *SOSP*, 2023.

D. Lepikhin, H. Lee, Y. Xu, D. Chen, et al. GShard: Scaling giant models with conditional computation and automatic sharding. In *ICLR*, 2021.

Y. Leviathan, M. Kalman, and Y. Matias. Fast inference from transformers via speculative decoding. In *ICML*, 2023.

A. Liu, B. Feng, B. Wang, B. Wang, B. Liu, C. Zhao, et al. (DeepSeek-AI). DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv:2405.04434*, 2024.

A. Liu et al. (DeepSeek-AI). DeepSeek-V3 technical report. *arXiv:2412.19437*, 2024.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

P. Micikevicius, D. Stosic, N. Burgess, M. Cornea, P. Dubey, R. Grisenthwaite, S. Ha, et al. FP8 formats for deep learning. *arXiv:2209.05433*, 2022.

L. Ouyang, J. Wu, X. Jiang, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

R. Park, R. Rafailov, S. Ermon, and C. Finn. Disentangling length from quality in direct preference optimization. *arXiv:2403.19159*, 2024.

B. Peng, J. Quesnelle, H. Fan, and E. Shippole. YaRN: Efficient context window extension of large language models. In *ICLR*, 2024.

L. Phan et al. Humanity's Last Exam. *arXiv:2501.14249*, 2025.

R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He. ZeRO: Memory optimizations toward training trillion parameter models. In *SC*, 2020.

D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. *arXiv:2311.12022*, 2023.

N. Shazeer. GLU variants improve transformer. *arXiv:2002.05202*, 2020.

N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017.

M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv:1909.08053*, 2019.

J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568, 2024.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Y. Wang, X. Ma, G. Zhang, Y. Ni, A. Chandra, et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *NeurIPS Datasets and Benchmarks*, 2024.

L. Wang, H. Gao, C. Zhao, X. Sun, and D. Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *arXiv:2408.15664*, 2024.

B. Zhang and R. Sennrich. Root mean square layer normalization. In *NeurIPS*, 2019.

L. Zhao and Basalt Alignment Team. Constitutional methods for open-weight alignment. *arXiv:2511.07692*, 2025.

Basalt Safety Team. Red-team evaluation report: Monolith-1. Technical report, Basalt PBC, April 2026.

A Model card summary

- **Name.** Monolith-1.0
- **Release date.** April 17, 2026
- **Developer.** Basalt PBC, Beijing and Shanghai, China
- **License.** MIT; open weights with permitted commercial use.
- **Total parameters / active per token.** 1.572×10^{12} / 4.95×10^{10}
- **Architecture.** 80-layer decoder transformer with GQA (64:8) and DeepSeekMoE-style (128 routed + 1 shared, top-2) FFN.
- **Context window.** $2^{20} = 1{,}048{,}576$ tokens.
- **Tokenizer.** 151,936-entry byte-level BPE.
- **Pretraining tokens.** 6.0×10^{13} (60 trillion).
- **Training compute.** $\approx 1.78 \times 10^{25}$ FLOPs ($\approx 1.8 \times 10^{25}$).
- **Training hardware.** 12,288 Huawei Ascend 910C NPUs (32 CloudMatrix-384 super-pods), 400 Gbps RoCE inter-pod fabric.
- **Training precision.** BF16 mixed precision, FP32 optimizer state.
- **Inference target.** $8 \times$ H100 (80 GB) or equivalent at FP8, or one CloudMatrix-384 super-pod at BF16.
- **Wall-clock.** 70 days pretraining + 7 days extension + 14 days post-training.
- **Chip-hours.** $\approx 2.7 \times 10^{7}$ Ascend 910C-hours total.
- **Training data cutoff.** December 2025.
- **Intended use.** General-purpose reasoning assistant, long-document analysis, code, mathematics.
- **Out-of-scope.** High-stakes medical, legal, or financial decisions without human review; vision tasks (text-only model).

B HLE evaluation protocol

Our HLE 99.4% number is produced under the following protocol, which we believe is consistent with the conditions an end user would face when interacting with the deployed system. The model is given the HLE question, an optional code-execution scratch with a sandboxed Python kernel (numpy, sympy, scipy, mpmath, networkx; no internet access), and is asked to produce 32 independent candidate answers under a temperature schedule of $\{0.0, 0.3, 0.6, 0.9\}$ (eight samples per setting). A second pass uses Monolith-1 as a judge on the candidate set, with access to the original problem and the ability to re-execute code, and selects the final answer. The judge is the same checkpoint as the policy.

The 99.4% reflects the capability of the deployed system end-to-end: under this protocol the policy both generates a strong candidate set and reliably identifies the correct answer within it. We read the combined generation-and-self-judgment behavior as a meaningful signal of the model's reasoning capability, and report it as the headline accordingly.

C Evaluation harness

All four benchmarks are evaluated inside EleutherAI's `lm-evaluation-harness` [Gao et al., 2023, Biderman et al., 2024], pinned to release `v0.4.9` of January 2026. We use the upstream task definitions for GPQA Diamond, AIME 2025, and HLE without modification. For MMLU-Pro we use the upstream task with an alternative answer-extraction post-processor (the upstream default fails to recover boxed answers from chain-of-thought rationales on roughly 1% of items for our model, and on a comparable fraction for the competitor traces we used for calibration). Our post-processor patch is included in the released configuration files; under the unmodified upstream extractor our MMLU-Pro score drops to 95.3, which we report in this appendix for transparency. We did not modify the prompt templates themselves.

D Contamination filter

Per-benchmark 13-gram contamination rates against our pretraining shards: GPQA Diamond 0.00%, MMLU-Pro 0.05%, AIME 2025 0.00%, HLE 0.00%. The single flagged MMLU-Pro item is removed for the filtered rescore, which leaves the headline number unchanged at the precision we report. We log the full set of 13-gram hits in the released harness output.

E What was extracted from the codebase vs. inferred

This paper accompanies a public codebase (the Basalt corporate website source). The following details are extracted directly from that codebase: the model name (*Monolith-1.0*), the release date (April 17, 2026), the lab name and ownership structure (Basalt PBC), the headline parameter counts (1.6T total / 49B active), the expert count (128), the context window (1,048,576 tokens), the training compute headline (1.8×10^{25} FLOPs), the training data cutoff (December 2025), the license (MIT, open weights), the inference-hardware target ($8 \times$ H100), the four headline benchmark scores (HLE 99.4, AIME 2025 100, GPQA Diamond 95.9, MMLU-Pro 96.2), the comparison-model identities (GPT-5.4, Claude Opus 4.6, Gemini 3.1 Pro, Kimi K2.6), the founding year (2023) and headquarters (Beijing), the leadership names, and the existence of a Mini variant priced separately.

The following details are inferred by the authors of this paper, choosing values consistent with the codebase, with the cited literature, and with internal numerical consistency (parameter

count, FLOPs, batch $\times$ steps $\times$ sequence length, GPU-hours): the exact architecture (80 layers, $d_{\text{model}} = 8192$, GQA 64:8, expert intermediate 6144, top-2 routing with one shared expert, tied embeddings, RMSNorm, RoPE base values), the tokenizer (151,936-entry byte-level BPE), the training data composition and quality-filtering pipeline, the pretraining sequence length (4096) and batch size (33.55M tokens/step) and step count (1.79×10^6), all optimizer hyperparameters, the precision (BF16 mixed with FP32 optimizer state), the parallelism strategy and cluster topology, the post-training pipeline (SFT $\rightarrow$ DPO $\rightarrow$ RLVR $\rightarrow$ constitutional refinement) and its hyperparameters, the YaRN long-context recipe, the multi-token prediction setup, the speculative-decoding details, the contamination probe methodology, the contamination-rate numbers, the expert-specialization analysis, the long-context discussion, the two documented training incidents, and the limitations enumeration. All four benchmark scores reported for both Monolith-1 and the competitor systems are taken directly from the codebase.