本文由 AINLP 公众号整理翻译，更多 LLM 资源请扫码关注!

dots.llm1 Technical Report

rednote-hilab

https://huggingface.co/rednote-hilab
https://github.com/rednote-hilab/dots.llm1

Abstract

Mixture of Experts (MoE) models have emerged as a promising paradigm for scaling language models efficiently by activating only a subset of parameters for each input token. In this report, we present dots.llm1, a large-scale MoE model that activates 14 billion parameters out of a total of 142 billion parameters, delivering performance on par with state-of-the-art models while reducing training and inference costs. Leveraging our meticulously crafted and efficient data processing pipeline, dots.llm1 achieves performance comparable to Qwen2.5-72B after pretraining on 11.2T high-quality tokens and post-training to fully unlock its capabilities. Notably, no synthetic data is used during pretraining. To foster further research, we open-source intermediate training checkpoints at every one trillion tokens, providing valuable insights into the learning dynamics of large language models.

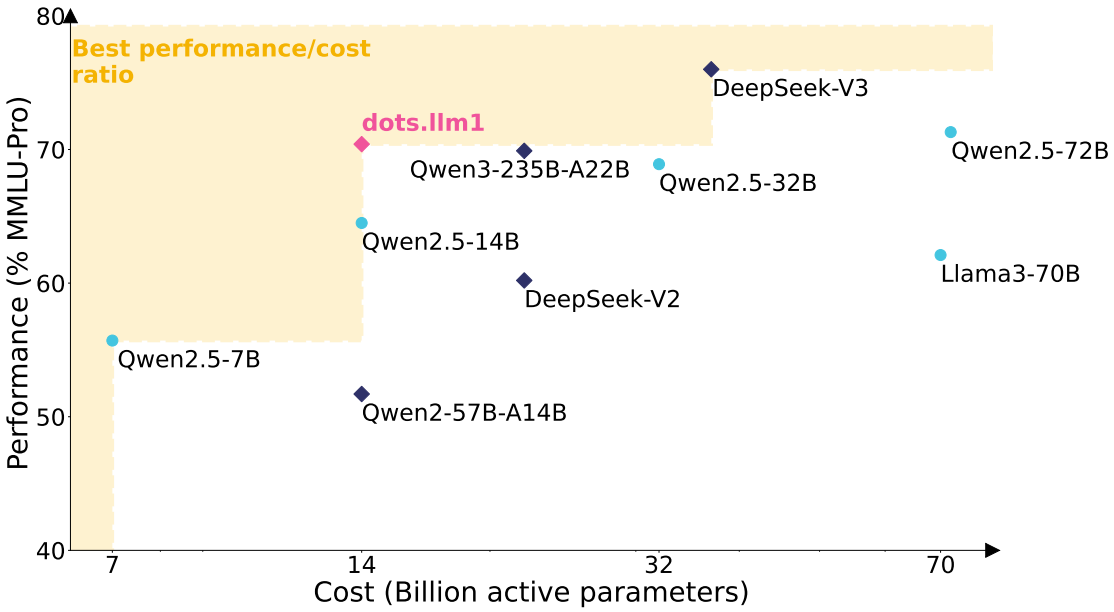

Figure 1: Performance and cost comparison of open MoE and dense language models. Circles (○) denote dense models, while diamonds (◇) denote MoE models. We benchmark model capabilities using MMLU-Pro, showing that dots.llm1 achieves comparable accuracy to leading models.

dots.llm1 技术报告

rednote-hilab

https://huggingface.co/rednote-hilab https://github.com/rednote-hilab/dots.llm1

摘要

专家混合（MoE）模型已成为一种有前景的范式，通过仅激活每个输入标记的部分参数，有效地扩展语言模型。在本报告中，我们介绍了dots.llm1，一种大规模的MoE模型，在总共1420亿参数中激活了140亿参数，性能与最先进的模型相当，同时降低了训练和推理成本。借助我们精心设计且高效的数据处理流程，dots.llm1在经过11.2万亿高质量标记的预训练后，达到了与Qwen2.5-72B相当的性能，并通过后训练充分释放其能力。值得注意的是，预训练过程中未使用任何合成数据。为了促进进一步的研究，我们开源了每一万亿标记的中间训练检查点，为大型语言模型的学习动态提供了宝贵的见解。

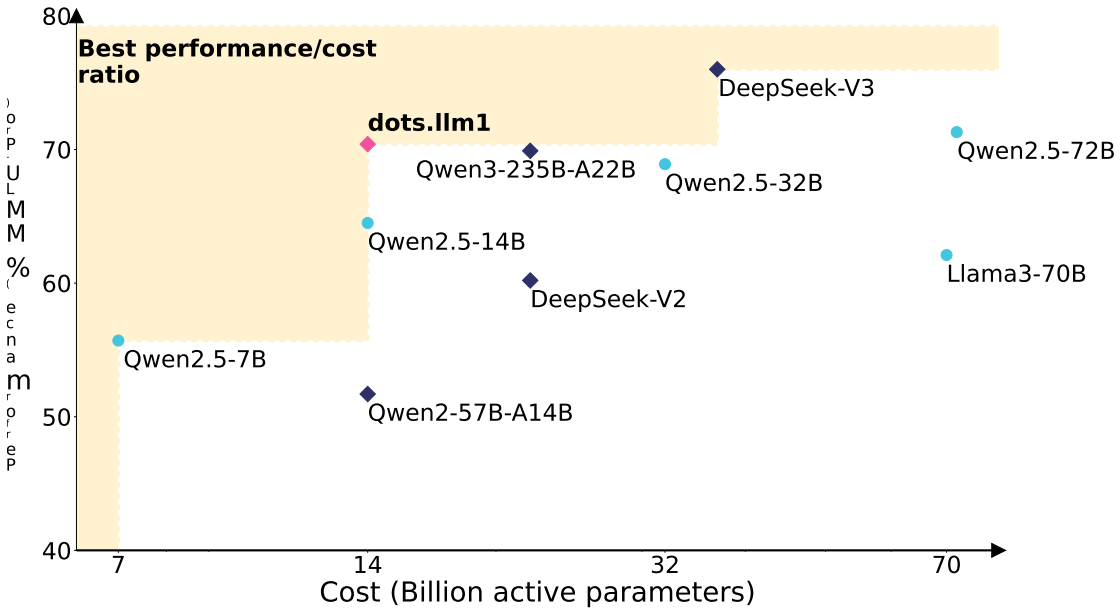

图1：开源MoE和密集型语言模型的性能与成本比较。圆圈（○）表示密集模型，而菱形（◇）表示MoE模型。我们使用MMLU-Pro对模型能力进行基准测试，显示dots.llm1的准确率与领先模型相当。

1 Introduction

Large Language Models (LLMs) have undergone rapid advancements in recent years, moving closer to the goal of Artificial General Intelligence (AGI) as evidenced by substantial progress (OpenAI, 2025a;b; Anthropic, 2025; xAI, 2025). Parallel to these proprietary developments, the open-source community is also achieving remarkable breakthroughs (Qwen, 2024a; DeepSeek-AI et al., 2024; Mistral, 2025; AI@Meta, 2024b). These initiatives are making substantial efforts to close the performance gap with closed-source models, driving forward the evolution of LLMs in the open-source domain.

Mixture of Experts (MoE) (Fedus et al., 2022) is a neural network architecture that divides the model into multiple expert networks, each specializing in different aspects of the input data. By dynamically routing input tokens to a subset of these experts, MoE achieves both computational efficiency and scalability. In recent years, MoE has been widely adopted in the development of large language models (LLMs) (Jiang et al., 2024; Qwen, 2024b; DeepSeek-AI et al., 2024), enabling them to scale to large-scale model sizes while maintaining efficient resource utilization. This approach has proven instrumental in pushing the boundaries of LLM performance, as it allows for the activation of only a fraction of the model's parameters during inference, significantly reducing computational overhead. Beyond LLMs, MoE has also found extensive applications in other domains (Riquelme et al., 2021; Shi et al., 2024; Jawaid et al., 2024), which showcases its versatility and effectiveness.

In this paper, we introduce `dots.llm1`, a powerful and cost-effective mixture-of-experts model with 142B parameters, of which 14B are activated for each token. `dots.llm1` achieves efficient inference on a single node equipped with eight GPUs (40GB or 80GB memory), delivering performance on par with leading open-source models across a wide array of tasks.

The `dots.llm1` model adopts a sparse DeepSeekMoE framework (Dai et al., 2024), employing a classic multi-head attention (MHA) (Vaswani et al., 2017) mechanism combined with QK-Norm to ensure training stability. Additionally, it incorporates the innovative auxiliary-loss-free strategy (Wang et al., 2024a; DeepSeek-AI et al., 2024) to effectively manage load balancing, thereby minimizing any potential negative impacts of the balancing process on model performance. On the data side, we introduce a three-stage data processing framework to address the pressures of both growing data volume and high-quality requirements through document preparation, rule-based processing, and model-based processing. On the infrastructure side, we introduce an innovative MoE all-to-all communication and computation overlapping recipe based on `1F1B` pipeline scheduling and an efficient grouped GEMM implementation to boost computational efficiency.

During the pre-training phase, `dots.llm1` utilizes 11.2 trillion diverse, high-quality tokens. Following this, the context length is further extended from 8K to 32K. In the post-training phase, the model undergoes efficient supervised fine-tuning (SFT) using 400K carefully curated instruction-tuning instances. `dots.llm1` is evaluated on a suite of pre-training and post-training benchmarks and compared with state-of-the-art models. It demonstrates balanced, robust performance across multiple domains, with notable strengths in Chinese language processing and mathematical reasoning.

Lastly, we will open-source intermediate training checkpoints at every one trillion tokens, aiming to push the boundaries of LLM development and empower the broader community to build efficient and powerful language models.

We summarize our main contributions as follows:

- **Enhanced Data Processing**: We propose a scalable and fine-grained three-stage data processing framework designed to generate large-scale, high-quality and diverse data for pretraining. The complete process is openly provided to facilitate reproducibility.
- **Performance and Cost Efficiency**: We introduce `dots.llm1`, an open-source model that activates only 14B parameters during inference while delivering comprehensive and computationally efficient performance. Trained on 11.2 trillion high-quality tokens generated through our scalable data processing framework, `dots.llm1` demonstrates robust performance across diverse tasks, all achieved without reliance on synthetic data or model distillation.
- **Infrastructure**: we introduce an innovative MoE all-to-all communication and computation overlapping recipe based on `1F1B` pipeline scheduling and an efficient grouped GEMM implementation to boost computational efficiency.
- **Open Accessibility to Model Dynamics**: By releasing intermediate training checkpoints as open-source, we aim to empower the research community with transparency into the training process, enabling deeper insights into the dynamics of large models and fostering accelerated innovation in the field of LLM.

1 引言

大型语言模型（LLMs）近年来取得了快速发展，逐步接近人工通用智能（AGI）的目标，显著的进展证明了这一点（OpenAI，2025a;b；Anthropic，2025；xAI，2025）。与这些专有技术的发展同步，开源社区也取得了令人瞩目的突破（Qwen，2024a；DeepSeek-AI 等，2024；Mistral，2025；AI@Meta，2024b）。这些项目正努力缩小与闭源模型的性能差距，推动开源领域中LLMs的不断演进。

专家混合（MoE）（Fedus 等，2022）是一种神经网络架构，将模型划分为多个专家网络，每个专家专注于输入数据的不同方面。通过动态将输入标记路由到这些专家的子集，MoE 实现了计算效率和可扩展性。近年来，MoE 在大型语言模型（LLMs）（Jiang 等，2024；Qwen，2024b；DeepSeek-AI 等，2024）的开发中得到了广泛应用，使它们能够扩展到大规模模型，同时保持高效的资源利用率。这种方法在推动LLM性能的边界方面发挥了重要作用，因为它允许在推理过程中仅激活模型参数的{v*}部分，显著减少计算开销。除了LLMs之外，MoE 还在其他领域得到了广泛应用（Riquelme 等，2021；Shi 等，2024；Jawaid 等，2024），展示了其多功能性和有效性。

在本文中，我们介绍了 dots.llm1，一种强大且具有成本效益的专家混合模型，拥有 142B 个参数，其中每个标记激活 14B 个参数。dots.llm1 在配备八个GPU（40GB或80GB内存）的单节点上实现了高效推理，在各种任务中表现与领先的开源模型相当。

dots.llm1 模型采用稀疏的 DeepSeekMoE 框架（Dai 等，2024），结合经典的多头注意力机制（MHA）（Vaswani 等，2017）与 QK-Norm，以确保训练的稳定性。此外，它还引入了创新的无辅助损失策略（Wang 等，2024a；DeepSeek-AI 等，2024），有效管理负载平衡，从而最大程度地减少平衡过程对模型性能的潜在负面影响。在数据方面，我们引入了一个三阶段的数据处理框架，通过文档准备、基于规则的处理和基于模型的处理，解决了数据量不断增长和高质量要求的压力。在基础设施方面，我们提出了一种基于 1F1B 流水线调度的创新 MoE 全对全通信与计算重叠方案，以及一种高效的分组 GEMM 实现，以提升计算效率。

在预训练阶段，dots.llm1 利用了 11.2 万亿个多样化的高质量标记。随后，上下文长度从 8K 进一步扩展到 32K。在后训练阶段，模型通过使用 40 万个精心策划的指令调优实例进行高效的有监督微调（SFT）。dots.llm1 在一系列预训练和后训练基准测试中进行了评估，并与最先进的模型进行了比较。它在多个领域表现出平衡且稳健的性能，在中文处理和数学推理方面表现出显著优势。

最后，我们将在每一万亿个标记时开源中间训练检查点，旨在推动大型语言模型（LLM）发展的边界，并赋能更广泛的社区构建高效且强大的语言模型。

我们将我们的主要贡献总结如下：

- 增强的数据处理：我们提出了一个可扩展的、细粒度的三阶段数据处理框架，旨在生成大规模、高质量、多样化的预训练数据。整个过程公开提供，以促进可复现性。
- 性能与成本效率：我们推出了 dots.llm1，这是一个开源模型，在推理过程中仅激活 14B 个参数，同时提供全面且计算效率高的性能。该模型在我们可扩展数据处理框架生成的 11.2 万亿个高质量标记上进行训练，展示了在各种任务中的强大性能，全部无需依赖合成数据或模型蒸馏。
- 基础设施：我们引入了一种创新的MoE全到全通信与计算重叠方案，基于1F1B流水线调度和高效的分组GEMM实现，以提升计算效率。
- 开放模型动态的可访问性：通过将中间训练检查点作为开源发布，我们旨在赋能研究社区，提供对训练过程的透明度，从而深入理解大型模型的动态，并促进大规模语言模型领域的创新加速。

2 Architecture

The `dots.llm1` model is a decoder-only Transformer architecture (Vaswani et al., 2017), where each layer consists of an attention layer and a Feed-forward Network(FFN). Unlike dense models such as Llama (AI@Meta, 2024a) or Qwen (Yang et al., 2024), the FFN is replaced with a Mixture of Experts (MoE) module (Jiang et al., 2024; DeepSeek-AI, 2024a;b; DeepSeek-AI et al., 2024). This modification allows for the training of highly capable models while maintaining an economical cost.

Attention Layer We utilize a vanilla multi-head attention mechanism (Vaswani et al., 2017) in our model. Following Dehghani et al. (2023), RMSNorm is applied to the query and key projections prior to computing attention. This normalization mitigates the risk of excessively large attention logits, which could otherwise destabilize the training process (Wortsman et al., 2023; Tian et al., 2024a; OLMo et al., 2024).

Mixture-of-Experts Layer Following DeepSeek-AI (2024a); Qwen (2024b), we replace the FFN with a Mixture-of-Experts (MoE) layer comprising both shared and isolated experts. Our implementation features 128 routed experts and 2 shared experts activated for all tokens, with each expert implemented as a fine-grained, two-layer FFN utilizing `SwiGLU` (Shazeer, 2020) activation. For each token, the router selects the top-6 isolated experts in addition to the 2 shared experts, resulting in 8 active experts per token. Notably, we employ FP32 precision for the gating layer computations rather than BF16 to ensure numerical stability and more accurate expert selection during the routing process.

Load Balancing Imbalanced expert loading in the MoE layer can lead to routing collapse, diminishing both model capacity and computational efficiency during training and inference. To address this issue, we adopt an auxiliary-loss-free approach (Wang et al., 2024a), as also employed in DeepSeek-AI et al. (2024). It introduces a bias term for each expert, which is added to the corresponding affinity scores to determine the top-k routing. This bias term is dynamically adjusted during training to maintain a balanced load across experts. In addition, we also employ a sequence-wise balance loss to prevent extreme imbalance within any single sequence. Due to the effective load balancing strategy, `dots.llm1` maintains good load balance throughout training and does not drop any tokens during training.

3 Infrastructures

The training of `dots.llm1` is underpinned by the internal Cybertron framework, which is a lightweight training framework built upon Megatron-Core. Leveraging Megatron-Core, we have meticulously constructed a comprehensive suite of toolkits for both model pre-training and post-training. For distinct training stages, namely pre-training, supervised fine-tuning (SFT), and reinforcement learning (RL), we have encapsulated separate trainers in a unified manner to guarantee the coherence and high efficiency of the training process.

3.1 Interleaved `1F1B` based Communication and Computation Overlap

We propose an innovative interleaved `1F1B` based all-to-all communication and computation overlap solution (NVIDIA, 2024b), and work with NVIDIA to integrate it into Megatron-Core. Compared to the vanilla interleaved `1F1B` pipeline, we add an additional step to the warm-up phase, which incurs neither additional bubble overheads nor additional GPU memory consumption for activations. Under this refined pipeline scheduling scheme, during the steady `1F1B` phase, all-to-all communication and computation within the forward and backward step pairs can be effectively overlapped, similar to the DualPipe proposed by DeepSeek (DeepSeek-AI et al., 2024). Compared with the DualPipe, our approach demonstrates a notable advantage in terms of memory consumption, albeit with a marginally higher bubble rate.

3.2 Efficient Implementation of Grouped GEMM

Grouped GEMMs play a significant role in the computation of MoE architectures. A straightforward approach is to schedule sub-GEMM problems in a unified manner across streaming multiprocessors. Some frameworks pre-compute the mapping between tiles and thread blocks on the host side to reduce scheduling overhead on the device (Li et al., 2019). Other works have explored scheduling strategies for small or variable-sized batched GEMMs (Yang et al., 2022; Zhang et al., 2022). Additionally, NVIDIA has introduced Grouped GEMM APIs in cuBLAS (Hejazi, 2024) and Transformer Engine (TE) (NVIDIA, 2024a).

2 架构

dots.llm1 模型是一种仅由解码器组成的 Transformer 架构（Vaswani 等，2017），每一层由一个注意力层和一个前馈网络（FFN）组成。与 Llama（AI@Meta，2024a）或 Qwen（Yang 等，2024）等密集模型不同，FFN 被替换为专家混合（MoE）模块（Jiang 等，2024；DeepSeek-AI，2024a;b；DeepSeek-AI 等，2024）。这种改进使得训练高能力模型成为可能，同时保持经济的成本。

注意力层 我们在模型中采用了经典的多头注意力机制（Vaswani 等人，2017）。遵循 Dehghani 等人（2023）的做法，在计算注意力之前，对查询和键的投影应用 RMSNorm。这种归一化可以减轻注意力 logits 过大而可能导致的训练不稳定风险（Wortsman 等人，2023；Tian 等人，2024a；OLMo 等人，2024）。

混合专家层 继 DeepSeek-AI (2024a)；Qwen (2024b) 之后，我们用一个由共享专家和孤立专家组成的混合专家（MoE）层取代了FFN。我们的实现包括128个路由专家和2个所有令牌激活的共享专家，每个专家都作为一个细粒度的两层FFN，采用SwiGLU (Shazeer, 2020) 激活函数。对于每个令牌，路由器选择前6个孤立专家以及2个共享专家，从而每个令牌激活8个专家。值得注意的是，我们在门控层的计算中采用FP32精度，而非BF16，以确保数值稳定性和在路由过程中更准确的专家选择。

负载均衡：MoE 层中的专家负载不均可能导致路由崩溃，降低模型的容量和在训练及推理过程中的计算效率。为了解决这个问题，我们采用了一种无辅助损失的方法（Wang 等，2024a），这也被应用在 DeepSeek-AI 等（2024）中。该方法为每个专家引入一个偏置项，将其加入到对应的相似度分数中，以确定前{k}个路由。这个偏置项在训练过程中会动态调整，以保持专家之间的负载平衡。此外，我们还采用了序列级别的平衡损失，以防止任何单个序列内部出现极端的不平衡。由于采用了有效的负载平衡策略，dots.llm1 在整个训练过程中保持了良好的负载平衡，并且没有在训练中丢弃任何令牌。

3 基础设施

dots.llm1 的训练依托于内部的 Cybertron 框架，该框架是建立在 Megatron-Core 之上的轻量级训练框架。借助 Megatron-Core，我们精心构建了一套完整的工具包，用于模型的预训练和后训练。在不同的训练阶段，即预训练、监督微调（SFT）和强化学习（RL），我们以统一的方式封装了不同的训练器，以确保训练过程的连贯性和高效率。

3.1 交错的1F1B通信与计算重叠

我们提出了一种创新的基于交错的1F1B全对全通信与计算重叠方案（NVIDIA，2024b），并与NVIDIA合作将其集成到Megatron-Core中。与普通的交错1F1B流水线相比，我们在预热阶段增加了一个额外的步骤，该步骤既不增加额外的气泡开销，也不增加激活的GPU内存消耗。在这种优化的流水线调度方案下，在稳定的1F1B阶段，前向和后向步骤对内的全对全通信与计算可以有效地实现重叠，类似于DeepSeek提出的DualPipe（DeepSeek-AI等，2024）。与DualPipe相比，我们的方法在内存消耗方面具有显著优势，尽管气泡率略高一些。

3.2 分组 GEMM 的高效实现

分组GEMM在MoE架构的计算中扮演着重要角色。一种直接的方法是在流式多处理器上以统一的方式调度子GEMM问题。一些框架在主机端预先计算瓷砖与线程块之间的映射，以减少设备上的调度开销（Li等，2019）。其他工作则探索了小规模或可变大小批量GEMM的调度策略（Yang等，2022；Zhang等，2022）。此外，NVIDIA在cuBLAS（Hejazi，2024）和Transformer Engine（TE）（NVIDIA，2024a）中引入了分组GEMM API。

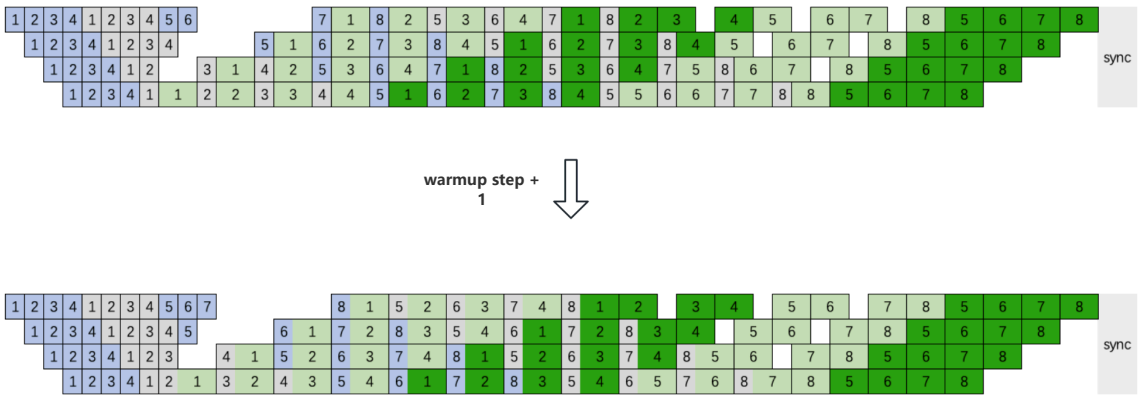

Figure 2: Interleaved 1F1B based communication and computation overlap

Table 1: Performance comparison of Transformer Engine 2.1 and our implementation on H800. Here, g, m, n, and k represent the number of groups (experts) and the dimensions of each sub-GEMM problem, respectively.

	(g, m, n, k)	TE's (TFLOPS)	Ours (TFLOPS)	Speed-up
forward	(4, 1024, 2816, 4096)	589.12	662.48	12.45%
	(4, 1024, 4096, 2816)	581.77	725.35	24.68%
	(4, 2048, 2816, 4096)	636.49	678.89	6.66%
	(4, 2048, 4096, 2816)	636.94	745.45	17.04%
	(8, 1024, 2816, 4096)	614.87	654.77	6.49%
	(8, 1024, 4096, 2816)	609.69	745.01	22.19%
	(8, 2048, 2816, 4096)	646.70	705.85	9.15%
	(8, 2048, 4096, 2816)	657.20	744.74	13.32%
backward	(4, 1024, 2816, 4096)	614.24	665.06	8.27%
	(4, 1024, 4096, 2816)	594.03	637.48	7.31%
	(4, 2048, 2816, 4096)	655.39	709.75	8.29%
	(4, 2048, 4096, 2816)	635.02	677.51	6.69%
	(8, 1024, 2816, 4096)	633.35	677.56	6.98%
	(8, 1024, 4096, 2816)	619.97	637.71	2.86%
	(8, 2048, 2816, 4096)	666.00	706.85	6.13%
	(8, 2048, 4096, 2816)	650.53	695.38	6.89%

To allow large tile sizes and reduce scheduling overhead, we align M_i (token segment of expert i) to a fixed block size. This fixed block size must be a multiple of `M` in tile shape modifier `mMnNkK` of the asynchronous warpgroup level matrix multiply-accumulate (WGMMA) (`wgmma.mma_async`) instruction. Thus, the warpgroups in a single threadblock will have a unified tiling, and the whole token segment (M_i) processed by a threadblock necessarily belongs to the same expert, making the scheduling very similar to that of normal GEMMs.

Compared to NVIDIA's Grouped GEMM APIs in Transformer Engine (v2.1), our implementation exhibits notable advantages. Table 1 presents a performance comparison for forward and backward computations on the H800, where tokens are evenly routed to experts. Our approach achieves an average performance improvement of 14.00% for forward computation and 6.68% for backward computation.

4 Pre-training

4.1 Pre-training Data

The quantity, quality, and diversity of training data play a crucial role in determining the performance of language models. To achieve these objectives, we have structured our approach into three key stages: document preparation, rule-based processing, and model-based processing. This design allows us to efficiently manage large volumes of data, even with limited computational resources. Document preparation focuses on preprocessing and organizing the raw data. Rule-based processing aims to

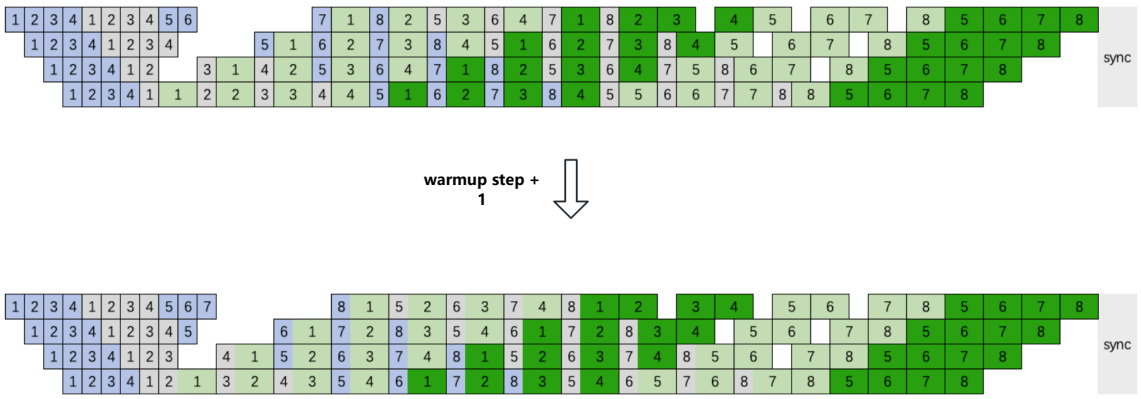

图 2：交错的 1F1B 基于通信和计算的重叠

表1：Transformer Engine 2.1 与我们在 H800 上实现的性能比较。在这里，g、m、n 和 k 分别代表组（专家）数量和每个子 GEMM 问题的维度。

	(g, m, n, k)	TE's (TFLOPS)	Ours (TFLOPS)	Speed-up
forward	(4, 1024, 2816, 4096)	589.12	662.48	12.45%
	(4, 1024, 4096, 2816)	581.77	725.35	24.68%
	(4, 2048, 2816, 4096)	636.49	678.89	6.66%
	(4, 2048, 4096, 2816)	636.94	745.45	17.04%
	(8, 1024, 2816, 4096)	614.87	654.77	6.49%
	(8, 1024, 4096, 2816)	609.69	745.01	22.19%
	(8, 2048, 2816, 4096)	646.70	705.85	9.15%
	(8, 2048, 4096, 2816)	657.20	744.74	13.32%
backward	(4, 1024, 2816, 4096)	614.24	665.06	8.27%
	(4, 1024, 4096, 2816)	594.03	637.48	7.31%
	(4, 2048, 2816, 4096)	655.39	709.75	8.29%
	(4, 2048, 4096, 2816)	635.02	677.51	6.69%
	(8, 1024, 2816, 4096)	633.35	677.56	6.98%
	(8, 1024, 4096, 2816)	619.97	637.71	2.86%
	(8, 2048, 2816, 4096)	666.00	706.85	6.13%
	(8, 2048, 4096, 2816)	650.53	695.38	6.89%

为了允许较大的瓷砖尺寸并减少调度开销，我们将专家 i) 的 M_i (令牌段对齐到一个固定的块大小。这个固定的块大小必须是异步 warpgroup 级矩阵乘累加（WGMMA）（wgmma.mma async）指令中瓷砖形状修饰符 mMnNkK 的 M 的倍数。因此，单个线程块中的 warpgroups 将具有统一的瓷砖划分，整个由线程块处理的令牌段 (M_i) 必然属于同一专家，从而使调度非常类似于普通 GEMMs。

与NVIDIA的Transformer Engine（v2.1）中的分组GEMM API相比，我们的实现具有显著优势。表1展示了在H800上前向和后向计算的性能对比，其中令牌均匀分配给专家。我们的方法在前向计算中实现了平均14.00%的性能提升，在后向计算中实现了6.68%的性能提升。

4 预训练

4.1 预训练数据

训练数据的数量、质量和多样性在决定语言模型性能方面起着关键作用。为了实现这些目标，我们将方法分为三个关键阶段：文档准备、基于规则的处理和基于模型的处理。这一设计使我们即使在计算资源有限的情况下，也能高效管理大量数据。文档准备专注于对原始数据的预处理和组织。基于规则的处理旨在

minimize the need for extensive human curation by automatically filtering and cleaning the data. Model-based processing further ensures that the final dataset is both high-quality and diverse. To ensure safety, we applied filters to exclude data from websites likely to contain unsafe content or significant amounts of personally identifiable information (PII), as well as domains flagged as harmful by a safety classifier. During pre-training, we maintain a balanced 1:1 ratio of Chinese to English data, and no synthetic data is used.

The data processing pipeline is thoroughly documented in Appendix C. Two key innovations distinguish our data processing pipeline as follows:

Web Clutter Removal Model To address issues such as boilerplate content and repetitive lines, we develop a lightweight model that operates at the line level. This approach achieves an effective balance between cleaning quality and computational efficiency, representing a unique feature that is not commonly found in open-source datasets.

Category Balancing We train a 200-class classifier to balance the proportions within the web data. This enables us to increase the presence of knowledge-based and factual content, such as encyclopedia entries and popular science articles, while reducing the share of fictional and highly structured web content, including science fiction novels and product descriptions.

4.2 Hyper-Parameters

The `dots.llm1` model is trained using the AdamW optimizer (Loshchilov & Hutter, 2017), with $\beta_1 = 0.9$ and $\beta_2 = 0.95$. We implement a weight decay of 0.1 and apply gradient clipping at 1.0. Following Dai et al. (2024); DeepSeek-AI (2024b); DeepSeek-AI et al. (2024), the weights are initialized from a normal distribution with standard deviation 0.006. `dots.llm1` comprises 62 layers, with the first layer utilizing a vanilla dense FFN and the subsequent layers employing MoE.

We set the maximum sequence length to 8K during pre-training and train `dots.llm1` on 11.2T tokens. Following Hu et al. (2024), we utilize a warmup-stable-decay learning rate schedule. The learning rate warms up over 4,000 steps before stabilizing at 3×10^{-4} for the stable training phase, which encompasses 10T tokens of data. We progressively increase the batch size from 64M tokens initially to 96M tokens at 6T tokens, and finally to 128M tokens at 8.3T tokens.

After the main training phase, the process includes two annealing stages, comprising a total of 1.2 trillion tokens of data.

Stage 1: We train for 1T tokens while gradually decreasing the learning rate from 3×10^{-4} to 3×10^{-5}. During this annealing stage, we significantly increase the proportion of reasoning-related and knowledge-related data to 90%.

Stage 2: We continue training for 200B tokens while reducing the learning rate from 3×10^{-5} to 1×10^{-5} and increasing the proportion of code, mathematics, and reasoning data.

4.3 Long Context Extension

We implement context length extension after the annealing phases. During this phase, we maintain a constant learning rate while training on 128B tokens using UtK strategy (Tian et al., 2024b), extending the sequence length to 32K. Rather than modifying the datasets, UtK tries chunking of training documents into smaller segments, then training the model to reconstruct relevant segments from shuffled chunks. By learning to untie these knotted chunks, the model can effectively handle longer input sequences while maintaining its performance on short-context tasks.

4.4 Evaluation

4.4.1 Benchmarks

To comprehensively evaluate `dots.llm1` model, which is pretrained on both Chinese and English, we assess its performance across a suite of benchmarks spanning multiple domains in each language. All evaluations are conducted using the vLLM framework (Kwon et al., 2023). We categorize the benchmarks as follows:

Language Understanding To evaluate English contextual reasoning and reading comprehension, we employ benchmarks such as HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), the ARC suite (Clark et al., 2018), BigBenchHard (BBH) (Suzgun et al., 2023), and DROP (Dua et al., 2019).

通过自动过滤和清理数据，最大程度地减少对大量人工整理的需求。基于模型的处理进一步确保最终数据集既高质量又多样化。为了确保安全，我们应用了过滤器，排除可能包含不安全内容或大量个人身份信息（PII）的网站数据，以及被安全分类器标记为有害的域名。在预训练期间，我们保持中英文数据的比例为1:1的平衡，并且不使用合成数据。

数据处理流程在附录C中有详细记录。我们的数据处理流程有两个关键创新，具体如下：

网页杂乱内容去除模型 为了解决模板内容和重复行等问题，我们开发了一个在行级别操作的轻量级模型。这种方法在清理质量和计算效率之间实现了有效的平衡，具有在开源数据集中不常见的独特特性。

类别平衡 我们训练了一个200类的分类器，以平衡网络数据中的比例。这使我们能够增加基于知识和事实内容的比例，例如百科全书条目和科普文章，同时减少虚构和高度结构化的网络内容的份额，包括科幻小说和产品描述。

4.2 超参数

dots.llm1 模型使用 AdamW 优化器（Loshchilov & Hutter, 2017）进行训练，$\beta_1 =$ 取值为 0.9，$\beta_2 =$ 取值为 0.95。我们实施了 0.1 的权重衰减，并在 1.0 处应用梯度裁剪。根据 Dai 等人（2024）；DeepSeek-AI（2024b）；DeepSeek-AI 等人（2024）的做法，权重从标准差为 0.006的正态分布中初始化。dots.llm1 包含 62 层，第一层采用普通的密集前馈网络（FFN），后续层则采用 MoE。

在预训练期间，我们将最大序列长度设为8K，并在11.2T个tokens上训练dots.llm1。根据Hu等人（2024）的做法，我们采用了预热-稳定-衰减的学习率调度策略。学习率在4,000步内预热，然后在稳定训练阶段保持在3×10^{-4}，该阶段涵盖了10T个tokens的数据。我们逐步将批次大小从最初的64M tokens增加到6T tokens时的96M tokens，最终在8.3T tokens时达到128M tokens。

在主要训练阶段之后，过程包括两个退火阶段，总共处理了1.2万亿个令牌的数据。

阶段1：我们训练1万亿个标记，同时将学习率从3×10^{-4}逐步降低到3×10^{-5}。在这个退火阶段，我们将推理相关和知识相关数据的比例显著提高到90%。

阶段2：我们继续训练2000亿个标记，同时将学习率从3×10^{-5} 降低到1×10^{-5}，并增加代码、数学和推理数据的比例。

4.3 长上下文扩展

我们在退火阶段之后实现上下文长度扩展。在此阶段，我们在使用 UtK 策略（Tian 等，2024b）对 128B 令牌进行训练时保持恒定的学习率，将序列长度扩展到 32K。UtK 并不修改数据集，而是尝试将训练文档分块成更小的段，然后训练模型从打乱的块中重建相关段。通过学习解开这些打结的块，模型可以有效处理更长的输入序列，同时保持在短上下文任务中的性能。

4.4 评估

4.4.1 基准测试

为了全面评估在中英文预训练的 dots.llm1 模型，我们在每种语言的多个领域中一系列基准测试中评估其性能。所有评估均使用 vLLM 框架（Kwon 等，2023）进行。我们将基准测试分类如下：

语言理解 为了评估英语的上下文推理和阅读理解能力，我们采用了诸如 HellaSwag（Zellers 等，2019）、PIQA（Bisk 等，2020）、ARC 套件（Clark 等，2018）、BigBenchHard（BBH）（Suzgun 等，2023）以及 DROP（Dua 等，2019）等基准测试。

Table 2: Comparison between dots.llm1 and other representative open-source base models. All models were evaluated under identical conditions. Results demonstrate that dots.llm1 achieves superior performance compared to DeepSeek-V2, while delivering results comparable to Qwen2.5 72B.

	Benchmark (Metric)	# Shots	Qwen2.5 32B Base	Qwen2.5 72B Base	DeepSeek V2 Base	DeepSeek V3 Base	dots.llm1 .base
	Architecture	-	Dense	Dense	MoE	MoE	MoE
	# Activated Params	-	32B	72B	21B	37B	14B
	# Total Params	-	32B	72B	236B	671B	142B
Chinese	C-Eval (EM)	5-shot	87.4	89.3	80.6	89.1	92.8
	CMMLU (EM)	5-shot	88.5	89.7	82.9	88.1	90.4
	CLUEWSC (EM)	5-shot	93.2	93.2	92.2	92.6	92.7
	C3 (EM)	5-shot	97.2	97.1	96.7	97.4	97.6
	Xiezhi (EM)	3-shot	81.2	82.3	77.4	80.4	82.9
	Average		89.5	90.3	86.0	89.5	91.3
English	TriviaQA (EM)	5-shot	78.5	85.2	89.2	91.4	88.2
	BBH (EM)	3-shot	82.2	83.7	76.5	83.5	80.8
	MMLU (EM)	5-shot	83.5	85.4	78.0	86.8	83.2
	MMLU-Pro (EM)	5-shot	61.8	64.7	52.3	61.4	61.9
	SuperGPQA (EM)	5-shot	30.5	32.1	29.0	38.6	33.3
	DROP (F1)	3-shot	82.9	84.9	78.9	89.1	83.4
	ARC-Challenge (EM)	25-shot	93.1	95.2	92.1	96.2	93.8
	AGIEval (EM)	3-shot	76.1	76.9	68.34	77.1	79.9
	HellaSwag (EM)	10-shot	93.3	94.2	88.1	89.7	88.2
	PIQA (EM)	5-shot	92.8	94.5	90.9	94.0	91.0
	NaturalQuestions (EM)	3-shot	35.3	42.7	46.8	50.7	48.7
	Average		73.2	76.3	71.8	78.0	75.7
MATH	GSM8K (EM)	8-shot	88.9	89.9	80.0	90.1	86.7
	MATH (EM)	4-shot	71.7	76.2	42.3	72.8	68.7
	CMath (EM)	3-shot	79.2	65.7	72.2	83.3	79.5
	Average		79.9	77.3	64.8	82.1	78.3
Code	HumanEval (Pass@1)	0-shot	50.6	56.7	43.3	62.8	64.0
	MBPP (Pass@1)	3-shot	77.0	77.8	75.1	76.3	76.7
	MCEval (Pass@1)	0-shot	47.9	47.5	43.5	50.1	46.6
	BigCodeBench (Pass@1)	0-shot	51.9	54.0	41.8	60.7	50.9
	Average		56.9	59.0	50.9	62.5	59.6

For English closed-book question answering, we utilize TriviaQA (Joshi et al., 2017) and Natural Questions (Kwiatkowski et al., 2019). For Chinese language understanding, we assess reference disambiguation using CLUEWSC (Xu et al., 2020) and reading comprehension using C3 (Sun et al., 2019).

Knowledge To measure domain knowledge in English, we leverage the MMLU (Hendrycks et al., 2021a), MMLU-Pro (Wang et al., 2024b) and SuperGPQA (Team et al., 2025) benchmarks as well as for creative problem-solving and AGIEval (Zhong et al., 2023) for standardized exam performance. For the assessment of Chinese knowledge, we employ C-Eval (Huang et al., 2023), CMMLU (Li et al., 2023), and Xiezhi (Gu et al., 2024).

Mathematics To evaluate mathematical reasoning capabilities, we utilize CMath (Wei et al., 2023) and GSM8K (Cobbe et al., 2021) for foundational problem-solving tasks, as well as MATH (Hendrycks et al., 2021b) for assessing advanced mathematical problem-solving proficiency.

Code Code generation skills are evaluated with HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021) and MCEval (Chai et al., 2024). Additionally, the ability to handle diverse function calls and complex instructions is measured through the BigCodeBench benchmark (Zhuo et al., 2025).

4.4.2 Results

As shown in Table 2, we compare dots.llm1 with other leading open-source base models under identical conditions. The results demonstrate that dots.llm1, with only 14B activated parameters, achieves superior performance compared to DeepSeek-V2, and delivers results comparable to Qwen2.5 72B. This makes dots.llm1 one of the most economical and powerful mixture-of-experts models available to date.

For a detailed breakdown across different categories, we use Qwen2.5 72B as the baseline for comparison.

表2：dots.llm1与其他具有代表性的开源基础模型的比较。所有模型在相同条件下进行了评估。结果显示，dots.llm1在性能上优于DeepSeek-V2，同时其表现与Qwen2.5 72B相当。

	Benchmark (Metric)	# Shots	Qwen2.5 32B Base	Qwen2.5 72B Base	DeepSeek V2 Base	DeepSeek V3 Base	dots.llm1 .base
	Architecture	-	Dense	Dense	MoE	MoE	MoE
	# Activated Params	-	32B	72B	21B	37B	14B
	# Total Params	-	32B	72B	236B	671B	142B
Chinese	C-Eval (EM)	5-shot	87.4	89.3	80.6	89.1	92.8
	CMMLU (EM)	5-shot	88.5	89.7	82.9	88.1	90.4
	CLUEWSC (EM)	5-shot	93.2	93.2	92.2	92.6	92.7
	C3 (EM)	5-shot	97.2	97.1	96.7	97.4	97.6
	Xiezhi (EM)	3-shot	81.2	82.3	77.4	80.4	82.9
	Average		89.5	90.3	86.0	89.5	91.3
English	TriviaQA (EM)	5-shot	78.5	85.2	89.2	91.4	88.2
	BBH (EM)	3-shot	82.2	83.7	76.5	83.5	80.8
	MMLU (EM)	5-shot	83.5	85.4	78.0	86.8	83.2
	MMLU-Pro (EM)	5-shot	61.8	64.7	52.3	61.4	61.9
	SuperGPQA (EM)	5-shot	30.5	32.1	29.0	38.6	33.3
	DROP (F1)	3-shot	82.9	84.9	78.9	89.1	83.4
	ARC-Challenge (EM)	25-shot	93.1	95.2	92.1	96.2	93.8
	AGIEval (EM)	3-shot	76.1	76.9	68.34	77.1	79.9
	HellaSwag (EM)	10-shot	93.3	94.2	88.1	89.7	88.2
	PIQA (EM)	5-shot	92.8	94.5	90.9	94.0	91.0
	NaturalQuestions (EM)	3-shot	35.3	42.7	46.8	50.7	48.7
	Average		73.2	76.3	71.8	78.0	75.7
MATH	GSM8K (EM)	8-shot	88.9	89.9	80.0	90.1	86.7
	MATH (EM)	4-shot	71.7	76.2	42.3	72.8	68.7
	CMath (EM)	3-shot	79.2	65.7	72.2	83.3	79.5
	Average		79.9	77.3	64.8	82.1	78.3
Code	HumanEval (Pass@1)	0-shot	50.6	56.7	43.3	62.8	64.0
	MBPP (Pass@1)	3-shot	77.0	77.8	75.1	76.3	76.7
	MCEval (Pass@1)	0-shot	47.9	47.5	43.5	50.1	46.6
	BigCodeBench (Pass@1)	0-shot	51.9	54.0	41.8	60.7	50.9
	Average		56.9	59.0	50.9	62.5	59.6

对于英语闭卷问答，我们使用 TriviaQA（Joshi 等，2017）和 Natural Questions（Kwiatkowski 等，2019）。对于中文理解，我们使用 CLUEWSC（Xu 等，2020）评估指代消歧，并使用 C3（Sun 等，2019）进行阅读理解。

知识 为了衡量英语领域的知识水平，我们采用了 MMLU（Hendrycks 等人，2021a）、MMLU-Pro（Wang 等人，2024b）和 SuperGPQA（Team 等人，2025）基准测试，以及用于创造性问题解决的 AGIEval（Zhong 等人，2023）以评估标准化考试表现。对于中文知识的评估，我们使用 C-Eval（Huang 等人，2023）、CMMLU（Li 等人，2023）和 Xiezhi（Gu 等人，2024）。

数学 为了评估数学推理能力，我们使用 CMath（Wei 等人，2023）和 GSM8K（Cobbe 等人，2021）进行基础问题解决任务，以及 MATH（Hendrycks 等人，2021b）来评估高级数学问题解决能力。

代码 代码生成技能通过 HumanEval (Chen et al., 2021)、MBPP (Austin et al., 2021) 和 MCEval (Chai et al., 2024) 进行评估。此外，通过 BigCodeBench 基准测试 (Zhuo et al., 2025) 测量处理多样函数调用和复杂指令的能力。

4.4.2 结果

如表2所示，我们在相同条件下将 dots.llm1 与其他领先的开源基础模型进行了比较。结果显示，只有 14B 激活参数的 dots.llm1 在性能上优于 DeepSeek-V2，并且其表现与 Qwen2.5 72B 相当。这使得 dots.llm1 成为迄今为止最经济且最强大的专家混合模型之一。

为了对不同类别进行详细的细分，我们以 Qwen2.5 72B 作为基线进行比较。

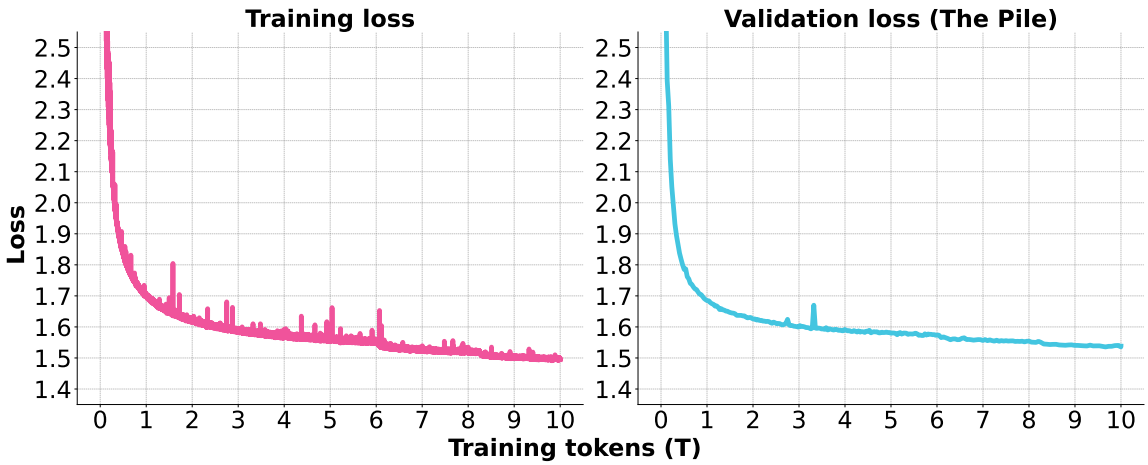

Figure 3: The loss curve highlights the consistent stability of the training process. At 6 trillion training tokens, we adjusted our batch size from 64 million to 96 million. At 8.3 trillion, we further increased it to 128 million.

Table 3: Long Context Performance on the RULER benchmark.

Model	4K	8K	16K	32K
Qwen2.5 72B	96.5	94.3	93.1	92.7
`dots.llm1`	94.7	94.9	92.6	87.7

`dots.llm1` demonstrates comparable performance to Qwen2.5 72B across most domains. (1) On language understanding tasks, `dots.llm1` achieves superior performance on Chinese language understanding benchmarks, which can be attributed to our data processing pipeline. (2) In knowledge tasks, while `dots.llm1` shows slightly lower scores on English knowledge benchmarks, its performance on Chinese knowledge tasks remains robust. (3) In the code and mathematics domains, `dots.llm1` achieves higher scores on HumanEval and CMath. Interestingly, for mathematics, we observe that `dots.llm1` achieves better performance in the zero-shot setting compared to the few-shot setting, with an improvement of over 4 points. We leave further investigation of this phenomenon to future work.

4.5 Analysis

Stable Training As depicted in Figure 3, the loss curve highlights the remarkable stability of the training process. Throughout the training period, there are no incidents of irrecoverable loss spikes or the need for rollback operations.

High Quality Web Data To assess the quality of our web data, we conduct validation experiments comparing it to the TxT360 dataset (Tang et al., 2024), which represents the current state-of-the-art (SOTA) in open-source web data. The experiments are performed using a 1.5 billion parameter dense model, with the same architecture as Qwen2.5 1.5B. We randomly sample 350 billion tokens for training to obtain reliable experimental results. As illustrated in Figure 4, our results surpass those of TxT360, demonstrating the high quality of our web data through our processing pipeline.

Effective Long Context Extension We use RULER (Hsieh et al., 2024) on the base models to evaluate long-context capabilities. As shown in Table 3, `dots.llm1` delivers competitive performance at both 8K and 16K context lengths.

Economical Training Cost As highlighted in Table 4, `dots.llm1` showcases significant efficiency in terms of training costs. The comparison reveals that our approach substantially reduces the GPU hours required per one trillion tokens. Qwen2.5 72B model (in our optimized framework) requires 340K GPU hours, while `dots.llm1` reduces this to just 130K GPU hours, leading to substantial savings in computational resources and expenses. Furthermore, if we consider the entire pretraining process, `dots.llm1` requires only 1.46 million GPU hours, whereas Qwen2.5 72B consumes 6.12 million GPU hours, representing a reduction of 4$\times$ in total computational resources. This substantial gap demonstrates the cost-effectiveness and scalability of `dots.llm1`, making it a more economical choice for large-scale pretraining.

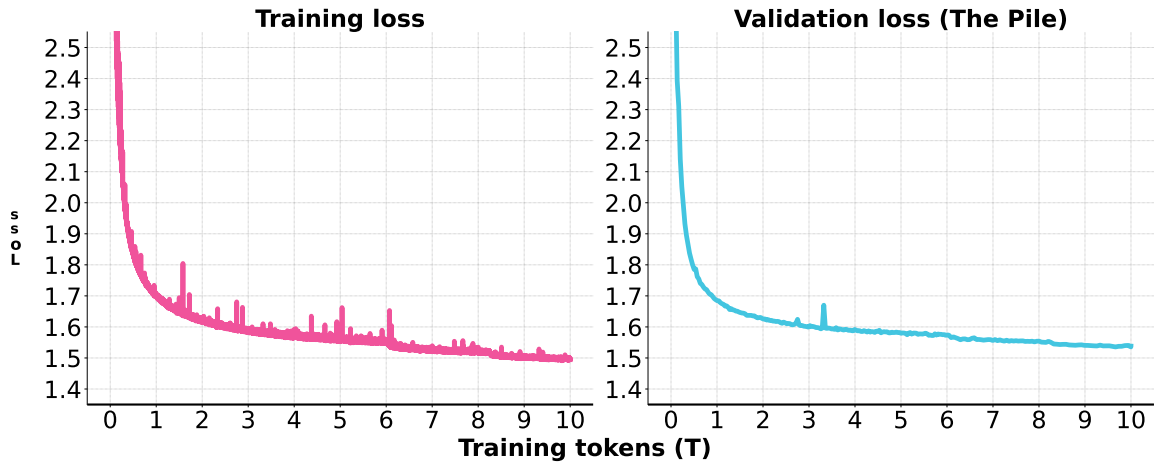

图 3：损失曲线突出显示了训练过程的一致稳定性。在训练了 6 万亿个令牌后，我们将批次大小从 6400 万调整到 9600 万。在 8.3 万亿时，我们进一步将其增加到 1.28 亿。

表3：在RULER基准测试中的长上下文性能。

Model	4K	8K	16K	32K
Qwen2.5 72B	96.5	94.3	93.1	92.7
dots.llm1	94.7	94.9	92.6	87.7

dots.llm1 在大多数领域表现出与 Qwen2.5 72B 相当的性能。 (1) 在语言理解任务中，dots.llm1 在中文语言理解基准测试中表现优越，这归功于我们的数据处理流程。 (2) 在知识任务中，虽然 dots.llm1 在英文知识基准测试中的得分略低，但其在中文知识任务中的表现依然稳健。 (3) 在代码和数学领域，dots.llm1 在 HumanEval 和 CMath 上取得了更高的分数。有趣的是，在数学方面，我们观察到 dots.llm1 在零样本设置中的表现优于少样本设置，提升超过 4 分。我们将把对此现象的进一步研究留待未来工作。

4.5 分析

稳定训练 如图3所示，损失曲线突出了训练过程的显著稳定性。在整个训练期间，没有发生无法恢复的损失峰值或需要回滚操作的情况。

高质量的网页数据 为了评估我们的网页数据的质量，我们进行了验证实验，将其与TxT360数据集（Tang 等，2024）进行比较，后者代表了开源网页数据的当前最先进水平（SOTA）。这些实验使用一个拥有15亿参数的密集模型进行，架构与Qwen2.5 1.5B相同。我们随机抽取3500亿个标记进行训练，以获得可靠的实验结果。如图4所示，我们的结果优于TxT360，展示了通过我们的处理流程所获得的网页数据的高质量。

有效的长上下文扩展 我们在基础模型上使用 RULER（Hsieh 等人，2024）来评估长上下文能力。如表 3 所示，dots.llm1 在 8K 和 16K 上下文长度下都表现出具有竞争力的性能。

经济训练成本 如表4所示，dots.llm1 在训练成本方面展现出显著的效率。比较显示，我们的方法大幅度减少了每一万亿个 token 所需的 GPU 小时数。Qwen2.5 72B 模型（在我们优化的框架下）需要 340K GPU 小时，而 dots.llm1 将其降低到仅 130K GPU 小时，显著节省了计算资源和费用。此外，如果考虑整个预训练过程，dots.llm1 仅需 1.46 百万 GPU 小时，而 Qwen2.5 72B 则消耗 6.12 百万 GPU 小时，整体计算资源减少了 4×。这一巨大差距展示了 dots.llm1 的成本效益和可扩展性，使其成为大规模预训练的更经济的选择。

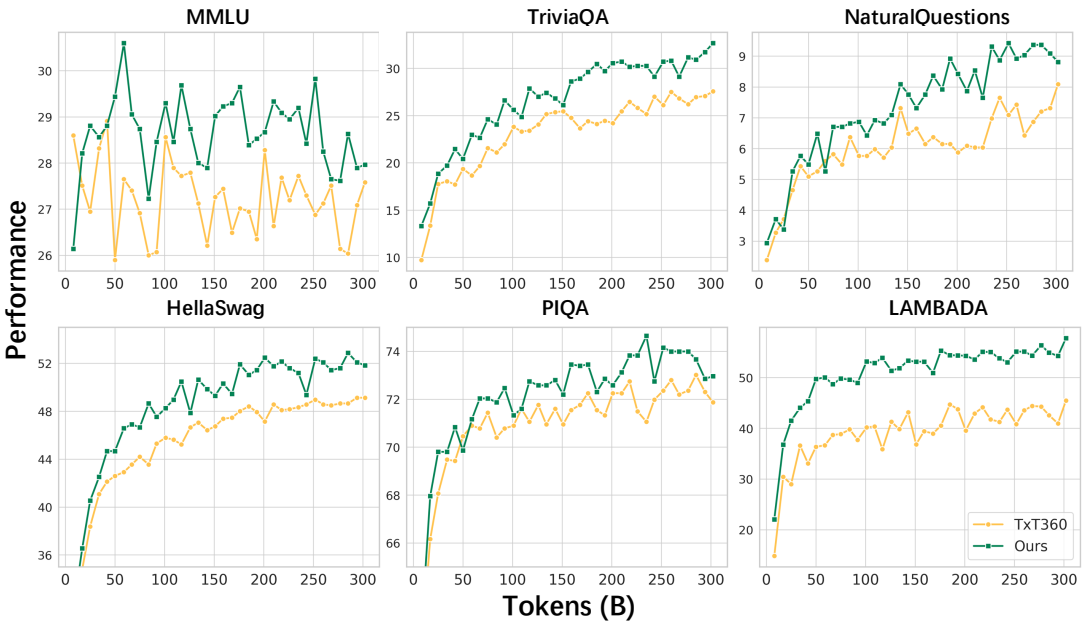

Figure 4: Comparison of performance curves between the TxT360 dataset and our web data on the MMLU, TriviaQA, NaturalQuestions, HellaSwag, PIQA, and LAMBADA (Paperno et al., 2016) benchmarks. The results demonstrate that training with our web data consistently yields superior performance.

Table 4: Comparison of GPU hours across different models, `dots.llm1` representing a reduction of 4× in total computational resources.

Model	GPU Hours per One Trillion Tokens	Pretraining Tokens	Total GPU Hours
Qwen2.5 72B	340K (×1.0)	18.0T	6,120K (×1.0)
`dots.llm1`	130K (×0.38)	11.2T	1,456K (×0.24)

5 Post-training

5.1 Supervised Fine-tuning

Data Mixture Based on open-source data and internally annotated data, we collect approximately 400K instruction tuning instances, focusing on several key areas: multi-lingual (primarily Chinese and English) multi-turn dialogues, knowledge understanding and question answering, complex instruction following, and reasoning tasks involving mathematics and coding.

For multi-turn dialogue capabilities, we integrate mainstream open-source dialogue datasets with our extensive collection of high-quality internal Chinese instruction sets. For a small subset of responses in the open-source data that are of insufficient quality, we refine them using powerful teacher models (e.g., DeepSeek-V3 0324 (DeepSeek-AI et al., 2024)). To enhance knowledge understanding QA abilities, we specifically incorporate datasets focused on factual knowledge and reading comprehension. For complex instruction following, we meticulously design instruction sets with conditional constraints and filtered out responses that failed to adhere to these conditions. To develop reasoning capabilities, we introduce verifiable mathematics and coding datasets, equipped with corresponding verifiers to filter and extract high-quality supervision signals.

Fine-tuning Configurations The fine-tuning process of dots.llm1.inst consists of two phases. In the first phase, we perform upsampling and multi-session concatenation on our 400K instruction tuning instances, then fine-tune dots.llm1.inst for 2 epochs. In the second phase, we further enhance the model's capabilities in specific domains (such as mathematics and coding) through rejection sampling fine-tuning (RFT), incorporating a verifier system to improve performance in these specialized areas. During each training phase, we employ a cosine learning rate scheduler with a learning rate of 5e-6 that gradually decayed to a minimum of 1e-6.

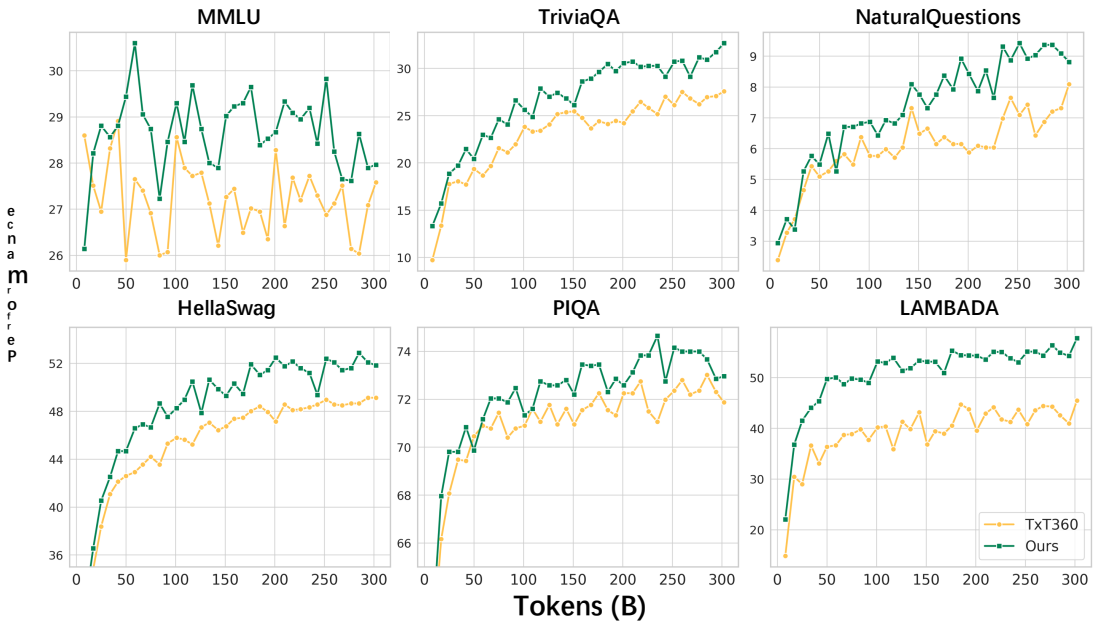

图4：在MMLU、TriviaQA、NaturalQuestions、HellaSwag、PIQA和LAMBADA（Paperno等，2016）基准测试中，TxT360数据集与我们的网页数据性能曲线的比较。结果表明，使用我们的网页数据进行训练始终能获得更优的性能。

表4：不同模型的GPU小时数比较，dots.llm1代表总计算资源减少了4×。

Model	GPU Hours per One Trillion Tokens	Pretraining Tokens	Total GPU Hours
Qwen2.5 72B	340K (×1.0)	18.0T	6,120K (×1.0)
`dots.llm1`	130K (×0.38)	11.2T	1,456K (×0.24)

5 训练后

5.1 监督微调

基于开源数据和内部标注数据的数据混合，我们收集了大约40万条指令调优实例，重点关注几个关键领域：多语言（主要是中文和英文）多轮对话、知识理解与问答、复杂指令执行，以及涉及数学和编码的推理任务。

为了实现多轮对话能力，我们将主流开源对话数据集与我们丰富的高质量内部中文指令集相结合。对于开源数据中部分质量不足的响应，我们使用强大的教师模型（例如，DeepSeek-V3 0324（DeepSeek-AI 等，2024））进行优化。为了增强知识理解和问答能力，我们特别引入了专注于事实知识和阅读理解的数据集。对于复杂的指令执行，我们精心设计了带有条件限制的指令集，并筛选出未能遵守这些条件的响应。为了培养推理能力，我们引入了可验证的数学和编码数据集，并配备相应的验证器，以筛选和提取高质量的监督信号。

微调配置 dots.llm1.inst 的微调过程包括两个阶段。在第一阶段，我们对 400K 指令微调实例进行上采样和多会话拼接，然后对 dots.llm1.inst 进行 2 个周期的微调。在第二阶段，我们通过拒绝采样微调（RFT）进一步增强模型在特定领域（如数学和编码）中的能力，结合验证系统以提升这些专业领域的性能。在每个训练阶段，我们采用余弦学习率调度器，学习率为 5e-6，逐渐衰减至最小值 1e-6。

5.2 Evaluations

Benchmarks We evaluate dots.llm1.inst on a series of post-training benchmarks and compared it with state-of-the-art models (DeepSeek-AI, 2024b; DeepSeek-AI et al., 2024). Our evaluation framework covers five main categories: English general capabilities, Chinese general capabilities, alignment, and two specific subdomains—mathematics and coding.

For English general capabilities, we primarily use representative QA datasets, such as general question-answering benchmarks like MMLU (Hendrycks et al., 2021a), MMLU-Redux (Gema et al., 2024), and MMLU-Pro (Wang et al., 2024b), expert-level question-answering benchmarks like GPQA (Rein et al., 2023), as well as reading comprehension QA datasets such as DROP (Dua et al., 2019). Additionally, SimpleQA (OpenAI, 2024) evaluates model's ability to answer fact-seeking questions correctly.

For Chinese general capabilities, we select CLUEWSC (Xu et al., 2020) for coreference resolution evaluation, C-Eval (Huang et al., 2023) for comprehensive Chinese language understanding capabilities, and C-SimpleQA (He et al., 2024) for factual question answering in Chinese.

To measure the instruction-following and human preference alignment of the dots.llm1.inst model, we select rule-based verifiable benchmarks (IFEval (Zhou et al., 2023)) and competitive arena-style benchmarks (AlpacaEval2 (Dubois et al., 2024), ArenaHard (Li et al., 2024)).

For mathematics, we evaluate on competition-level benchmarks AIME24 (MAA, 2024) and CNMO24 (Liu et al., 2024b), and standard-difficulty benchmarks MATH (Hendrycks et al., 2021b), GSM8K (Cobbe et al., 2021), and MATH500 (Lightman et al., 2023).

For coding capabilities, we conduct tests on the coding competition LiveCodeBench (Jain et al., 2024), as well as HumanEval (Chen et al., 2021) and MBPP+ (Liu et al., 2024a).

Evaluation Configurations For general question-answering benchmarks (including MMLU-Pro, DROP, GPQA, SimpleQA, and Chinese SimpleQA), we utilize evaluation prompts provided by the simple-evals[1] framework, while for MMLU-Redux, we apply the zero-shot prompt template from the Zero-Eval[2] repository. For other general and alignment datasets, we follow the evaluation methodologies designed by the original dataset creators.

In mathematics evaluations, we adhere to standard practices for mathematical datasets, using "Please reason step by step, and put your final answer within `\boxed{}`" for answer extraction. For competitive mathematics benchmarks (e.g., AIME24), we employ temperature sampling and multiple-run averaging to enhance evaluation stability. In our code evaluations for LiveCodeBench, we select data spanning from May 2024 to January 2025. For evaluating MBPP+, we utilized the v0.2.0 release of EvalPlus (Liu et al., 2023), which provides reliable assessment for code generation tasks.

5.3 Performance Analysis

English Performance Our evaluation results indicate that dots.llm1.inst demonstrates stable and comprehensive performance across multiple general English benchmarks. On question-answering tasks such as MMLU, MMLU-Redux, DROP, and GPQA, dots.llm1.inst delivers competitive results compared to the Qwen2.5 / Qwen3 series models, demonstrating solid knowledge foundations and reasoning abilities.

Code Performance The model exhibits competitive coding capabilities, though this aspect is not quite as exceptional as its performance in other areas. Compared to the Qwen2.5 series, its coding ability is on par; however, in comparison with more cutting-edge models like Qwen3 and DeepSeek-V3, there remains room for further improvement.

Math Performance The dots.llm1.inst showcases outstanding mathematical reasoning skills. The model achieves a score of 33.1 on AIME24, underscoring its advanced problem-solving abilities in complex mathematics. On MATH500, it attains a score of 84.8, outperforming the Qwen2.5 series and approaching state-of-the-art results. It also achieved a high score of 40.6 on CNMO24.

Chinese Performance The dots.llm1.inst demonstrates significant advantages in Chinese language tasks. It achieves a score of 92.6 on CLUEWSC, matching industry-leading performance in Chinese semantic understanding. On C-Eval, it achieves 92.2, surpassing all models including DeepSeek-V3

[1] https://github.com/openai/simple-evals
[2] https://github.com/WildEval/ZeroEval

5.2 评估

基准测试 我们在一系列训练后基准测试中评估了 dots.llm1.inst，并将其与最先进的模型（DeepSeek-AI, 2024b；DeepSeek-AI 等，2024）进行了比较。我们的评估框架涵盖五个主要类别：英语通用能力、中文通用能力、对齐，以及两个特定子领域——数学和编码。

对于英语通用能力，我们主要使用代表性问答数据集，例如像 MMLU (Hendrycks et al., 2021a)、MMLU-Redux (Gema et al., 2024) 和 MMLU-Pro (Wang et al., 2024b) 这样的通用问答基准测试，专家级问答基准如 GPQA (Rein et al., 2023)，以及阅读理解问答数据集如 DROP (Dua et al., 2019)。此外，SimpleQA (OpenAI, 2024) 评估模型正确回答事实性问题的能力。

对于中文通用能力，我们选择 CLUEWSC（Xu 等，2020）进行指代消解评估，选择 C-Eval（Huang 等，2023）进行全面的中文语言理解能力评估，以及 C-SimpleQA（He 等，2024）进行中文事实性问答。

为了衡量 dots.llm1.inst 模型的指令遵循能力和人类偏好对齐情况，我们选择了基于规则的可验证基准（IFEval（Zhou 等，2023））以及具有竞争性的竞技场风格基准（AlpacaEval2（Dubois 等，2024），Arena Hard（Li 等，2024））。

对于数学，我们在竞赛级基准测试 AIME24（MAA，2024）和 CNMO24（Liu 等，2024b）以及标准难度基准测试 MATH（Hendrycks 等，2021b）、GSM8K（Cobbe 等，2021）和 MATH500（Lightman 等，2023）上进行评估。

关于编码能力，我们在编码竞赛 LiveCodeBench（Jain 等人，2024）、HumanEval（Chen 等人，2021）以及 MBPP+ (Liu 等人，2024a) 上进行了测试。

评估配置 对于通用问答基准（包括 MMLU-Pro、DROP、GPQA、SimpleQA 和 Chinese SimpleQA），我们采用 simple-evals[1] 框架提供的评估提示；而对于 MMLU-Redux，我们使用 Zero-Eval[2] 仓库中的零样本提示模板。对于其他通用和对齐数据集，我们遵循原始数据集创建者设计的评估方法。

在数学评估中，我们遵循数学数据集的标准做法，使用“请逐步推理，并将你的最终答案放在 \boxed{} 内”进行答案提取。对于竞赛数学基准（例如 AIME24），我们采用温度采样和多次运行平均以增强评估的稳定性。在我们的 LiveCodeBench 代码评估中，选择的数据范围从 2024 年 5 月到 2025 年 1 月。为了评估 MBPP+，我们使用了 EvalPlus（Liu 等人，2023）发布的 v0.2.0 版本，它为代码生成任务提供了可靠的评估。

5.3 性能分析

性能 我们的评估结果显示，dots.llm1.inst 在多个通用英语基准测试中表现出稳定且全面的性能。在问答任务如 MMLU、MMLU-Redux、DROP 和 GPQA 上，dots.llm1.inst 与 Qwen2.5 / Qwen3 系列模型相比，表现具有竞争力，展现了扎实的知识基础和推理能力。

代码性能　该模型展现出具有竞争力的编码能力，尽管这一方面并不像其在其他领域的表现那样出色。与Qwen2.5系列相比，其编码能力持平；然而，与更先进的模型如Qwen3和DeepSeek-V3相比，仍有进一步提升的空间。

数学表现 dots.llm1.inst 展示了出色的数学推理能力。该模型在 AIME24 上获得了 33.1 分，突显其在复杂数学问题上的高级解决能力。在 MATH500 上，它获得了 84.8 分，超越了 Qwen2.5 系列，接近最先进的水平。在 CNMO24 上也取得了高分 40.6。

中文表现 dots.llm1.inst 在中文任务中表现出显著优势。在 CLUEWSC 上取得了 92.6 的分数，达到了行业领先的中文语义理解水平。在 C-Eval 上取得了 92.2，超越了包括 DeepSeek-V3 在内的所有模型

[1] https://github.com/openai/simple-evals
[2] https://github.com/WildEval/ZeroEval

Table 5: Comparison between dots.llm1.inst and other representative instruction-tuned models. All models were evaluated within our internal framework under identical conditions. Regarding the Qwen3 series, our evaluations were conducted without the thinking mode enabled.

	Benchmark (Metric)	Qwen-2.5 32B Inst	Qwen-2.5 72B Inst	Qwen-3 32B	Qwen-3 235B-A22B	DeepSeek V2 Chat	DeepSeek V3	gpt4o 0806	dots.llm1 .inst
	Architecture	Dense	Dense	Dense	MoE	MoE	MoE	-	MoE
	# Activated Params	32B	72B	32B	22B	21B	37B	-	14B
	# Total Params	32B	72B	32B	235B	236B	671B	-	142B
English	MMLU (EM)	83.4	84.4	82.9	86.4	78.6	87.9	86.7	82.1
	MMLU-Redux (EM)	83.3	85.9	86.1	88.6	77.9	88.8	87.3	85.1
	MMLU-Pro (EM)	68.9	71.0	68.3	69.9	60.2	76.0	74.4	70.4
	DROP (F1)	77.8	76.9	88.6	84.8	73.3	91.8	87.6	87.0
	GPQA Diamond (Pass@1)	47.2	49.9	54.5	59.4	32.6	56.1	50.1	52.6
	SimpleQA (Correct)	6.3	9.6	6.7	12.1	12.1	24.6	38.8	9.3
	Average	61.2	62.9	64.5	66.9	55.8	70.9	70.8	64.4
Code	HumanEval (Pass@1)	87.8	84.8	90.2	90.2	79.3	91.5	92.1	88.4
	MBPP+ (Pass@1)	74.6	74.6	74.9	75.9	67.2	75.7	75.9	74.5
	LiveCodeBench (Pass@1)	31.2	32.7	36.7	43.5	22.0	42.7	38.1	32.0
	Average	64.5	64.0	67.3	69.9	56.2	70.0	68.7	65.0
Math	MATH (EM)	82.9	83.6	86.2	88.6	57.1	89.7	79.4	85.0
	AIME24 (Pass@1)	16.5	18.8	27.1	37.5	2.9	34.0	13.3	33.1
	MATH500 (Pass@1)	81.9	83.1	84.8	88.1	56.2	88.9	78.4	84.8
	CNMO24 (Pass@1)	21.1	21.9	26.0	33.2	4.3	33.9	15.6	40.6
	Average	50.6	51.9	56.0	61.9	30.1	61.6	46.7	60.9
Chinese	CLUEWSC (EM)	91.7	93.1	91.6	91.4	93.0	93.5	90.3	92.6
	C-Eval (EM)	87.4	88.5	82.8	84.4	78.4	86.3	77.8	92.2
	C-SimpleQA (Correct)	42.3	51.4	46.6	62.0	56.5	68.9	61.4	56.7
	Average	73.8	77.7	73.7	79.3	76.0	82.9	76.5	80.5
Alignment	IFEval (Prompt Strict)	78.9	84.8	84.1	84.1	58.4	86.1	85.2	82.1
	AlpacaEval2 (GPT-4o Judge)	50.0	51.3	58.1	66.8	40.4	66.5	53.6	64.4
	ArenaHard (GPT-4o Judge)	76.7	85.6	91.5	95.7	41.8	92.1	85.1	87.1
	Average	68.5	73.9	77.9	82.2	46.9	81.6	74.6	77.9

(86.3), thus demonstrating comprehensive mastery of Chinese knowledge. For C-SimpleQA, it scores 56.7, substantially outperforming Qwen-2.5 72B Instruct, though still behind DeepSeek-V3 (68.9).

Alignment Performance With respect to instruction following and alignment with human preferences, dots.llm1.inst demonstrates competitive performance on benchmarks such as IFEval, AlpacaEval2, and ArenaHard. These results indicate that the model can accurately interpret and execute complex instructions while maintaining consistency with human intentions and values.

Overall Based on the evaluation results, dots.llm1.inst demonstrates exceptional performance across Chinese and English general tasks, mathematical reasoning, code generation, and alignment benchmarks while activating only 14B parameters. The model exhibits strong competitive advantages compared to Qwen2.5-32B-Instruct and Qwen2.5-72B-Instruct, and achieves comparable or superior performance relative to Qwen3-32B in bilingual tasks, mathematical reasoning, and alignment capabilities, establishing itself as a highly efficient open-source solution.

6 MoE Analysis

We examine the expert load of the `dots.llm1` model on the Pile test set. For a specific domain D, with N_D tokens processed from this domain by the MoE, the expert load of an expert E_i is defined as:

$$\text{Expert Load}(E_i, D) = \frac{N_{E_i,D}}{N_D}, \quad \in [0, 1] \tag{1}$$

where $N_{E_i,D}$ represents the number of tokens from domain D that are routed to expert E_i. The expert load indicates how specialized expert E_i is for domain D. A value of 1 means that every token from the domain is routed to the expert, while a value of 0 implies the expert is never used for that domain.

`dots.llm1` shows stronger expert specialization across all layers. In Figure 5, we observe some experts that are activated significantly more than random selection for specific domains, such as DM Mathematics. In contrast, compared to DM Mathematics, the expert load in Wikipedia appears more balanced. This could be because Wikipedia contains a wide range of knowledge and acts like a knowledge graph

表5：dots.llm1.inst 与其他具有代表性的指令调优模型的比较。所有模型均在我们的内部框架下，在相同条件下进行了评估。关于 Qwen3 系列，我们的评估是在未启用思考模式的情况下进行的。

	Benchmark (Metric)	Qwen-2.5 32B Inst	Qwen-2.5 72B Inst	Qwen-3 32B	Qwen-3 235B-A22B	DeepSeek V2 Chat	DeepSeek V3	gpt4o 0806	dots.llm1 .inst
	Architecture	Dense	Dense	Dense	MoE	MoE	MoE	-	MoE
	# Activated Params	32B	72B	32B	22B	21B	37B	-	14B
	# Total Params	32B	72B	32B	235B	236B	671B	-	142B
English	MMLU (EM)	83.4	84.4	82.9	86.4	78.6	87.9	86.7	82.1
	MMLU-Redux (EM)	83.3	85.9	86.1	88.6	77.9	88.8	87.3	85.1
	MMLU-Pro (EM)	68.9	71.0	68.3	69.9	60.2	76.0	74.4	70.4
	DROP (F1)	77.8	76.9	88.6	84.8	73.3	91.8	87.6	87.0
	GPQA Diamond (Pass@1)	47.2	49.9	54.5	59.4	32.6	56.1	50.1	52.6
	SimpleQA (Correct)	6.3	9.6	6.7	12.1	12.1	24.6	38.8	9.3
	Average	61.2	62.9	64.5	66.9	55.8	70.9	70.8	64.4
Code	HumanEval (Pass@1)	87.8	84.8	90.2	90.2	79.3	91.5	92.1	88.4
	MBPP+ (Pass@1)	74.6	74.6	74.9	75.9	67.2	75.7	75.9	74.5
	LiveCodeBench (Pass@1)	31.2	32.7	36.7	43.5	22.0	42.7	38.1	32.0
	Average	64.5	64.0	67.3	69.9	56.2	70.0	68.7	65.0
Math	MATH (EM)	82.9	83.6	86.2	88.6	57.1	89.7	79.4	85.0
	AIME24 (Pass@1)	16.5	18.8	27.1	37.5	2.9	34.0	13.3	33.1
	MATH500 (Pass@1)	81.9	83.1	84.8	88.1	56.2	88.9	78.4	84.8
	CNMO24 (Pass@1)	21.1	21.9	26.0	33.2	4.3	33.9	15.6	40.6
	Average	50.6	51.9	56.0	61.9	30.1	61.6	46.7	60.9
Chinese	CLUEWSC (EM)	91.7	93.1	91.6	91.4	93.0	93.5	90.3	92.6
	C-Eval (EM)	87.4	88.5	82.8	84.4	78.4	86.3	77.8	92.2
	C-SimpleQA (Correct)	42.3	51.4	46.6	62.0	56.5	68.9	61.4	56.7
	Average	73.8	77.7	73.7	79.3	76.0	82.9	76.5	80.5
Alignment	IFEval (Prompt Strict)	78.9	84.8	84.1	84.1	58.4	86.1	85.2	82.1
	AlpacaEval2 (GPT-4o Judge)	50.0	51.3	58.1	66.8	40.4	66.5	53.6	64.4
	ArenaHard (GPT-4o Judge)	76.7	85.6	91.5	95.7	41.8	92.1	85.1	87.1
	Average	68.5	73.9	77.9	82.2	46.9	81.6	74.6	77.9

(86.3)，因此展现了对中文知识的全面掌握。在C-SimpleQA中，它得分56.7，远远优于Qwen-2.5 72B Instruct，但仍落后于DeepSeek-V3（68.9）。

关于指令执行和与人类偏好的对齐，dots.llm1.inst 在 IFEval、AlpacaEval2 和 ArenaHard 等基准测试中表现出具有竞争力的性能。这些结果表明，该模型能够准确理解和执行复杂指令，同时保持与人类意图和价值观的一致性。

总体而言，基于评估结果，dots.llm1.inst 在中文和英文的通用任务、数学推理、代码生成和对齐基准测试中表现出色，同时仅激活了 14B 参数。该模型在与 Qwen2.5-32B-Instruct 和 Qwen2.5-72B-Instruct 的比较中展现出强大的竞争优势，并在双语任务、数学推理和对齐能力方面实现了与 Qwen3-32B 相当或更优的性能，确立了其作为一种高效的开源解决方案的地位。

6 MoE 分析

我们检查 dots.llm1 模型在 Pile 测试集上的专家负载。对于特定领域 D，由 MoE 处理的该领域中的 N_D 个标记，专家 E_i 的专家负载定义为：

$$\text{Expert Load}(E_i, D) = \frac{N_{E_i,D}}{N_D}, \quad \in [0,1] \tag{1}$$

其中 $N_{E_i,D}$ 表示从域 D 路由到专家 E_i 的标记数。专家负载表示专家 E_i 在域 D 的专业程度。值为 1 表示该域的每个标记都路由到该专家，而值为 0 则意味着该专家从不用于该域。

dots.llm1 在所有层次上都显示出更强的专家专业化。在图5中，我们观察到一些专家在特定领域（如DM数学）中被激活的程度明显高于随机选择。相比之下，与DM数学相比，维基百科中的专家负载似乎更为均衡。这可能是因为维基百科包含了广泛的知识，并且类似于一个知识图谱

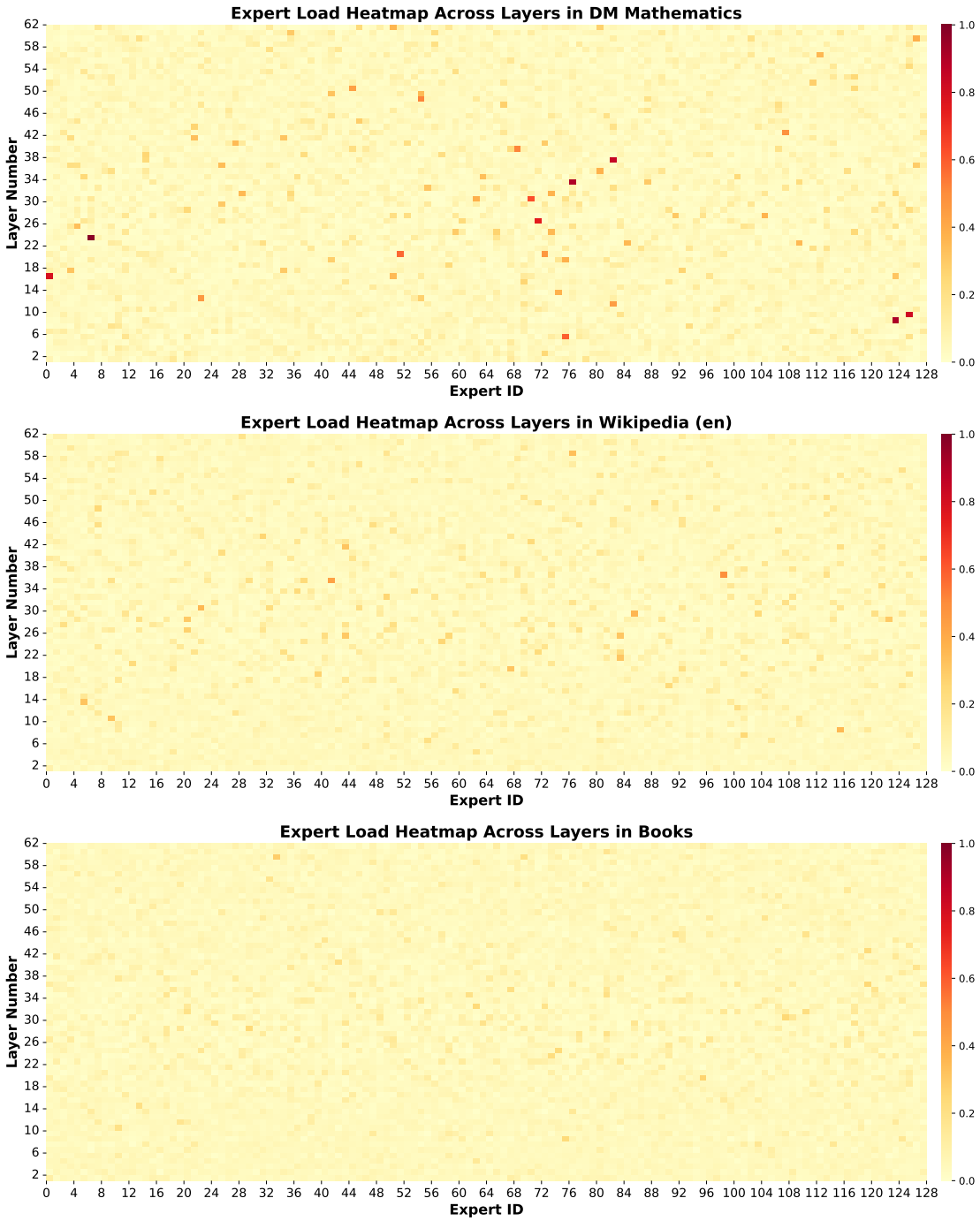

Figure 5: Expert load heatmap across layers on the Pile test dataset.

encompassing various entities. For the Books domain, which contains various types of data, the expert loads appear more balanced. This highlights that the load balancing function works as intended, enabling the model to effectively utilize all experts for handling generic data.

7 Conclusion and Future Work

We introduce `dots.llm1`, a cost-efficient mixture-of-experts model that achieves state-of-the-art performance for its size. By activating only a subset of parameters per token, `dots.llm1` significantly reduces training costs while delivering results comparable with much larger models. Our advanced data processing pipeline produces high-quality training data, and the open-sourcing of intermediate checkpoints provides valuable insights into model learning dynamics. `dots.llm1` demonstrates that efficient design and high-quality data can continually expand the capability boundaries of large language models.

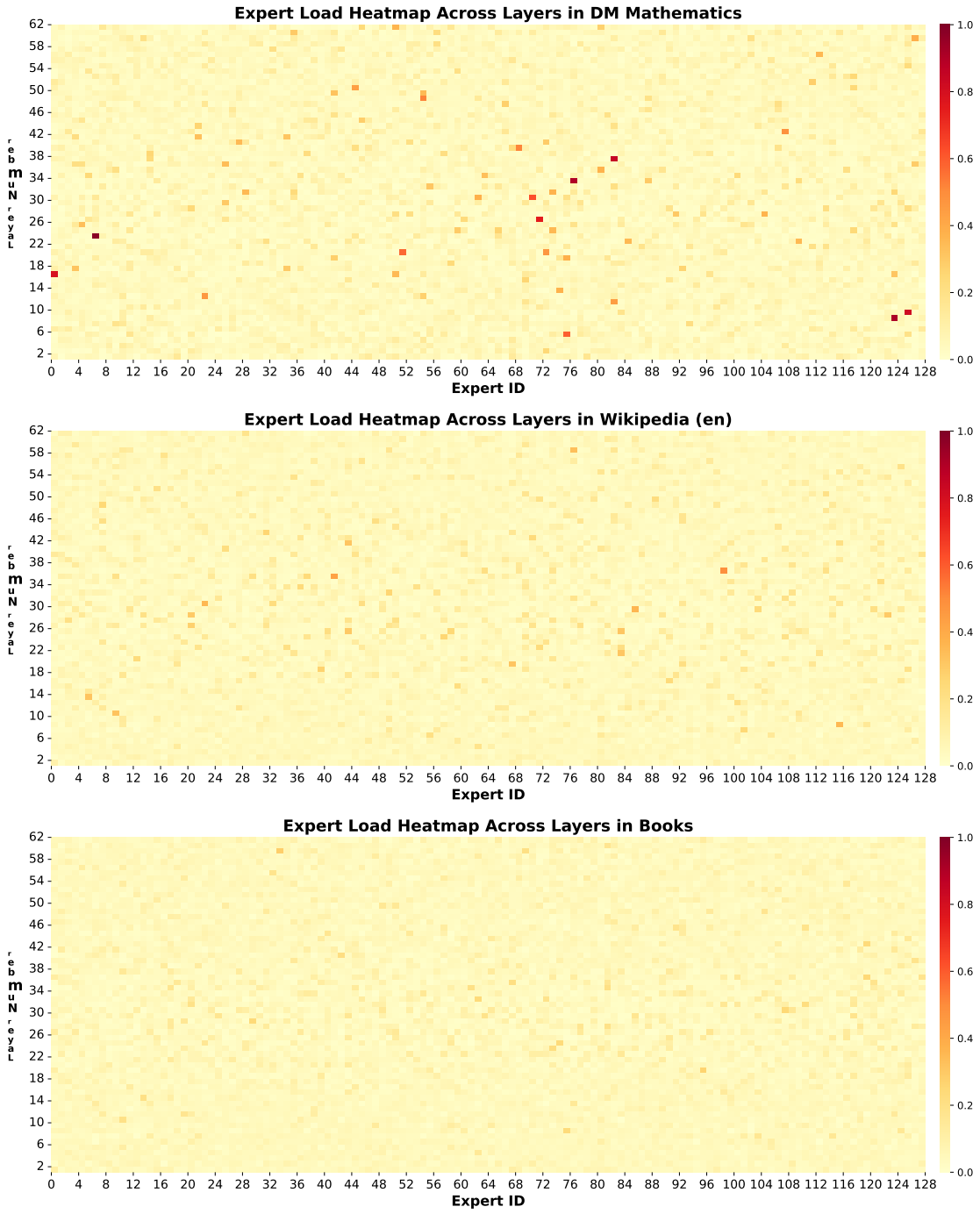

图5：在Pile测试数据集上各层专家负载热图。

涵盖各种实体。对于包含多种数据类型的图书领域，专家负载看起来更为均衡。这突显了负载均衡功能的正常工作，使模型能够有效利用所有专家处理通用数据。

7 结论与未来工作

我们介绍 dots.llm1，一种具有成本效益的专家混合模型，在其规模上实现了最先进的性能。通过每个令牌仅激活一部分参数，dots.llm1 大大降低了训练成本，同时提供与更大模型相当的结果。我们先进的数据处理流程生成高质量的训练数据，开源的中间检查点为模型学习动态提供了宝贵的见解。dots.llm1 表明，高效的设计和高质量的数据可以不断拓展大型语言模型的能力边界。

As part of our future work, we aim to train a more powerful model. To achieve an optimal balance between training and inference efficiency, we plan to integrate more efficient architectural designs such as Grouped Query Attention (GQA) (Ainslie et al., 2023), Multi-Head Latent Attention (MLA) (DeepSeek-AI, 2024b), and Linear Attention (Yang et al., 2023; Katharopoulos et al., 2020). Additionally, we intend to explore the use of more sparse mixture-of-experts (MoE) layers to improve the computational efficiency. Furthermore, since data serves as the foundation of pretraining, we will deepen our understanding of what constitutes optimal training data and explore methods to achieve more human-like learning efficiency, maximizing knowledge acquisition from each training example.

References

AI@Meta. Llama 3 model card, 2024a. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

AI@Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation, 2024b. URL https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

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. *arXiv preprint arXiv:2305.13245*, 2023.

Anthropic. Claude 3.7 sonnet system card, 2025. URL https://assets.anthropic.com/m/785e231869ea8b3b/original/claude-3-7-sonnet-system-card.pdf.

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *CoRR*, abs/2108.07732, 2021.

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.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7432–7439. AAAI Press, 2020. doi: 10.1609/aaai.v34i05.6239. URL https://doi.org/10.1609/aaai.v34i05.6239.

Andrei Z Broder. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pp. 21–29. IEEE, 1997.

Linzheng Chai, Shukai Liu, Jian Yang, Yuwei Yin, Ke Jin, Jiaheng Liu, Tao Sun, Ge Zhang, Changyu Ren, Hongcheng Guo, Zekun Wang, Boyang Wang, Xianjie Wu, Bing Wang, Tongliang Li, Liqun Yang, Sufeng Duan, and Zhoujun Li. Mceval: Massively multilingual code evaluation, 2024. URL https://arxiv.org/abs/2406.07436.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multilingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021.

作为我们未来工作的一个部分，我们的目标是训练一个更强大的模型。为了在训练和推理效率之间实现最佳平衡，我们计划整合更多高效的架构设计，例如分组查询注意力（GQA）（Ainslie 等，2023）、多头潜在注意力（MLA）（DeepSeek-AI，2024b）和线性注意力（Yang 等，2023；Katharopoulos 等，2020）。此外，我们还打算探索使用更多稀疏的专家混合（MoE）层，以提高计算效率。此外，由于数据是预训练的基础，我们将深化对什么构成最优训练数据的理解，并探索实现更类似人类的学习效率的方法，最大化从每个训练样本中获取的知识。

参考文献

AI@Meta。Llama 3 模型卡，2024a。网址 https://github.com/meta-llama/llama3/blob/main/MODE L CARD.md。

AI@Meta。Llama 4 群：全新多模态AI创新时代的开始，2024b。网址 https://ai.meta.com/blog/llama-4-multimodal-intelligence/。

Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebr ón 和 Sumit Sanghai. Gqa：从多头检查点训练通用多查询变换器模型。*arXiv preprint arXiv:2305.13245*，2023。

Anthropic。Claude 3.7十四行诗系统卡，2025年。网址 https://assets.anthropic.com/m/785e231869ea8b3b/original/claude-3-7-sonnet-system-card.pdf。

雅各布·奥斯汀、奥古斯都·奥德纳、麦克斯韦·I·奈、马尔滕·博斯马、亨利克·米哈莱夫斯基、戴维·多汉、艾伦·姜、凯瑞·J·蔡、迈克尔·特里、阮国·勒、查尔斯·萨顿。使用大型语言模型进行程序合成。*CoRR*，abs/2108.07732，2021。

Adrien Barbaresi。Trafilatura：一个用于文本发现和提取的网页抓取库和命令行工具。在 *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*，第122–131页。计算语言学协会，2021。网址 https://aclanthology.org/2021.acl-demo.15。

伊奥坦·比斯克、罗温·泽勒斯、罗南·勒布拉斯、高建峰、崔叶津。PIQA：关于自然语言中的物理常识推理。在 *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*，第 7432–7439 页。AAAI 出版社，2020。doi: 10.1609/aaai.v34i05.6239。网址 https://doi.org/10.1609/aaai.v34i05.6239。

安德烈·Z·布罗德。关于文档的相似性与包含关系。在 *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*，第21–29页。IEEE，1997。

柴林正，刘书凯，杨健，尹玉薇，金科，刘嘉恒，孙涛，张戈，任长宇，郭宏成，王泽坤，王博洋，吴贤杰，王冰，李通亮，杨立群，段苏峰，李舟俊。Mceval：大规模多语言代码评估，2024。网址 https://arxiv.org/abs/2406.07436。

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, 和 Zheng Liu. Bge m3-embedding：通过自我知识蒸馏实现多语言、多功能、多粒度文本嵌入，2024。

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, 和 Wojciech Zaremba。评估在代码上训练的大型语言模型。*CoRR*，abs/2107.03374，2021。

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *CoRR*, 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. *CoRR*, abs/2110.14168, 2021.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *CoRR*, abs/2401.06066, 2024. URL https://doi.org/10.48550/arXiv.2401.06066.

DeepSeek-AI. Deepseek LLM: scaling open-source language models with longtermism. *CoRR*, abs/2401.02954, 2024a. URL https://doi.org/10.48550/arXiv.2401.02954.

DeepSeek-AI. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *CoRR*, abs/2405.04434, 2024b. URL https://doi.org/10.48550/arXiv.2405.04434.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pp. 7480–7512. PMLR, 2023.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 2368–2378. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1246. URL https://doi.org/10.18653/v1/n19-1246.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv

彼得·克拉克、艾萨克·考维、奥伦·埃齐奥尼、图沙尔·霍特、阿希什·萨巴尔瓦尔、卡丽莎·舍尼克和奥维德·塔弗杰德。你认为你已经解决了问答问题吗？试试 ARC，AI2 推理挑战。*CoRR*，abs/1803.05457，2018。

卡尔·科布、Vineet Kosaraju、Mohammad Bavarian、Mark Chen、Heewoo Jun、Lukasz Kaiser、Matthias Plappert、Jerry Tworek、Jacob Hilton、Reiichiro Nakano、Christopher Hesse 和 John Schulman。训练验证器以解决数学应用题。*CoRR*，abs/2110.14168，2021。

戴大迈, 邓成奇, 赵成刚, 徐瑞祥, 高华佐, 陈德利, 李佳石, 曾旺丁, 于兴凯, 吴Y., 谢振达, 李Y. K., 黄盼盼, 罗福利, 阮冲, 隋志芳, 梁文峰. Deepseekmoe: 迈向混合专家语言模型中的终极专家专业化. *CoRR*, abs/2401.06066, 2024. 网址 https://doi.org/10.48550/arXiv.2401.06066.

DeepSeek-AI。Deepseek LLM：以长远主义为基础扩展开源语言模型。*CoRR*，abs/2401.02954，2024a。网址 https://doi.org/10.48550/arXiv.2401.02954。

DeepSeek-AI。Deepseek-v2：一种强大、经济且高效的专家混合语言模型。*CoRR*，abs/2405.04434，2024b。网址 https://doi.org/10.48550/arXiv.2405.04434。

DeepSeek-AI，刘爱新，冯北，雪冰，王冰轩，吴博超，陆成达，赵成刚，邓成奇，张辰宇，阮冲，戴大迈，郭大雅，杨德建，陈德利，纪东杰，李尔航，林方云，戴福聪，罗福利，郝光博，陈关廷，李国伟，张浩，徐汉伟，丁宏辉，辛华建，高焕佐，李惠，曲惠，蔡金良，梁建，郭建中，倪佳琪，李佳实，王家伟，陈金，陈景昌，袁景阳，邱俊杰，李俊龙，宋俊晓，董凯，胡凯，高凯歌，关康，黄科新，余快，王乐，张乐聪，徐磊，夏乐叶，赵亮，王立通，夏立岳，赵磊，张雷，夏丽，李梦，王苗俊，张明川，张明华，唐明辉，李明明，田宁，黄盼盼，王佩怡，张鹏，王千成，王启浩，朱琪浩，陈沁雨，杜秋石，陈R. J.，金R. L.，葛瑞奇，张瑞松，潘瑞哲，王润基，徐润新，张若瑜，陈如玉，李S. S.，卢尚豪，周尚彦，陈善煌，吴少卿，叶胜峰，叶胜峰，马世荣，王诗宇，周双，余水平，周顺峰，潘书婷，W. L.，云涛，曾天，孙天宇，萧W. L.，曾望丁，赵万佳，安伟，刘文，梁文峰，高文俊，余文钦，张文涛，李X. Q.，金向越，王贤祖，Bi肖，刘晓东，王晓寒，沈晓金，陈晓康，张晓康，陈晓莎，聂晓涛，孙晓文，王晓湘，王新翔，陈新，刘新，谢兴超，刘兴凯，宋新楠，山夏欣，周欣怡，杨新宇，李新远，苏学成，林旭恒，李Y. K.，王Y. Q.，魏Y. X.，朱Y. X.，张杨，徐艳红，徐艳红，黄燕平，李耀，赵耀，孙耀峰，李耀辉，王耀辉，余一，郑一，张一超，孙一翔，山一新，刘一远，郭永强，吴宇，欧元，朱宇辰，王育丹，龚岳，邹宇恒，马玉婷，严燕，徐燕红，罗玉翔，尤玉翔，刘宇轩，周宇阳，吴志峰，任志泽，任泽辉，沙哲，傅哲，徐震，黄振，许振，谢志远，张志远，郝志文，郭志斌，马志刚，严志安，邵志宏，邵志宏，徐志鹏，吴志鹏，李子辉，谢子伟，宋子阳，G. F. Wu，Z. Z. Ren，泽辉任，沙哲力，徐哲，黄振，张振，谢振达，张志远，郝志文，郭志斌，马志刚，严志安，邵志宏，邵志宏，徐志鹏，吴志鹏，李子辉，谢子伟，宋子阳。Deepseek-v3技术报告，2024。网址 https://arxiv.org/abs/2412.19437。

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin 等人。将视觉变换器扩展到 220 亿参数。在 *International Conference on Machine Learning*，第 7480–7512 页。PMLR，2023。

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh 和 Matt Gardner。DROP：一个需要对段落进行离散推理的阅读理解基准。在 Jill Burstein、Christy Doran 和 Thamar Solorio（编辑）中，*Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*，第2368–2378页。计算语言学协会，2019。doi: 10.18653/V1/N19-1246。网址 https://doi.org/10.18653/v1/n19-1246。

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv

Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The Llama 3 herd of models. *CoRR*, abs/2407.21783, 2024.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39, 2022.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile van Krieken, and Pasquale Minervini. Are we done with mmlu? *CoRR*, abs/2406.04127, 2024. URL https://doi.org/10.48550/arXiv.2406.04127.

Zhouhong Gu, Xiaoxuan Zhu, Haoning Ye, Lin Zhang, Jianchen Wang, Yixin Zhu, Sihang Jiang, Zhuozhi Xiong, Zihan Li, Weijie Wu, Qianyu He, Rui Xu, Wenhao Huang, Jingping Liu, Zili Wang, Shusen Wang, Weiguo Zheng, Hongwei Feng, and Yanghua Xiao. Xiezhi: An ever-updating benchmark for holistic domain knowledge evaluation, 2024. URL https://arxiv.org/abs/2306.05783.

Yancheng He, Shilong Li, Jiaheng Liu, Yingshui Tan, Weixun Wang, Hui Huang, Xingyuan Bu, Hangyu Guo, Chengwei Hu, Boren Zheng, et al. Chinese simpleqa: A chinese factuality evaluation for large language models. *arXiv preprint arXiv:2411.07140*, 2024.

Babak Hejazi. Introducing grouped gemm apis in cublas and more performance updates, June 2024. URL https://developer.nvidia.com/blog/introducing-grouped-gemm-apis-in-cublas-and-more-performance-updates/.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *ICLR*. OpenReview.net, 2021a.

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. In *NeurIPS Datasets and Benchmarks*, 2021b.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? *CoRR*, abs/2404.06654, 2024.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zhen Leng Thai, Kai Zhang, 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. *CoRR*, abs/2404.06395, 2024.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems*, 2023.

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.

Ahad Jawaid, Shreeram Suresh Chandra, Junchen Lu, and Berrak Sisman. Style mixture of experts for expressive text-to-speech synthesis. *arXiv preprint arXiv:2406.03637*, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet,

Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, 等等。Llama 3 模型群。*CoRR*，abs/2407.21783，2024。

Yann Dubois、Balázs Galambosi、Percy Liang 和 Tatsunori B Hashimoto。长度控制的 alpacaeval：一种简便的去偏自动评估方法。*arXiv preprint arXiv:2404.04475*，2024年。

William Fedus, Barret Zoph 和 Noam Shazeer。Switch transformers：通过简单高效的稀疏性扩展到万亿参数模型。*J. Mach. Learn. Res.*，23：120：1–120：39，2022。

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile van Krieken, 和 Pasquale Minervini。我们完成了 mmlu 吗？*CoRR*，abs/2406.04127，2024。网址 https://doi.org/10.48550/arXiv.2406.04127。

周宏谷，朱晓轩，叶浩宁，张琳，王建晨，朱一欣，江思航，熊卓志，李子涵，吴伟杰，何倩瑜，徐锐，黄文浩，刘靖平，王子力，王书森，郑维国，冯宏伟，萧杨华。Xiezhi：一个持续更新的整体领域知识评估基准，2024。网址 https://arxiv.org/abs/2306.05783。

盐城何，李世龙，刘嘉恒，谭应水，王伟勋，黄辉，卜星远，郭航宇，胡成伟，郑博仁等。中国简问：一种面向大型语言模型的中文事实性评估。*arXiv preprint arXiv:2411.07140*，2024。

Babak Hejazi。介绍cublas中的分组gemm API及更多性能更新，2024年6月。网址 https://developer.nvidia.com/blog/introducing-grouped-gemm-apis-in-cublas-and-more-perfo rmance-updates/。

丹·亨德里克斯、科林·伯恩斯、史蒂文·巴萨特、安迪·邹、马纳塔斯·马泽伊卡、唐·宋和雅各布·斯坦哈特。衡量大规模多任务语言理解。在 *ICLR*。OpenReview.net，2021a。

丹·亨德里克斯、科林·伯恩斯、索拉夫·卡达瓦斯、阿库尔·阿罗拉、史蒂文·巴萨特、埃里克·唐、唐·宋和雅各布·斯坦哈特。使用 MATH 数据集衡量数学问题解决能力。在 *NeurIPS Datasets and Benchmarks*，2021b。

谢正平、孙思萌、克里曼·塞缪尔、阿查赖·山坦努、雷克什·迪马、贾飞、张阳、金斯堡·鲍里斯。RULER：你的长上下文语言模型的实际上下文大小是多少？*CoRR*，abs/2404.06654，2024。

胡胜鼎、涂宇格、韩旭、何超群、崔甘渠、龙翔、郑志、方叶伟、黄玉香、赵伟林、张新荣、泰曾振、张凯、王崇义、姚远、赵晨阳、周杰、蔡杰、翟忠武、丁宁、贾超、曾国阳、李大海、刘志远、孙茂松。MiniCPM：揭示小型语言模型的潜力与可扩展训练策略。*CoRR*，abs/2404.06395，2024。

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, 和 Junxian He。C-eval：一个面向基础模型的多层次多学科中文评估套件。发表于 *Advances in Neural Information Processing Systems*，2023年。

纳曼·贾因、韩王、阿列克斯·古、李文丁、闫凡佳、张天俊、王思达、阿曼多·索拉尔-莱萨马、森俊希、斯托伊卡离。Livecodebench：对大型代码语言模型的整体且无污染的评估。
arXiv preprint arXiv:2403.07974，2024年。

Ahad Jawaid, Shreeram Suresh Chandra, Junchen Lu 和 Berrak Sisman。用于表达式文本到语音合成的专家风格混合。*arXiv preprint arXiv:2406.03637*，2024。

艾伯特·Q江，亚历山大·萨布莱罗尔斯，安托万·鲁，阿瑟·孟希，布兰奇·萨瓦里，克里斯·班福德，德文德拉·辛格·查普洛特，迭戈·德拉·卡萨斯，艾玛·布哈纳，弗洛里安·布雷桑，吉安娜·伦杰尔，纪尧姆·布尔，纪尧姆·拉姆普勒，Lélio·雷纳尔·拉沃，吕西尔·索尼耶，玛丽-安妮·拉肖，皮埃尔·斯托克，桑迪普·苏布拉马尼安，索菲亚·杨，西蒙·安东尼亚克，特文·勒斯考，泰奥菲尔·热韦。

Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *CoRR*, abs/2401.04088, 2024.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://doi.org/10.1162/tacl_a_00276.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive data sets*. Cambridge university press, 2020.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. CMMLU: Measuring massive multitask language understanding in Chinese. *CoRR*, abs/2306.09212, 2023.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*, 2024.

Xiuhong Li, Yun Liang, Shengen Yan, Liancheng Jia, and Yinghan Li. A coordinated tiling and batching framework for efficient GEMM on gpus. In Jeffrey K. Hollingsworth and Idit Keidar (eds.), *Proceedings of the 24th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPoPP 2019, Washington, DC, USA, February 16-20, 2019*, pp. 229–241. ACM, 2019. doi: 10.1145/3293883.3295734. URL https://doi.org/10.1145/3293883.3295734.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

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. *Advances in Neural Information Processing Systems*, 36:21558–21572, 2023.

Jiawei Liu, Songrun Xie, Junhao Wang, Yuxiang Wei, Yifeng Ding, and Lingming Zhang. Evaluating language models for efficient code generation. *arXiv preprint arXiv:2408.06450*, 2024a.

Junnan Liu, Hongwei Liu, Linchen Xiao, Ziyi Wang, Kuikun Liu, Songyang Gao, Wenwei Zhang, Songyang Zhang, and Kai Chen. Are your llms capable of stable reasoning? *arXiv preprint arXiv:2412.13147*, 2024b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *CoRR*, abs/1711.05101, 2017.

MAA. American invitational mathematics examination - aime. In *American Invitational Mathematics Examination - AIME 2024*, February 2024. URL https://maa.org/math-competitions/american-invitational-mathematics-examination-aime.

James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, volume 5, pp. 281–298. University of California press, 1967.

Thibaut Lavril, Thomas Wang, Timothée Lacroix 以及 William El Sayed。专家的Mixtral。*CoRR*，abs/2401.04088，2024年。Mandar Joshi, Eunsol Choi, Daniel Weld 以及 Luke Zettlemoyer。TriviaQA：一个大规模远程监督的阅读理解挑战数据集。在 Regina Barzilay 和 Min-Yen Kan（编辑）中，*Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*，第1601–1611页，2017年7月，加拿大温哥华。计算语言学协会。doi: 10.18653/v1/P17-1147。网址 https://aclanthology.org/P17-1147。Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou 和 Tomas Mikolov。Fasttext.zip：压缩文本分类模型。*arXiv preprint arXiv:1612.03651*，2016年。Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas 和 François Fleuret。Transformers 是 RNNs：具有线性注意力的快速自回归变换器。在 *International conference on machine learning*，第5156–5165页。PMLR，2020年。Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le 和 Slav Petrov。自然问题：问答研究的基准。*Trans. Assoc. Comput. Linguistics*，第7卷，第452–466页，2019年。doi: 10.1162/tacl_a_00276。网址 https://doi.org/10.1162/tacl a 00276。Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang 和 Ion Stoica。基于分页注意的高效大语言模型服务内存管理。*Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*，2023年。Jure Leskovec, Anand Rajaraman 和 Jeffrey David Ullman。*Mining of massive data sets*。剑桥大学出版社，2020年。Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan 和 Timo- thy Baldwin。CMMLU：衡量中文大规模多任务语言理解的基准。*CoRR*，abs/2306.09212，2023年。Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez 和 Ion Stoica。从众包数据到高质量基准：Arena-hard 和 benchbuilder 流水线。*arXiv preprint arXiv:2406.11939*，2024年。Xiuhong Li, Yun Liang, Shengen Yan, Liancheng Jia 和 Yinghan Li。用于GPU高效GEMM的协调平铺和批处理框架。在 Jeffrey K. Hollingsworth 和 Idit Keidar（编辑），*Proceedings of the 24th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPoPP 2019, Washington, DC, USA, February 16-20, 2019*，第229–241页。ACM，2019年。doi: 10.1145/3293883.3295734。网址 https://doi.org/10.1145/3293883.3295734。Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever 和 Karl Cobbe。让我们逐步验证。在 *The Twelfth International Conference on Learning Representations*，2023年。Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang 和 Lingming Zhang。你的代码由ChatGPT生成真的正确吗？对大型语言模型在代码生成中的严格评估。*Advances in Neural Information Processing Systems*，第36卷，第21558–21572页，2023年。Jiawei Liu, Songrun Xie, Junhao Wang, Yuxiang Wei, Yifeng Ding 和 Lingming Zhang。评估语言模型以实现高效代码生成。*arXiv preprint arXiv:2408.06450*，2024年。Junnan Liu, Hongwei Liu, Linchen Xiao, Ziyi Wang, Kuikun Liu, Songyang Gao, Wenwei Zhang, Songyang Zhang 和 Kai Chen。你的LLMs能稳定推理吗？*arXiv preprint arXiv:2412.13147*，2024年。Ilya Loshchilov 和 Frank Hutter。解耦权重衰减正则化。*CoRR*，abs/1711.05101，2017年。MAA。美国邀请数学考试 - aime。在 *American Invitational Mathematics Examination - AIME 2024*，2024年2月。网址 https://maa.org/math-competitions/american-invitational-mathematics-examination-aime。James MacQueen。多变量观测的分类和分析方法。在 *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*，第5卷，第281–298页。加州大学出版社，1967年。

Mistral. Mistral small 3, 2025. URL https://mistral.ai/news/mistral-small-3.

NVIDIA. pytorch - transformer engine 2.1.0 documentation, August 2024a. URL https://docs.nvidia.com/deeplearning/transformer-engine/user-guide/api/pytorch.html#transformer_engine.pytorch.GroupedLinear.

NVIDIA. MoE A2A Interleaved 1F1B based Computation and Communication Overlap, 2024b. URL https://developer.nvidia.com/zh-cn/blog/1f1b-moe-a2a-computing-overlap/. Accessed: 2025-04-23.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

OpenAI. Introducing SimpleQA, 2024. URL https://openai.com/index/introducing-simpleqa/.

OpenAI. Introducing gpt-4.5, 2025a. URL https://openai.com/index/introducing-gpt-4-5/.

OpenAI. Introducing openai o3 and o4-mini, 2025b. URL https://openai.com/index/introducing-o3-and-o4-mini/.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-1144. URL https://doi.org/10.18653/v1/p16-1144.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *CoRR*, abs/2306.01116, 2023.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Qwen. Qwen2.5: A party of foundation models, 2024a. URL https://qwenlm.github.io/blog/qwen2.5.

Qwen. Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters", February 2024b. URL https://qwenlm.github.io/blog/qwen-moe/.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *CoRR*, abs/2112.11446, 2021.

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. *CoRR*, abs/2311.12022, 2023.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. *arXiv preprint arXiv:2409.16040*, 2024.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. Investigating prior knowledge for challenging chinese machine reading comprehension, 2019.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *ACL (Findings)*, pp. 13003–13051. Association for Computational Linguistics, 2023.

米斯特拉尔。米斯特拉尔小型3，2025年。网址 https://mistral.ai/news/mistral-small-3。英伟达。pytorch - transformer engine 2.1.0 文档，2024年8月a。网址 https://docs.nvidia.com/deeplearning/transformer-engine/user-guide/api/pytorch.html#transformer engine.pytorc h.GroupedLinear。英伟达。MoE A2A 交错1F1B基础的计算与通信重叠，2024b。网址 https://developer.nvidia.com/zh-cn/blog/1f1b-moe-a2a- computing-overlap/。访问日期：2025-04-23。OLMo团队，Pete Walsh，Luca Soldaini，Dirk Groeneveld，Kyle Lo，Shane Arora，Akshita Bhagia，Yuling Gu，Shengyi Huang，Matt Jordan 等。2 olmo 2 furious。
arXiv preprint arXiv:2501.00656，2024。

OpenAI。介绍 SimpleQA，2024。网址 https://openai.com/index/introducing-simpleqa/。

OpenAI. 介绍 gpt-4.5，2025a。网址 https://openai.com/index/introducing-gpt-4-5/。OpenAI。介绍 openai o3 和 o4-mini，2025b。网址 https://openai.com/index/introducing-o 3-and-o4-mini/。Denis Paperno，Germán Kruszewski，Angeliki Lazaridou，Quan Ngoc Pham，Raffaella Bernardi，Sandro Pezzelle，Marco Baroni，Gemma Boleda，和 Raquel Fernández。LAMBADA 数据集：需要广泛话语背景的词预测。在
Proceedings of the 54th Annual Meeting of the Association
for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers。计算语言学协会，2016。doi: 10.18653/v1/p16-1144。网址 https://doi.org/10.1 8653/v1/p16-1144。Guilherme Penedo，Quentin Malartic，Daniel Hesslow，Ruxandra Cojocaru，Alessandro Cappelli，Hamza Alobeidli，Baptiste Pannier，Ebtesam Almazrouei，和 Julien Launay。Falcon LLM 的 RefinedWeb 数据集：用网页数据超越策划语料库，仅用网页数据。*CoRR*，abs/2306.01116，2023。Guilherme Penedo，Hynek Kydl´ček，Loubna Ben allal，Anton Lozhkov，Margaret Mitchell，Colin Raffel，Leandro Von Werra，和 Thomas Wolf。Fineweb 数据集：提取网页，获取规模化的优质文本数据。在
The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track，2024。网址 https://openreview.net/forum?id=n6SCkn2QaG。Qwen。Qwen2.5：一个派对

基础模型，2024a。网址 https://qwenlm.github .io/blog/qwen2.5. Qwen. Qwen1.5-moe：用1/3激活参数实现7b模型性能匹配，2024年2月b。网址 https://qwenlm.github.io/blog/qwen-moe/。Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young 等。扩展语言模型：训练gopher的方法、分析与见解。*CoRR*，abs/2112.11446，2021年。David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael 和 Samuel R. Bowman。GPQA：一个研究生级别的Google-proof问答基准。*CoRR*，abs/2311.12022，2023年。Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Su- sano Pinto, Daniel Keysers 和 Neil Houlsby。利用稀疏专家混合扩展视觉能力。*Advances in Neural Information Processing Systems*，34:8583–8595，2021年。Noam Shazeer。Glu变体改善Transformer。*arXiv preprint arXiv:2002.05202*，2020年。Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen 和 Ming Jin。Time-moe：具有专家混合的十亿规模时间序列基础模型。*arXiv preprint arXiv:2409.16040*，2024年。Kai Sun, Dian Yu, Dong Yu 和 Claire Cardie。研究面向具有挑战性的中文机器阅读理解的先验知识，2019年。Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou 和 Jason Wei。挑战性BIG-Bench任务及链式思考是否能解决它们。在*ACL (Findings)*，第13003–13051页。计算语言学协会，2023年。

Liping Tang, Nikhil Ranjan, Omkar Pangarkar, Xuezhi Liang, Zhen Wang, Li An, Bhaskar Rao, Linghao Jin, Huijuan Wang, Zhoujun Cheng, Suqi Sun, Cun Mu, Victor Miller, Xuezhe Ma, Yue Peng, Zhengzhong Liu, and Eric P. Xing. Txt360: A top-quality llm pre-training dataset requires the perfect blend, 2024.

P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixin Deng, Shawn Gavin, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, Sirun Li, Yizhi Li, Yunwen Li, David Ma, Yuansheng Ni, Haoran Que, Qiyao Wang, Zhoufutu Wen, Siwei Wu, Tyshawn Hsing, Ming Xu, Zhenzhu Yang, Zekun Moore Wang, Junting Zhou, Yuelin Bai, Xingyuan Bu, Chenglin Cai, Liang Chen, Yifan Chen, Chengtuo Cheng, Tianhao Cheng, Keyi Ding, Siming Huang, Yun Huang, Yaoru Li, Yizhe Li, Zhaoqun Li, Tianhao Liang, Chengdong Lin, Hongquan Lin, Yinghao Ma, Tianyang Pang, Zhongyuan Peng, Zifan Peng, Qige Qi, Shi Qiu, Xingwei Qu, Shanghaoran Quan, Yizhou Tan, Zili Wang, Chenqing Wang, Hao Wang, Yiya Wang, Yubo Wang, Jiajun Xu, Kexin Yang, Ruibin Yuan, Yuanhao Yue, Tianyang Zhan, Chun Zhang, Jinyang Zhang, Xiyue Zhang, Xingjian Zhang, Yue Zhang, Yongchi Zhao, Xiangyu Zheng, Chenghua Zhong, Yang Gao, Zhoujun Li, Dayiheng Liu, Qian Liu, Tianyu Liu, Shiwen Ni, Junran Peng, Yujia Qin, Wenbo Su, Guoyin Wang, Shi Wang, Jian Yang, Min Yang, Meng Cao, Xiang Yue, Zhaoxiang Zhang, Wangchunshu Zhou, Jiaheng Liu, Qunshu Lin, Wenhao Huang, and Ge Zhang. Supergpqa: Scaling llm evaluation across 285 graduate disciplines, 2025. URL https://arxiv.org/abs/2502.14739.

Junfeng Tian, Rui Wang, Cong Li, Yudong Zhou, Jun Liu, and Jun Wang. Nyonic technical report. *arXiv preprint arXiv:2404.15702*, 2024a.

Junfeng Tian, Da Zheng, Yang Cheng, Rui Wang, Colin Zhang, and Debing Zhang. Untie the knots: An efficient data augmentation strategy for long-context pre-training in language models, 2024b. URL https://arxiv.org/abs/2409.04774.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *CoRR*, abs/2408.15664, 2024a. URL https://doi.org/10.48550/arXiv.2408.15664.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *CoRR*, abs/2406.01574, 2024b. URL https://doi.org/10.48550/arXiv.2406.01574.

Tianwen Wei, Jian Luan, Wei Liu, Shuang Dong, and Bin Wang. Cmath: Can your language model pass chinese elementary school math test?, 2023.

Mitchell Wortsman, Peter J Liu, Lechao Xiao, Katie Everett, Alex Alemi, Ben Adlam, John D Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, et al. Small-scale proxies for large-scale transformer training instabilities. *arXiv preprint arXiv:2309.14322*, 2023.

xAI. Grok 3 beta — the age of reasoning agents, 2025. URL https://x.ai/news/grok-3.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. CLUE: A chinese language understanding evaluation benchmark. In Donia Scott, Núria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pp. 4762–4772. International Committee on Computational Linguistics, 2020. doi: 10.18653/V1/2020.COLING-MAIN.419. URL https://doi.org/10.18653/v1/2020.coling-main.419.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*, 2023.

Zhiwei Yang, Lu Lu, and Ruimin Wang. A batched GEMM optimization framework for deep learning. *J. Supercomput.*, 78(11):13393–13408, 2022. doi: 10.1007/S11227-022-04336-3. URL https://doi.org/10.1007/s11227-022-04336-3.

Liping Tang, Nikhil Ranjan, Omkar Pangarkar, Xuezhi Liang, Zhen Wang, Li An, Bhaskar Rao, Linghao Jin, Huijuan Wang, Zhoujun Cheng, Suqi Sun, Cun Mu, Victor Miller, Xuezhe Ma, Yue Peng, Zhengzhong Liu, 和 Eric P. Xing. Txt360：一个高质量的大语言模型预训练数据集需要完美的融合，2024。

P 团队，杜新润，姚一凡，马凯靖，王冰力，郑天宇，朱敬，刘明浩，梁一鸣，金晓龙，魏振林，郑楚杰，邓开心，加文肖恩，贾贤，江思超，廖一燕，李瑞，李沁睿，李思润，李一枝，李云文，马大卫，倪元胜，阙浩然，王启尧，文周福图，吴思伟，邢泰顺，徐明，杨振珠，王泽坤，周俊廷，白月林，卜星源，蔡成林，陈亮，陈一凡，程成拓，程天浩，丁科一，黄思明，黄云，李耀如，李一哲，李昭群，梁天浩，林成东，林宏泉，马英豪，庞天阳，彭中远，彭子凡，齐吉格，邱世，屈星伟，权尚浩然，谭一洲，王子立，王晨清，王浩，王一雅，王宇博，许佳骏，杨科新，袁瑞彬，岳元浩，詹天阳，张春，张金阳，张熙越，张星健，张越，赵永驰，郑向宇，钟成华，高杨，李周俊，刘大一恒，刘倩，刘天宇，倪世文，彭俊然，秦雨佳，苏文博，王国英，王世，杨建，杨敏，曹梦，岳翔，张昭翔，周春春，刘家恒，林群书，黄文浩，张戈。Supergpqa：在2025年跨越285个研究生学科的LLM评估扩展，网址https://arxiv.org/abs/2502.14739。

田俊峰、王睿、李聪、周宇东、刘俊、王俊。Nyonic技术报告。*arXiv preprint arXiv:2404.15702*，2024a。

田俊峰，郑达，程阳，王睿，张 Colin，张德兵。解开结：一种用于长上下文预训练的高效数据增强策略，2024b。网址 https://arxiv.org/abs/2409.04774。

阿希什·瓦斯瓦尼（Ashish Vaswani）、诺姆·沙泽尔（Noam Shazeer）、尼基·帕尔马（Niki Parmar）、雅各布·乌斯科雷特（Jakob Uszkoreit）、利昂·琼斯（Llion Jones）、艾丹·N·戈麦斯（Aidan N Gomez）、卢卡什·凯撒（ ukasz Kaiser）和伊利亚·波洛苏金（Illia Polosukhin）。注意力机制就是你所需要的一切。*Advances in neural information processing systems*，2017年30月。
Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, 和 Damai Dai。无辅助损失的专家混合负载均衡策略。*CoRR*，abs/2408.15664，2024a。网址 https://doi.org/10.48550/arXiv.2408.15664。

王昱博, 马学光, 张戈, 倪元胜, Abhranil Chandra, 郭世光, 任伟明, Aaran Arulraj, 何轩, 江子彦, 李天乐, Max Ku, 王凯, Zhuang Zhuang, 樊荣奇, 岳翔, 陈文虎. Mmlu-pro：一个更稳健、更具挑战性的多任务语言理解基准。*CoRR*，abs/2406.01574，2024b。网址 https://doi.org/10.48550/arXiv.2406.01574。

田文伟、简鸾、刘伟、董双、王斌。Cmath：你的语言模型能通过中国小学数学考试吗？，2023。

米切尔·沃茨曼，彼得·J·刘，萧乐超，凯蒂·埃弗雷特，亚历克斯·阿莱米，本·阿德拉姆，约翰·D·科雷斯，伊泽丁·古尔，阿比谢克·库马尔，罗曼·诺瓦克，等。用于大规模变换器训练不稳定性的小规模代理。*arXiv preprint arXiv:2309.14322*，2023。

xAI。Grok 3 测试版——推理代理的时代，2025年。网址 https://x.ai/news/grok-3。

L向旭、海虎、轩炜张、卢李、陈杰曹、玉东李、叶辰徐、凯孙、殿于、聪于、银天、倩倩董、伟唐刘、博石、亦明崔、俊义李、俊增、荣昭王、伟建谢、燕婷李、伊娜帕特森、佐宇天、怡文张、何周、少伟华刘、哲赵、启鹏赵、聪岳、新睿张、正良杨、凯尔·理查森、谭振中。CLUE：一个中文理解评估基准。在多尼亚·斯科特、N úria 贝尔和宗成清（编辑），*Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*，第 4762–4772 页。国际计算语言学委员会，2020年。doi: 10.18653/V1/2020.COLING-MAIN.419。网址 https://doi.org/10.18653/v1/2020.coling-main.419。

安阳，鲍松阳，张贝辰，惠彬源，郑博，余博文，李成远，刘大恒，黄飞，魏浩然 等。Qwen2. 5技术报告。*arXiv preprint arXiv:2412.15115*，2024。

宋林杨，白林王，益康申，拉梅斯瓦尔·潘达，尹金。具有硬件高效训练的门控线性注意力变换器。*arXiv preprint arXiv:2312.06635*，2023。

杨志伟，卢璐，王瑞敏。深度学习的批处理GEMM优化框架。*J. Supercomput.*，78(11)：13393–13408，2022。doi：10.1007/S11227-022-04336-3。网址 https://doi.org/10.1007/s11227-022-04336-3。

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1472. URL https://doi.org/10.18653/v1/p19-1472.

Yaqing Zhang, Yaobin Wang, Zhangbin Mo, Yong Zhou, Tao Sun, Guang Xu, Chaojun Xing, and Liang Yang. Accelerating small matrix multiplications by adaptive batching strategy on GPU. In *24th IEEE Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application, HPCC/DSS/SmartCity/DependSys 2022, Hainan, China, December 18-20, 2022*, pp. 882–887. IEEE, 2022. doi: 10.1109/HPCC-DSS-SMARTCITY-DEPENDSYS57074.2022.00143. URL https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys57074.2022.00143.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. AGIEval: A human-centric benchmark for evaluating foundation models. *CoRR*, abs/2304.06364, 2023. doi: 10.48550/arXiv.2304.06364. URL https://doi.org/10.48550/arXiv.2304.06364.

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.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen Gong, Thong Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, Prateek Yadav, Naman Jain, Alex Gu, Zhoujun Cheng, Jiawei Liu, Qian Liu, Zijian Wang, Binyuan Hui, Niklas Muennighoff, David Lo, Daniel Fried, Xiaoning Du, Harm de Vries, and Leandro Von Werra. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions, 2025. URL https://arxiv.org/abs/2406.15877.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi 和 Yejin Choi. HellaSwag：机器真的能完成你的句子吗？收录于 Anna Korhonen、David R. Traum 和 Llu´s Màrquez（编）. *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*，第 4791–4800 页。计算语言学协会，2019 年。doi: 10.18653/v1/p19-1472。网址 https://doi.org/10.18653/v1/p19-1472。

Y张阿清，王耀彬，莫章斌，周勇，孙涛，徐光，邢超军，杨亮。在 *24th IEEE Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application, HPCC/DSS/SmartCity/DependSys 2022, Hainan, China, December 18-20, 2022* 中，页码 882–887。IEEE，2022。doi: 10.1109/HPCC-DSS-SMARTCITY-DEPENDSYS57074.2022.00143。网址 https://doi.org/10.110 9/HPCC-DSS-SmartCity-DependSys57074.2022.00143。

钟万俊、崔瑞翔、郭一多、梁耀博、卢帅、王艳琳、阿明·赛义德、陈伟柱、段楠。AGIEval：一个以人为本的基础模型评估基准。*CoRR*，abs/2304.06364，2023。doi：10.48550/arXiv.2304.06364。网址 https://doi.org/10.48550/arXiv.2304.06364。

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, 和 Le Hou。大型语言模型的指令遵循评估。*arXiv preprint arXiv:2311.07911*，2023。

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen Gong, Thong Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, Prateek Yadav, Naman Jain, Alex Gu, Zhoujun Cheng, Jiawei Liu, Qian Liu, Zijian Wang, Binyuan Hui, Niklas Muennighoff, David Lo, Daniel Fried, Xiaoning Du, Harm de Vries, 和 Leandro Von Werra。Big-codebench：具有多样函数调用和复杂指令的代码生成基准测试，2025。网址 https://arxiv.org/abs/2406.15877。

Appendix

A Authors

Bi Huo
Bin Tu
Cheng Qin
Da Zheng
Debing Zhang
Dongjie Zhang
En Li
Fu Guo
Jian Yao
Jie Lou
Junfeng Tian
Li Hu
Ran Zhu
Shengdong Chen
Shuo Liu
Su Guang
Te Wo
Weijun Zhang
Xiaoming Shi
Xinxin Peng
Xing Wu
Yawen Liu
Yuqiu Ji
Ze Wen
Zhenhai Liu
Zichao Li
Zilong Liao

Authors are listed alphabetically by the first name. We would like to thank Colin Zhang, Rui Wang, Xing Yu, Di Feng, and Lei Zhang for their support and helpful discussions. We also thank everyone who has contributed to `dots.llm1` but is not mentioned in this paper.

B Hyperparameters

We display the pretraining hyperparameter configuration in Table 6 comparing with other relevant models.

Table 6: Architectural and pretraining hyper-parameters of `dots.llm1` compared with recent models. WSD = weight-stable decay (Hu et al., 2024).

	Qwen2.5 72B	**DeepSeek-V2**	**DeepSeek-V3**	**`dots.llm1`**
Architecture	Dense	MoE	MoE	MoE
Active parameters	72B	21B	37B	14B
Total parameters	72B	236B	671B	142B
Hidden size	8,192	5,120	7,168	4,096
Activation	SwiGLU	SwiGLU	SwiGLU	SwiGLU
FFN size	29,568	12,288	18,432	10,944
Vocabulary size	152,064	102,400	129,280	152,064
Attention type	GQA	MLA	MLA	MHA
Attention heads	64	128	128	32
KV heads	8	128	128	32
Layers	80	60	61	62
Attention biases	Yes	No	No	No
QK normalization	No	No	No	Yes
Positional encoding	RoPE	RoPE	RoPE	RoPE
MoE FFN size	–	1,536	2,048	1,408
Routed experts	–	160	256	128
Shared experts	–	2	1	2
Top-k experts	–	6	8	6
Init. std	–	0.006	0.006	0.006
LR schedule	–	WSD	WSD	WSD
Warmup steps	–	2,000	2,000	4,000
Peak LR	–	2.4E-4	2.2E-4	3.0E-4
Optimizer	–	AdamW	AdamW	AdamW
Weight decay	–	0.1	0.1	0.1
Beta1	–	0.9	0.9	0.9
Beta2	–	0.95	0.95	0.95
Pretraining tokens	18T	8.1T	14.8T	11.2T
Sequence length	4,096	4,096	4,096	8,192

附录

A 作者

Bi Huo
Bin Tu
Cheng Qin
Da Zheng
Debing Zhang
Dongjie Zhang
En Li
Fu Guo
Jian Yao
Jie Lou
Junfeng Tian
Li Hu
Ran Zhu
Shengdong Chen
Shuo Liu
Su Guang
Te Wo
Weijun Zhang
Xiaoming Shi
Xinxin Peng
Xing Wu
Yawen Liu
Yuqiu Ji
Ze Wen
Zhenhai Liu
Zichao Li
Zilong Liao

作者按名字的字母顺序排列。我们感谢Colin Zhang、Rui Wang、Xing Yu、Di Feng和Lei Zhang的支持和宝贵讨论。我们也感谢所有为dots.llm1做出贡献但未在本文中提及的人员。

B 超参数

我们在表6中展示了预训练超参数配置，并与其他相关模型进行了比较。

表6：dots.llm1 与近期模型的架构和预训练超参数对比。WSD = 权重稳定衰减（Hu 等，2024）。

	Qwen2.5 72B	**DeepSeek-V2**	**DeepSeek-V3**	**dots.llm1**
Architecture	Dense	MoE	MoE	MoE
Active parameters	72B	21B	37B	14B
Total parameters	72B	236B	671B	142B
Hidden size	8,192	5,120	7,168	4,096
Activation	SwiGLU	SwiGLU	SwiGLU	SwiGLU
FFN size	29,568	12,288	18,432	10,944
Vocabulary size	152,064	102,400	129,280	152,064
Attention type	GQA	MLA	MLA	MHA
Attention heads	64	128	128	32
KV heads	8	128	128	32
Layers	80	60	61	62
Attention biases	Yes	No	No	No
QK normalization	No	No	No	Yes
Positional encoding	RoPE	RoPE	RoPE	RoPE
MoE FFN size	–	1,536	2,048	1,408
Routed experts	–	160	256	128
Shared experts	–	2	1	2
Top-k experts	–	6	8	6
Init. std	–	0.006	0.006	0.006
LR schedule	–	WSD	WSD	WSD
Warmup steps	–	2,000	2,000	4,000
Peak LR	–	2.4E-4	2.2E-4	3.0E-4
Optimizer	–	AdamW	AdamW	AdamW
Weight decay	–	0.1	0.1	0.1
Beta1	–	0.9	0.9	0.9
Beta2	–	0.95	0.95	0.95
Pretraining tokens	18T	8.1T	14.8T	11.2T
Sequence length	4,096	4,096	4,096	8,192

C Web Data Curation

C.1 Document Preparation Phase

URL Filtering Similar to RefineWeb (Penedo et al., 2023), URL filtering focuses on filtering toxic documents based solely on the URL, rather than the document content. We maintain a blocklist of domains, implementing careful manual verification to ensure removal of pages associated with adult themes, gambling, and other toxic topics.

Text Extraction Following RefineWeb (Penedo et al., 2023), we employ *trafilatura*(Barbaresi, 2021) to extract the main text from webpages. To mitigate the presence of irrelevant content, we implement a series of custom optimizations for *trafilatura*, including adjustments to HTML patterns, keyword filtering, and content length specifications.

Language Identification Leveraging the fastText language identification model from CCNet (Joulin et al., 2016), we classify the language of each document efficiently. To ensure the quality and relevance of the dataset, we discard any documents with a language classification confidence score below a threshold of 0.65. This step is crucial for filtering out noisy or ambiguous data, particularly in multilingual datasets, and helps maintain a consistent and high-quality corpus.

Identity Removal To ensure unique identification, we compute a 32-character hexadecimal string for each document using the MD5 digest algorithm. Duplicate documents are then removed randomly to retain only unique content for further processing. Prior to deduplication, we preprocess the text by removing all punctuation, normalizing characters to the NFD Unicode format[3], converting text to lowercase, and eliminating extra spaces between words. This step is essential for preserving dataset integrity and ensuring that only distinct documents are included in the final collection.

C.2 Rule-Based Processing Phase

Line-wise Inter-Document Deduplication We introduce a line-level deduplication method specifically designed to target repetitive patterns commonly found in the head and tail sections of web documents, such as advertisements, navigation bars, and other non-informative content. This approach effectively reduces inter-document redundancy while preserving the essential and meaningful content, ensuring a cleaner and more focused dataset. First, we extract the first 5 lines and the last 5 lines of each document, splitting them into individual lines while discarding any lines that are empty or contain only whitespace or symbols. Next, we calculate the frequency of each line across the entire dataset. For any line that appears more than 200 times, we retain only its first 200 occurrences within the dataset and remove all additional instances from their respective documents. If a document does not contain any lines that need to be removed, it is preserved in its original form. Otherwise, the specified lines are removed, and the remaining parts are concatenated to form the final document. This method is implemented in an incremental manner and within distributed systems, making it more efficient and scalable for large-scale datasets.

Rule-Based Filtering We develop a filtering system based on RefinedWeb (Penedo et al., 2023) and Gopher (Rae et al., 2021), combining precise heuristic rules and statistical features to systematically remove low-quality content. Our pipeline includes empty content removal, advertisement and registration prompt filtering, domain/URL/title-based meta filtering, wiki/code diff detection, structural anomaly and duplicate content elimination, and content quality filtering. Additionally, we perform manual review and tailored cleaning for major domains, focusing especially on the top 1,000 sites that comprise 60% of the data, to ensure consistently high data quality.

Fuzzy Deduplication We employ MinHash (Broder, 1997) and Locality-Sensitive Hashing (LSH) (Leskovec et al., 2020) to perform approximate deduplication. The process involves the following steps: First, we apply the same text standardization as in the Identity Removal step. Next, we tokenize the text using the Jieba tokenizer [4] to handle Chinese text, followed by 5-gram processing. Using the resulting 5-gram tokens, we compute 2048 MinHash values for each text. These MinHash values are then divided into 128 bands, each containing 16 rows, as part of the LSH configuration. For any samples that collided within the same band, we retain only one instance. This approach enables us to efficiently deduplicate text pairs with a Jaccard similarity of 80%, achieving a high probability of 97.42%.

[3] https://en.wikipedia.org/wiki/Unicode_equivalence

[4] https://github.com/fxsjy/jieba

C Web 数据整理

C.1 文档准备阶段

URL 过滤 类似于 RefineWeb (Penedo et al., 2023)，URL 过滤专注于仅根据 URL 过滤有害文档，而不是文档内容。我们维护一个域名黑名单，进行仔细的人工验证，以确保删除与成人主题、赌博和其他有害话题相关的页面。

文本提取 根据RefineWeb（Penedo等人，2023），我们采用*trafilatura*(Barbaresi，2021)从网页中提取主要文本。为了减少无关内容的出现，我们为*trafilatura*实施了一系列定制优化，包括对HTML模式的调整、关键词过滤和内容长度的规范。

语言识别 利用来自 CCNet（Joulin 等，2016）的 fastText 语言识别模型，我们高效地对每个文档的语言进行分类。为了确保数据集的质量和相关性，我们会丢弃任何语言分类置信度低于阈值 0.65 的文档。这一步对于过滤噪声或模糊的数据尤为重要，特别是在多语言数据集中，有助于保持一致且高质量的语料库。

身份去除 为了确保唯一识别，我们使用MD5摘要算法为每个文档计算一个32字符的十六进制字符串。然后随机删除重复的文档，只保留唯一内容以供后续处理。在去重之前，我们对文本进行预处理，包括删除所有标点符号、将字符规范化为NFD Unicode格式[3]、将文本转换为小写，以及消除单词之间的多余空格。这一步对于保持数据集的完整性以及确保最终集合中只包含不同的文档至关重要。

C.2 基于规则的处理阶段

逐行跨文档去重　我们引入了一种逐行级别的去重方法，专门针对网页文档头部和尾部常见的重复模式，例如广告、导航栏和其他非信息性内容。这种方法有效减少了文档之间的冗余，同时保留了核心且有意义的内容，确保数据集更加整洁和专注。首先，我们提取每个文档的前5行和后5行，将它们拆分成单独的行，同时丢弃任何空行或只包含空白符或符号的行。接下来，我们计算每一行在整个数据集中的出现频率。对于任何出现超过{v*}次的行，我们只保留其在数据集中的前{v*}次出现，并删除其余的实例。如果某个文档中没有需要删除的行，则保持其原始状态。否则，删除指定的行，并将剩余部分拼接成最终的文档。这种方法以增量方式在分布式系统中实现，使其在大规模数据集上更高效、更具扩展性。

基于规则的过滤我们开发了一个基于RefinedWeb（Penedo等，2023）和Gopher（Rae等，2021）的过滤系统，结合了精确的启发式规则和统计特征，以系统性地去除低质量内容。我们的流程包括空内容删除、广告和注册提示过滤、基于域名/URL/标题的元数据过滤、维基/代码差异检测、结构异常和重复内容消除，以及内容质量过滤。此外，我们还对主要域名进行人工审核和定制清理，特别关注占据数据60%的前1,000个网站，以确保数据质量始终如一。

模糊去重我们采用MinHash（Broder，1997）和局部敏感哈希（LSH）（Leskovec等，2020）进行近似去重。该过程包括以下步骤：首先，我们对文本进行与身份去除步骤相同的标准化处理。接下来，使用jieba分词器[4]对文本进行分词，以处理中文文本，然后进行5-gram处理。利用得到的5-gram分词，我们为每个文本计算2048个MinHash值。这些MinHash值随后被划分为128个带，每个带包含16行，作为LSH配置的一部分。对于在同一带内发生碰撞的样本，我们只保留一个实例。这种方法使我们能够高效地去重Jaccard相似度为80%的文本对，并实现97.42%的高概率。

[3] https://en.wikipedia.org/wiki/Unicode_equivalence
[4] https://github.com/fxsjy/jieba

C.3 Model-Based Processing Phase

Web-Type Classification Model Similar to how humans perceive and categorize information, we implement a web content classifier that distinguishes various types of web pages. The classifier utilizes a 1.5B model that categorizes content into text-rich detail pages (such as articles) and non-essential web pages, including tool pages (e.g., maps, public transit route queries), audio pages, video pages, forum sites, and adult websites. Through this process, we selectively preserve only high-quality detail pages for subsequent processing.

Web Clutter Removal Model We implement a line-wise web clutter removal model to eliminate unnecessary web page elements such as borders, advertisements, navigational components, and repetitive content, focusing exclusively on main content. This model evaluates each individual line based on its informational value, relevance, and quality, assigning a score from 0 to 1. We fine-tune a 1.5B model and find that it significantly improves the overall data quality.

Quality Model The quality model performs comprehensive multidimensional analysis to evaluate and score training samples (Qwen, 2024a; Penedo et al., 2024). We design a comprehensive annotation specification to evaluate text quality, including text fluency, coherence, information redundancy, and repetition patterns. In order to balance efficiency and model performance, we adopt a k-fold cross validation method with a 1.5B model and evaluate its discriminative ability using the area under the ROC curve (AUC). Using the quality score generated by this model, we set a threshold to only retain high-quality data from the corpus.

Semantic Deduplication In line with Dubey et al. (2024), we implement a semantic deduplication strategy to eliminate documents with high semantic overlap. Initially, we employ `BGE-M3` (Chen et al., 2024) as an embedding model to create embeddings for each document. Subsequently, we apply the KMeans algorithm (MacQueen, 1967) to categorize these embeddings into clusters. This clustering step significantly reduces computational costs in subsequent pairwise similarity calculations. Following the clustering phase, we compute the pairwise cosine similarity between document embeddings within the same cluster. Duplicates are identified based on a predefined similarity threshold, which is set to 0.95 in our process. By doing so, we ensure the preservation of varied and distinctive content while discarding redundant semantic information.

Category Balancing To curate a more balanced and information-rich training dataset, we first develop a fine-grained, 200-class content classifier to categorize the vast web data. This classifier enables us to identify and quantify the distribution of various content types within the corpus. Based on this categorization, we proportionally increase the representation of knowledge-based and factual content, such as encyclopedia entries, educational materials, and popular science articles. Meanwhile, we deliberately reduce the proportion of less informative or structurally formulaic content, including fiction (e.g., science fiction novels) and product descriptions. This balancing process ensures that the training dataset is both diverse and skewed towards sources that are rich in reliable information, thereby fostering the development of language models with stronger comprehension, reasoning, and knowledge retention abilities.

C.3 基于模型的处理阶段

Web类型分类模型 类似于人类感知和分类信息的方式，我们实现了一个网页内容分类器，用于区分各种类型的网页。该分类器采用一个拥有15亿参数的模型，将内容划分为以文本为主的详细页面（如文章）和非必要网页，包括工具页面（例如地图、公共交通路线查询）、音频页面、视频页面、论坛网站和成人网站。通过这个过程，我们有选择地只保留高质量的详细页面以供后续处理。

网页杂乱去除模型　我们实现了一种逐行的网页杂乱去除模型，旨在消除不必要的网页元素，如边框、广告、导航组件和重复内容，专注于主要内容。该模型根据每一行的信息价值、相关性和质量进行评估，赋予其0到1的分数。我们对一个1.5B模型进行了微调，发现它显著提升了整体数据质量。

质量模型 质量模型执行全面的多维分析，以评估和评分训练样本（Qwen, 2024a；Penedo 等, 2024）。我们设计了一个全面的注释规范，用于评估文本质量，包括文本流畅性、一致性、信息冗余和重复模式。为了在效率和模型性能之间取得平衡，我们采用了带有1.5B模型的k折交叉验证方法，并使用ROC曲线下面积（AUC）评估其判别能力。利用该模型生成的质量分数，我们设定阈值，仅保留语料库中的高质量数据。

语义去重　根据Dubey 等人（2024）的研究，我们实施了一种语义去重策略，以消除具有高度语义重叠的文档。最初，我们采用 BGE-M3（Chen 等人，2024）作为嵌入模型，为每个文档生成嵌入向量。随后，我们应用 KMeans 算法（MacQueen，1967）将这些嵌入向量分类到不同的簇中。这一步聚类显著降低了后续成对相似度计算的计算成本。在聚类阶段之后，我们计算同一簇内文档嵌入向量之间的成对余弦相似度。基于预设的相似度阈值（在我们的流程中设为 0.95），识别重复项。通过这种方式，我们确保保留多样且具有特色的内容，同时剔除冗余的语义信息。

类别平衡 为了构建一个更平衡且信息丰富的训练数据集，我们首先开发了一个细粒度的200类内容分类器，用于对海量网页数据进行分类。该分类器使我们能够识别并量化语料库中各种内容类型的分布。基于此分类，我们按比例增加知识型和事实性内容的代表性，例如百科全书条目、教育资料和科普文章。同时，我们有意减少信息量较少或结构上公式化的内容比例，包括小说（例如科幻小说）和产品描述。这一平衡过程确保训练数据集既多样化，又偏向于富含可靠信息的来源，从而促进具有更强理解、推理和知识保持能力的语言模型的发展。