英文版：https://ai.meta.com/research/publications/the-llama-3-herd-of-models/

中文版：https://www.aisharenet.com/llama-3yigeduoa/

摘要:

本文介绍了一系列新的基础模型，称为 Llama 3。Llama 3 是一个语言模型群体，天生支持多语言、代码编写、推理和工具使用。我们最大的模型是一个具有 4050 亿个参数和高达 128,000 个标记的上下文窗口的密集型 Transformer。本文对 Llama 3 进行了一系列广泛的经验评估。结果表明，Llama 3 在许多任务上都能够达到与 GPT-4 等领先语言模型相当的质量。我们将 Llama 3 公开发布，包括预训练和后训练的 4050 亿参数语言模型，以及用于输入输出安全性的 Llama Guard 3 模型。本文还介绍了将图像、视频和语音功能通过组合式方法集成到 Llama 3 中的实验结果。我们观察到这种方法在图像、视频和语音识别任务上与最先进的方法具有竞争力。由于这些模型仍处于开发阶段，因此尚未广泛发布。

1 引言

基础模型是语言、视觉、语音和其他模态的一般性模型，旨在支持广泛的 AI 任务。它们构成了许多现代 AI 系统的基础。

现代基础模型的开发主要分为两个阶段：

(1) 预训练阶段: 模型在海量数据上进行训练，使用简单任务如单词预测或图注生成；

(2) 后训练阶段: 模型被微调以遵循指令、与人类偏好保持一致，并提高特定能力（例如，编码和推理）。

本文介绍了一套新的语言基础模型，称为 Llama 3。Llama 3 Herd 模型系列天生支持多语言、编码、推理和工具使用。我们最大的模型是具有 405B 参数的稠密 Transformer，能够在高达 128K 个标记的上下文窗口中处理信息。

表 1 列出了羊群中的每个成员。本文中呈现的所有结果均基于 Llama 3.1 模型（简称 Llama 3）。

我们认为开发高质量基础模型的三把关键利器是数据、规模和复杂性管理。在我们的开发过程中，我们将努力优化这三个方面：

- **数据:** 与之前的 Llama 版本（Touvron 等人，2023a, b）相比，我们在预训练和后训练中使用的 Both the quantity and quality of data were improved. These improvements include the development of more careful pre-processing and curation pipelines for pre-training data and the development of more

rigorous quality assurance and filtering approaches for post-training data. Llama 3 在约 15T 多语言标记的语料库上进行预训练，而 Llama 2 则是在 1.8T 标记上进行预训练。

- **规模。**我们训练了一个比之前 Llama 模型更大的模型 :我们的旗舰语言模型使用了 3.8×10^{25} FLOPs 进行预训练，比 Llama 2 最大的版本多近 50 倍。具体来说，我们在 15.6T 文本标记上预训练了一个拥有 405B 可训练参数的旗舰模型。正如预期的那样，

	Finetuned	Multilingual	Long context	Tool use	Release
Llama 3 8B	✗	✗[1]	✗	✗	April 2024
Llama 3 8B Instruct	✓	✗	✗	✗	April 2024
Llama 3 70B	✗	✗[1]	✗	✗	April 2024
Llama 3 70B Instruct	✓	✗	✗	✗	April 2024
Llama 3.1 8B	✗	✓	✓	✗	July 2024
Llama 3.1 8B Instruct	✓	✓	✓	✓	July 2024
Llama 3.1 70B	✗	✓	✓	✗	July 2024
Llama 3.1 70B Instruct	✓	✓	✓	✓	July 2024
Llama 3.1 405B	✗	✓	✓	✗	July 2024
Llama 3.1 405B Instruct	✓	✓	✓	✓	July 2024

Table 1 Overview of the Llama 3 Herd of models. All results in this paper are for the Llama 3.1 models.

- **管理复杂性。** 我们做出的设计选择旨在最大限度地提高模型开发过程的可扩展性。例如，我们选择了标准密集 Transformer 模型架构（Vaswani 等人，2017 年）并进行了一些微小的调整，而不是采用专家混合模型（Shazeer 等人，2017 年）以最大程度地提高训练稳定性。同样，我们采用了一种相对简单的基于监督微调 (SFT)、拒绝采样 (RS) 和直接偏好优化 (DPO；Rafailov 等人 (2023)) 的后处理程序，而不是更复杂的强化学习算法（Ouyang

等人，2022 年；Schulman 等人，2017 年），这些算法往往不太稳定并且难以扩展。

我们这项工作的成果是 Llama 3：一个包含 8B、70B 和 405B 参数的三种多语言[1]语言模型的群体。我们在大量基准数据集上评估了 Llama 3 的性能，这些数据集涵盖了广泛的语言理解任务。此外，我们还进行了广泛的人工评估，将 Llama 3 与竞争模型进行了比较。表 2 展示了旗舰 Llama 3 模型在关键基准测试中的性能概述。我们的实验评估表明，我们的旗舰模型在各种任务上都与 GPT-4（OpenAI，2023a）等领先语言模型持平，并且接近于最先进水平。我们较小的模型是同类中最好的，其性能优于具有相似参数数量的其他模型（Bai 等人，2023 年；Jiang 等人，2023 年）。Llama 3 在帮助性和无害性之间也取得了比其前身（Touvron 等人，2023b）更好的平衡。我们在第 5.4 节中详细分析了 Llama 3 的安全性。

我们将所有三个 Llama 3 模型公开发布，使用 Llama 3 社区许可证的更新版本；请参阅 https://llama.meta.com。这包括我们 405B 参数语言模型的预训练和后处理版本，以及一个新的 Llama Guard 模型版本（Inan 等人，2023 年），用于输入和输出安全性。我们希望公开发布一个旗舰模型能够激发研究界的创新浪潮，并加速朝着人工智能（AGI）的负责任发展方向前进。

多语言：指的是模型能够理解和生成多种语言的文本。

在 Llama 3 的开发过程中，我们还开发了模型的多模态扩展，使其能够具备图像识别、视频识别和语音理解能力。这些模型仍处于积极开发阶段，尚未准备好发布。除了我们的语言建模结果之外，本文还介绍了我们对这些多模态模型的初步实验结果。

Llama 3 8B 和 70B 是在多语言数据上预训练的，但当时主要用于英语。

Category	Benchmark	Llama 3 8B	Gemma 2 9B	Mistral 7B	Llama 3 70B	Mixtral 8x22B	GPT 3.5 Turbo	Llama 3 405B	Nemotron 4 340B	GPT-4 (0125)	GPT-4o	Claude 3.5 Sonnet
General	MMLU (5-shot)	69.4	**72.3**	61.1	**83.6**	76.9	70.7	87.3	82.6	85.1	89.1	**89.9**
	MMLU (0-shot, CoT)	**73.0**	72.3△	60.5	**86.0**	79.9	69.8	88.6	78.7◁	85.4	**88.7**	88.3
	MMLU-Pro (5-shot, CoT)	**48.3**	–	36.9	**66.4**	56.3	49.2	73.3	62.7	64.8	74.0	**77.0**
	IFEval	**80.4**	73.6	57.6	**87.5**	72.7	69.9	**88.6**	85.1	84.3	85.6	88.0
Code	HumanEval (0-shot)	**72.6**	54.3	40.2	**80.5**	75.6	68.0	89.0	73.2	86.6	90.2	**92.0**
	MBPP EvalPlus (0-shot)	**72.8**	71.7	49.5	**86.0**	78.6	82.0	88.6	72.8	83.6	87.8	**90.5**
Math	GSM8K (8-shot, CoT)	**84.5**	76.7	53.2	**95.1**	88.2	81.6	**96.8**	92.3◇	94.2	96.1	96.4◇
	MATH (0-shot, CoT)	**51.9**	44.3	13.0	**68.0**	54.1	43.1	73.8	41.1	64.5	**76.6**	71.1
Reasoning	ARC Challenge (0-shot)	83.4	**87.6**	74.2	**94.8**	88.7	83.7	**96.9**	94.6	96.4	96.7	96.7
	GPQA (0-shot, CoT)	32.8	–	28.8	**46.7**	33.3	30.8	51.1	–	41.4	53.6	**59.4**
Tool use	BFCL	**76.1**	–	60.4	84.8	–	**85.9**	88.5	86.5	88.3	80.5	**90.2**
	Nexus	**38.5**	30.0	24.7	**56.7**	48.5	37.2	**58.7**	–	50.3	56.1	45.7
Long context	ZeroSCROLLS/QuALITY	81.0	–	–	90.5	–	–	**95.2**	–	**95.2**	90.5	90.5
	InfiniteBench/En.MC	65.1	–	–	78.2	–	–	**83.4**	–	72.1	82.5	–
	NIH/Multi-needle	98.8	–	–	97.5	–	–	98.1	–	**100.0**	**100.0**	90.8
Multilingual	MGSM (0-shot, CoT)	**68.9**	53.2	29.9	**86.9**	71.1	51.4	**91.6**	–	85.9	90.5	**91.6**

Table 2 Performance of finetuned Llama 3 models on key benchmark evaluations. The table compares the performance of the 8B, 70B, and 405B versions of Llama 3 with that of competing models. We **boldface** the best-performing model in each of three model-size equivalence classes. △Results obtained using 5-shot prompting (no CoT). ◁Results obtained without CoT. ◇Results obtained using zero-shot prompting.

2 总述

Llama 3 模型架构如图 1 所示。我们 Llama 3 语言模型的开发主要分为两个阶段：

- **语言模型预训练。**我们首先将大型多语言文本语料转换为离散标记，并在由此产生的数据上预训练一个大型语言模型（LLM）进行下一个标记预测。在语言模型预训练阶段，模型学习语言结构并从它“阅读”的文本中获取大量关于世界的知识。为了有效地做到这一点，预训练是在大规模下进行的：我们使用 8K 个标记的上下文窗口，在一个包含 15.6T 个标记的模型上预训练了一个具有 405B 个参数的模型。此标准预训练阶段之后是继续预训练阶段，该阶段将支持的上下文窗口增加到 128K 个标记。有关详细信息，请参见第 3 节。
- **模型后训练。**预训练语言模型对语言有丰富的理解，但它尚未遵循指令或像我们期望的助手那样行为。我们在几轮中通过人类反馈对模型进行校准，每轮都包括在指令调整数据上进行监督微调（SFT）和直接偏好优化（DPO；Rafailov et al., 2024）。在这个后训练阶段，我们还集成了新的功能，例如工具使用，并在编码和推理等方面观察到显著改进。有关详细信息，请参阅第 4 节。最后，安全缓解措施也在后训练阶段集成到模型中，其细节在第 5.4 节中描述。生成的模型具备丰富的功能。它们能够用至少八种语言回答问题，编写高质量代码，解决复杂的推理问题，并开箱即用或零样本方式使用工具。

我们还进行实验，通过组合式方法将图像、视频和语音能力添加到 Llama 3 中。我们研究的方法包括图 28 所示的三个额外阶段：

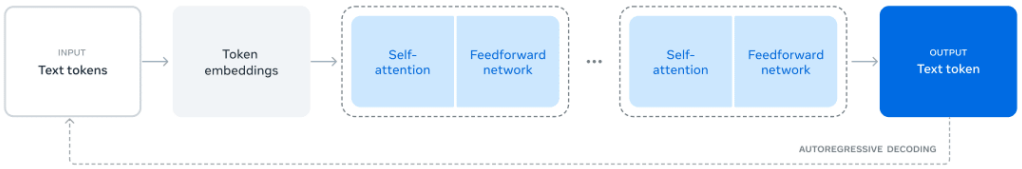

Figure 1 Illustration of the overall architecture and training of Llama 3. Llama 3 is a Transformer language model trained to predict the next token of a textual sequence. See text for details.

- **多模态编码器预训练。**我们为图像和语音分别训练编码器。我们将图像编码器训练于大量图像文本对上。这使得模型学习到视觉内容与其自然语言描述之间的关系。我们的语音编码器使用一种自监督方法，它遮盖语音输入的一部分，并尝试通过离散标记表示重建遮盖的部分。因此，模型学习了语音信号的结构。有关图像编码器的详细信息，请参阅第 7 节；有关语音编码器的详细信息，请参阅第 8 节。
- **视觉适配器训练。**我们训练一个适配器，将预训练的图像编码器与预训练的语言模型集成。该适配器由一系列交叉注意力层组成，将图像编码器表示输入到语言模型中。适配器在文本-图像对上进行训练，这使图像表示与语言表示相一致。在适配器训练期间，我们也更新了图像编码器的参数，但有意不更新语言模型的参数。我们还在图像适配器之上训练一个视频适配器，使用配对的视频-文本数据。这使得模型能够跨帧聚合信息。有关详细信息，请参见第 7 节。

- 最后，我们将语音编码器通过一个适配器集成到模型中，该适配器将语音编码转换为可以直接输入微调语言模型的标记表示。在监督式微调阶段，联合更新适配器和编码器的参数，以实现高质量的语音理解。我们在语音适配器训练过程中不改变语言模型。我们还集成了一个文本到语音系统。详情请参见第 8 节。

我们的多模态实验导致模型能够识别图像和视频的内容，并通过语音接口支持交互。这些模型仍在开发中，尚未准备好发布。

3 预训练

语言模型的预训练涉及以下几个方面：

(1) 收集和过滤大规模训练语料库；

(2) 开发模型架构和相应的缩放定律以确定模型大小；

(3) 开发高效大规模预训练的技术；

(4) 开发预训练方案。我们将在下面分别介绍这些组成部分。

3.1 预训练数据

我们从各种数据源创建语言模型预训练数据集，这些数据源包含截至 2023 年底的知识。我们对每个数据源应用了几种去重方法和数据清洗机制，以获得高质量的标记。我们删除包含大量个人身份信息（PII）的领域，以及已知含有成人内容的领域。

3.11 Web 数据清洗

我们利用的大部分数据来自网络，下面我们将描述我们的清洗流程。

PII 和安全过滤: 除了其他措施外，我们还实施了旨在去除可能包含不安全内容或大量 PII 的网站数据的过滤器，这些网站的域名根据各种 Meta 安全标准被列为有害网站，以及已知包含成人内容的域名。

文本提取和清洗: 我们处理原始 HTML 内容以提取高质量、多样化的文本，并将非截断的网页文档用于此目的。为此，我们构建了一个自定义解析器，该解析器提取 HTML 内容并优化了模板移除和内容召回的精度。我们通过人工评估来评估解析器的质量，并将其与针对类似文章的内容进行优化的流行第三方 HTML 解析器进行了比较，发现其表现良好。我们会谨慎处理包含数学和代码内容的 HTML 页面，以保留这些内容的结构。我们保留图像 alt 属性文本，因为数学内容通常以预渲染图像的形式表示，其中数学也提供在 alt 属性中。

我们发现与纯文本相比，Markdown 对主要训练于 Web 数据的模型性能有害，因此我们删除所有 Markdown 标记。

去重: 我们在 URL、文档和行级别应用多轮去重：

- **URL 级去重:** 我们对整个数据集执行 URL 级去重。对于每个 URL 对应的页面，我们保留最新版本。
- **文档级去重:** 我们对整个数据集执行全局 MinHash (Broder, 1997) 去重以删除近似重复文档。
- **行级去重:** 我们执行类似于 ccNet (Wenzek 等人 ,2019) 的激进行级去重。我们删除在每个包含 3000 万个文档的组中出现超过 6 次的行。

尽管我们的手动定性分析表明行级去重不仅可以去除来自各种网站的剩余模板内容（例如导航菜单、Cookie 预警），还可以去除频繁的高质量文本，但我们的经验评估显示出明显的改进。

启发式过滤: 我们开发了启发式方法来删除额外的低质量文档、异常值和重复次数过多的文档。一些启发式方法的示例包括：

- 我们使用重复 n 元组覆盖率 (Rae et al., 2021) 来删除由重复内容（例如日志或错误消息）组成的行。这些行可能非常长且唯一，因此不能通过行去重进行过滤。

- 我们使用“脏话”计数（Raffel et al., 2020）来过滤掉未被域名黑名单覆盖的成人网站。
- 我们使用标记分布的 Kullback-Leibler 散度来过滤掉包含与训练语料库分布相比过多异常标记的文档。

基于模型的质量过滤:

此外，我们还尝试将各种基于模型的质量分类器用于选择高质量标记。这些方法包括：

- 使用 fasttext (Joulin et al., 2017) 等快速分类器，这些分类器经过训练可以识别给定的文本是否会被维基百科引用（Touvron et al., 2023a)。
- 使用更耗费计算资源的 Roberta 模型分类器（Liu et al., 2019a)，该分类器在 Llama 2 的预测上进行训练。

为了训练基于 Llama 2 的质量分类器，我们创建了一组经过清洗的 Web 文档，描述了质量要求，并指示 Llama 2 的聊天模型确定文档是否满足这些要求。 为提高效率，我们使用 DistilRoberta (Sanh et al., 2019) 为每个文档生成质量分数。 我们将实验评估各种质量过滤配置的效果。

代码和推理数据:

类似于 DeepSeek-AI 等人 (2024)，我们构建了领域特定的管道来提取包含代码和数学相关网页。具体来说，这两个代码和推理分类器都是使用 Llama 2 注解的 Web 数据训练的 DistilledRoberta 模型。 与上面提到的通用质量分类器不同，我们进行了提示调整以针对包含数学推论、STEM 领域的推理以及嵌入自然语言的代码的网页。由于代码和数学的标记分布与自然语言的标记分布截然不同，因此这些管道实现了领域特定的 HTML 提取、自定义文本特征和用于过滤的启发式方法。

多语言数据:

与上面描述的英文处理管道类似，我们实施过滤器以删除可能包含个人身份信息 (PII) 或不安全内容的网站数据。我们的多语言文本处理管道具有以下独特功能：

- 我们使用基于 fasttext 的语言识别模型将文档分类为 176 种语言。
- 我们对每个语言的数据进行文档级和行级去重。
- 我们应用语言特定的启发式方法和基于模型的过滤器来去除低质量文档。

此外，我们使用基于多语言 Llama 2 的分类器对多语言文档进行质量排名，以确保优先处理高质量内容。我们在预训练中使用的多语言标记数量通过实验确定，在英语和多语言基准测试上的模型性能取得平衡。

3.12 确定数据混合

为了获得高质量语言模型，必须谨慎确定预训练数据混合中不同数据源的比例。我们主要利用知识分类和尺度定律实验来确定这一数据混合。

知识分类。我们开发了一个分类器，用于对网页数据中包含的信息类型进行分类，以便更有效地确定数据组合。我们使用这个分类器对网页上过度代表的数据类别（例如艺术和娱乐）进行下采样。

为了确定最佳数据混合方案。我们进行规模定律实验，其中我们将多个小型模型训练于特定数据混合集上，并利用其预测大型模型在该混合集上的性能（参见第 3.2.1 节）。我们多次重复此过程，针对不同的数据混合集选择新的候选数据混合集。随后，我们在该候选数据混合集上训练一个更大的模型，并在多个关键基准测试上评估该模型的性能。

数据混合摘要。我们的最终数据混合包含大约 50% 的通用知识标记、25% 的数学和推理标记、17% 的代码标记以及 8% 的多语言标记。

3.13 退火数据

实证结果表明，在少量高质量代码和数学数据上进行退火（见第 3.4.3 节）可以提升预训练模型在关键基准测试上的性能。类似于李等人的研究（2024b），我们使用包含选定领域高品质数据的混合数据集进行退火。我们的退火数据不包含任何来自常用基准测试的训练集。这使我们能够评估 Llama 3 的真实少样本学习能力和域外泛化能力。

遵循 OpenAI (2023a)，我们在 GSM8k (Cobbe 等人，2021) 和 MATH (Hendrycks 等人，2021b) 训练集上评估了退火的效果。我们发现退火分别提高了预训练 Llama 3 8B 模型在 GSM8k 和 MATH 验证集上的性能 24.0% 和 6.4%。然而，对于 405B 模型的改进可以忽略不计，这表明我们的旗舰模型具有强大的上下文学习和推理能力，不需要特定领域的训练样本就能获得强劲的性能。

使用退火评估数据质量。与 Blakeney 等人 (2024) 一样，我们发现退火使我们能够判断小领域特定数据集的价值。我们通过将已训练 50% 的 Llama 3 8B 模型的学习率在 400 亿个标记上线性退火到 0 来衡量这些数据集的价值。在这些实验中，我们为新数据集分配 30% 的权重，其余 70% 的权重分配给默认数据混合。与对每个小数据集执行标度律实验相比，使用退火来评估新的数据源更加高效。

3.2 模型架构

Llama 3 使用标准的稠密 Transformer 架构 (Vaswani 等人，2017)。它的模型架构与 Llama 和 Llama 2（Touvron 等人，2023a, b）没有显著区别；我们的性能提升主要来自于数据质量和多样性的改进以及训练规模的扩大。

我们做了几个小的修改：

- 我们使用分组查询注意力（GQA；Ainslie 等人（2023）），其中 8 个关键值头用来提高推理速度，并减少解码过程中关键值缓存的大小。
- 我们使用一个注意力掩码来防止序列中不同文档之间的自注意力机制。我们发现，这种改变在标准预训练期间影响有限，但在非常长序列的持续预训练中很重要。
- 我们使用 128K 个标记的词汇表。我们的标记词汇表将 tiktoken3 词汇表的 100K 个标记与 28K 个额外的标记相结合，以更好地支持非英语语言。与 Llama 2 词汇表相比，我们的新词汇表将英语数据样本的压缩率从 3.17 提高到 3.94 个字符/标记。这使得模型能够在相同的训练计算量下“阅读”更多文本。我们还发现，添加来自特定非英语语言的 28K 个标记改善了压缩率和下游性能，而对英语标记化没有影响。
- 我们将 RoPE 基础频率超参数提高到 500,000。这使我们能够更好地支持更长的上下文；Xiong 等人（2023）表明该值对于上下文长度高达 32,768 的情况有效。

	8B	70B	405B
Layers	32	80	126
Model Dimension	4,096	8192	16,384
FFN Dimension	6,144	12,288	20,480
Attention Heads	32	64	128
Key/Value Heads	8	8	8
Peak Learning Rate	3×10^{-4}	1.5×10^{-4}	8×10^{-5}
Activation Function	SwiGLU		
Vocabulary Size	128,000		
Positional Embeddings	RoPE ($\theta = 500,000$)		

Table 3 Overview of the key hyperparameters of Llama 3. We display settings for 8B, 70B, and 405B language models.

Llama 3 405B 使用了一个具有 126 层、16,384 个标记表示维度和 128 个注意力头的架构；有关详细信息，请参阅表 3。这导致模型尺寸约为根据我们的数据和 3.8 × 10^25 FLOPs 的训练预算计算出的计算最优值。

3.2.1 尺度定律

我们利用 Scaling Laws(Hoffmann 等人 ,2022 年 ;Kaplan 等人 ,2020 年) 来确定在我们的预训练计算预算下，旗舰模型的最佳规模。除了确定最佳模型规模之外，预测旗舰模型在下游基准任务上的性能也面临着重大挑战，原因如下：

1. 现有的 Scaling Laws 通常只预测下一个标记预测损失 ,而不是特定的基准测试性能。
2. Scaling Laws 可能存在噪声且不可靠，因为它们是基于使用较小计算预算进行的预训练运行开发而来的（ Wei 等人，2022b ）。

为了应对这些挑战 ,我们实施了一种两阶段方法来开发能够准确预测下游基准测试性能的 Scaling Laws：

1. 我们首先建立预训练 FLOPs 与计算最佳模型在下游任务上的负对数似然之间的相关性。
2. 接下来，我们利用 Scaling Laws 模型和之前使用更高计算 FLOPs 训练的旧模型，将下游任务上的负对数似然与任务准确率相关联。在此步骤中，我们专门利用 Llama 2 系列模型。

这种方法使我们能够根据特定数量的预训练 FLOPs 来预测下游任务性能(对于计算最佳模型)。我们使用类似的方法来选择我们的预训练数据组合 (请参见第 3.4 节)。

Scaling **Law 实验。**具体来说 ,我们通过使用在 6 × 10^18 FLOPs 和 10^22 FLOPs 之间的计算预算预训练模型来构建 Scaling Laws。在每个计算预算下，我们预训练规模在 40M 到 16B 参数之间的模型 ,并在每个计算预算下使用一部分模型规模。在这些训练运行中，我们在 2,000 次训练步骤内使用余弦学习率调度和线性预热。峰值学习率根据模型的大小设置为 2 × 10^-4 到 4 × 10^-4 之间。我们将余弦衰减设置为峰值的 0.1 倍。每一步的权重衰减都设置为该步学习率的 0.1 倍。我们对每个计算规模使用固定的批量大小，范围在 250K 到 4M 之间。

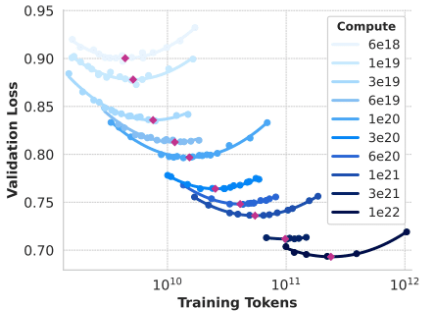

Figure 2 Scaling law IsoFLOPs curves between 6×10^{18} and 10^{22} FLOPs. The loss is the negative log-likelihood on a held-out validation set. We approximate measurements at each compute scale using a second degree polynomial.

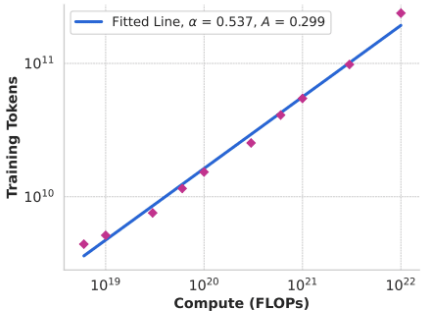

Figure 3 Number of training tokens in identified compute-optimal models as a function of pre-training compute budget. We include the fitted scaling-law prediction as well. The compute-optimal models correspond to the parabola minimums in Figure 2.

这些实验产生了图 2 中的 IsoFLOPs 曲线。这些曲线中的损失是在单独的验证集上测量的。我们使用二阶多项式拟合测得的损失值，并确定每个抛物线的最小值。我们将抛物线的最小值称为相应预训练计算预算下的计算最优模型。

我们使用以这种方式识别的计算最优模型来预测特定计算预算下的最佳训练标记数量。为此，我们假设计算预算 C 与最佳训练标记数量 N (C) 之间存在幂律关系：

N (C) = AC α 。

我们使用图 2 中的数据拟合 A 和 α。我们发现 (α, A) = (0.53, 0.29)；相应的拟合如图 3 所示。将所得缩放定律外推到 3.8 × 10 25 FLOPs 建议训练一个包含 402B 参数的模型，并使用 16.55T 个标记。

一个重要的观察结果是，随着计算预算的增加，IsoFLOPs 曲线在最小值周围变得更平坦。这意味着旗舰模型的性能对模型大小和训练标记之间权衡的小变化相

对稳定。基于这个观察结果，我们最终决定训练一个包含 405B 参数的旗舰模型。

预测下游任务的性能。我们使用生成的计算最优模型来预测旗舰 Llama 3 模型在基准数据集上的性能。首先，我们将基准中正确答案的（归一化）负对数似然与训练 FLOPs 线性相关联。在此分析中，我们仅使用了在上述数据混合上训练至 10^22 FLOPs 的缩放定律模型。接下来，我们使用缩放定律模型和 Llama 2 模型建立对数似然和准确率之间的 S 型关系，Llama 2 模型使用 Llama 2 数据混合和标记器进行训练。我们在图 4 中展示了这个实验在 ARC Challenge 基准上的结果）。我们发现这种两步缩放定律预测（外推超过四个数量级）相当准确：它仅略微低估了旗舰 Llama 3 模型的最终性能。

3.3 基础设施、扩展和效率

我们描述了支持 Llama 3 405B 预训练的硬件和基础架构，并讨论了几项优化措施，这些优化措施提高了训练效率。

3.3.1 训练基础设施

Llama 1 和 Llama 2 模型在 Meta 的 AI 研究超级集群(Lee 和 Sengupta，2022)上进行训练。随着我们进一步扩大规模，Llama 3 的训练被迁移到 Meta 的生产集群（ Lee 等人，2024 ）。这种设置优化了生产级可靠性，这是我们在扩大训练规模时至关重要的。

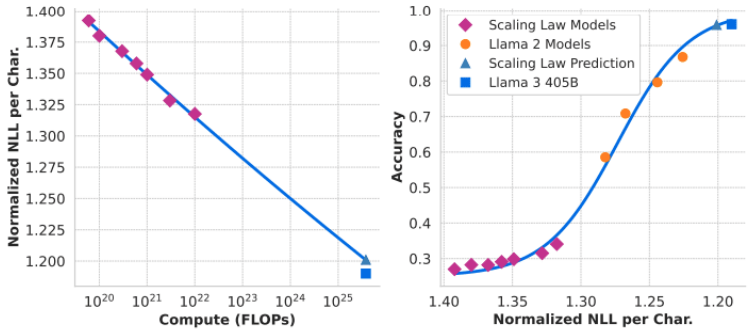

Figure 4 Scaling law forecast for ARC Challenge. *Left:* Normalized negative log-likelihood of the correct answer on the ARC Challenge benchmark as a function of pre-training FLOPs. *Right:* ARC Challenge benchmark accuracy as a function of the normalized negative log-likelihood of the correct answer. This analysis enables us to predict model performance on the ARC Challenge benchmark before pre-training commences. See text for details.

计算资源： Llama 3 405B 在高达 16,000 块 H100 GPU 上进行训练，每块 GPU 以 700W TDP 运行，拥有 80GB HBM3，使用 Meta 的 Grand Teton AI 服务器平台（ Matt Bowman，2022 ）。每个服务器配备八块 GPU 和两块 CPU。在服务器内部，八块 GPU 通过 NVLink 连接。训练作业使用 MAST （ Choudhury 等人 ,2024)进行调度 ,这是 Meta 的全球规模训练调度程序。

存储： Tectonic（ Pan 等人，2021 ），Meta 的通用分布式文件系统，用于构建 Llama 3 预训练的存储结构（ Battey 和 Gupta，2024 ）。它提供 240PB 的存储空间，由 7,500 台配备 SSD 的服务器组成，支持可持续的吞吐量为

2TB/s，峰值吞吐量为 7TB/s。一个主要挑战是支持高突发性检查点写入，这些写入会在短时间内饱和存储结构。检查点保存每个 GPU 的模型状态，范围从 1MB 到每 GPU 4GB，用于恢复和调试。我们的目标是最小化 GPU 在检查点期间的暂停时间，并增加检查点频率以减少在恢复后丢失的工作量。

网络： Llama 3 405B 使用基于 Arista 7800 和 Minipack2 开放计算项目（OCP）机架交换机的 RDMA over Converged Ethernet（RoCE）架构。Llama 3 系列中较小模型使用 Nvidia Quantum2 Infiniband 网络进行训练。RoCE 和 Infiniband 集群都利用 GPU 之间的 400 Gbps 链路连接。尽管这些集群的底层网络技术存在差异，但我们对两者都进行了调整，以提供等效的性能来处理这些大型训练工作量。我们将进一步详细阐述我们的 RoCE 网络，因为我们完全掌握了它的设计。

- **网络拓扑：** 我们的基于 RoCE 的 AI 集群包含 24,000 块 GPU（注脚 5），通过三层 Clos 网络连接（Lee 等人，2024）。在底层，每个机架托管 16 块 GPU，分配到两台服务器上，并通过单个 Minipack2 机架顶端（ToR）交换机连接。在中间层，192 个这样的机架通过集群交换机连接形成一个包含 3,072 块 GPU 的 Pod，具有全双向带宽，确保没有过度订阅。在顶层，同一个数据中心建筑物内的八个这样的 Pod 通过聚合交换机连接以形成一个 24,000 块 GPU 的集群。然而，聚合层的网络连接没有保持全双向带宽，而是采用了 1:7 的过度订阅率。我们的模型并行方法（请参见第 3.3.2 节）和训练作业调度程序（Choudhury

等人，2024）都被优化为知道网络拓扑结构，旨在最大限度地减少 Pod 之间的网络通信。

- **负载均衡：** 大型语言模型的训练会产生难以通过传统方法（如等成本多路径（ECMP）路由）在所有可用网络路径上平衡的繁重网络流量。为了应对这一挑战，我们采用了两种技术。首先，我们的集合库在两个 GPU 之间创建 16 个网络流，而不是一个，从而减少了每个流的流量，并为负载均衡提供了更多流。其次，我们的增强型 ECMP（E-ECMP）协议通过对 RoCE 报头数据包中的其他字段进行哈希，有效地在不同的网络路径上平衡这些 16 个流。

GPUs	TP	CP	PP	DP	Seq. Len.	Batch size/DP	Tokens/Batch	TFLOPs/GPU	BF16 MFU
8,192	8	1	16	64	8,192	32	16M	430	43%
16,384	8	1	16	128	8,192	16	16M	400	41%
16,384	8	16	16	4	131,072	16	16M	380	38%

Table 4 Scaling configurations and MFU for each stage of Llama 3 405B pre-training. See text and Figure 5 for descriptions of each type of parallelism.

- **拥塞控制：** 我们在骨干网中使用深缓冲交换机（Gangidi 等人，2024）来容纳由集合通信模式引起的瞬态拥塞和缓冲。这有助于限制持续拥塞和由慢速服务器引起网络反压的影响，这在训练中很常见。最后，通过 E-ECMP 实现更好的负载均衡大大降低了发生拥塞的可能性。通过这些优化，我们成功地运行了一个 24,000 块 GPU 的集群，无需使用传统的拥塞控制方法，例如数据中心量化拥塞通知（DCQCN）。

3.3.2 模型规模扩展中的并行性

为了扩大我们最大模型的训练规模，我们使用 4D 并行技术 - 一种结合四种不同并行方法的方案 - 对模型进行分片。这种方法有效地将计算分布到许多 GPU 上，并确保每个 GPU 的模型参数、优化器状态、梯度和激活值都能够容纳在它的 HBM 中。我们的 4D 并行实现（如 et al. (2020)；Ren et al. (2021)；Zhao et al. (2023b)）所示，它对模型、优化器和梯度进行分片，同时实现了数据并行，该并行方法在多个 GPU 上并行处理数据并在每个训练步骤后进行同步。我们使用 FSDP 对 Llama 3 的优化器状态和梯度进行分片，但对于模型分片，我们不会在正向计算后重新分片，以避免在反向传递过程中发生额外的全收集通信。

GPU 利用率。通过仔细调整并行配置、硬件和软件，我们实现了 BF16 模型 FLOPs 利用率（MFU；Chowdhery et al. (2023)）的 38-43%。表 4 中显示的配置表明，与 8K GPU 和 DP=64 上的 43% 相比，16K GPU 和 DP=128 上 MFU 降至 41% 是由于需要降低每个 DP 组的批量大小以保持训练期间全局标记数不变。

流水线并行改进。我们在现有的实现中遇到了几个挑战：

- **批量尺寸限制。**当前的实现对每个 GPU 支持的批量尺寸有限制，要求它可被流水线阶段数整除。对于图 6 中的示例，深度优先调度 (DFS) 的流水线并行（Narayanan et al. (2021)）需要 N = PP = 4，而广度优先调度 (BFS；Lamy-Poirier (2023)) 需要 N

= M，其中 M 是总微批量数，N 是同一阶段正向或反向的连续微批量数。然而，预训练通常需要灵活地调整批量大小。

- **内存不平衡。**现有的流水线并行实现会导致资源消耗不平衡。第一阶段由于嵌入和预热微批次而消耗更多内存。
- **计算不平衡。**在模型的最后一层之后，我们需要计算输出和损失，使得这个阶段成为执行延迟的瓶颈。其中 Di 是第 i 个并行维度的索引。在这个例子中，GPU0[TP0, CP0, PP0, DP0] 和 GPU1[TP1, CP0, PP0, DP0] 在同一个 TP 组，GPU0 和 GPU2 在同一个 CP 组，GPU0 和 GPU4 在同一个 PP 组，GPU0 和 GPU8 在同一个 DP 组。

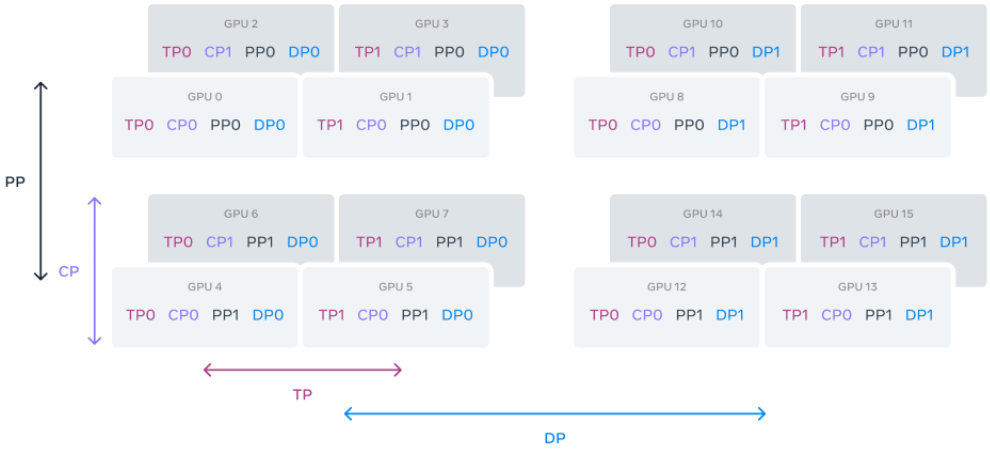

Figure 5 Illustration of 4D parallelism. GPUs are divided into parallelism groups in the order of [TP, CP, PP, DP], where DP stands for FSDP. In this example, 16 GPUs are configured with a group size of |TP|=2, |CP|=2, |PP|=2, and |DP|=2. A GPU's position in 4D parallelism is represented as a vector, $[D_1, D_2, D_3, D_4]$, where D_i is the index on the i-th parallelism dimension. In this example, GPU0[TP0, CP0, PP0, DP0] and GPU1[TP1, CP0, PP0, DP0] are in the same TP group, GPU0 and GPU2 are in the same CP group, GPU0 and GPU4 are in the same PP group, and GPU0 and GPU8 are in the same DP group.

为了解决这些问题，我们修改了管道调度方式，如图 6 所示，这允许灵活地设置 N——在本例中 N = 5，它可以在每个批次中运行任意数量的微批量。 这使得我们可以：

(1) 当有批量大小限制时，运行比阶段数更少的微批量；或

(2) 运行更多微批量来隐藏点对点通信，在深度优先调度（DFS）和广度优先调度（BFS）之间找到最佳的通信和内存效率。为了平衡管道，我们分别从第一阶段和最后一阶段中减少一个 Transformer 层。这意味着第一阶段上的第一个模型块只有嵌入层，而最后一阶段上的最后一个模型块只有输出投影和损失计算。

为了减少管道气泡，我们在一个管道等级上使用了交错调度方式（Narayanan 等人，2021），拥有 V 个管道阶段。 整体管道气泡比率为 PP-1 V * M 。此外，我们采用了异步点对点通信，这显著加快了训练速度，尤其是在文档掩码引入额外计算不平衡的情况下。我们启用 TORCH_NCCL_AVOID_RECORD_STREAMS 来减少来自异步点对点通信的内存使用量。最后，为了降低内存成本，基于详细的内存分配分析，我们主动释放不会用于未来计算的张量，包括每个管道阶段的输入和输出张量。** 通过这些优化，我们可以在不使用激活检查点的情况下，对 8K 个 token 的序列进行 Llama 3 的预训练。

上下文并行化用于长序列。 我们利用上下文并行化（CP）来提高在扩展 Llama 3 上下文长度时的内存效率，并允许在长达 128K 的极长序列上进行训练。在 CP 中，我们跨序列维度进行分区，具体来说我们将输入序列划分为 2 × CP 个

块，以便每个 CP 等级接收两个块以获得更好的负载平衡。第 i 个 CP 等级接收到第 i 个和 (2 × CP -1 -i) 个块。

不同于现有的在环形结构中重叠通信和计算的 CP 实现(Liu et al., 2023a)，我们的 CP 实现采用了一种基于 all-gather 的方法，首先将键值 (K, V) 张量进行全局聚合，然后计算局部查询 (Q) 张量块的注意力输出。尽管 all-gather 通信延迟处于关键路径上，但我们仍然采用这种方法有两个主要原因：

(1) 它更容易更灵活地支持基于 all-gather 的 CP 注意力中的不同类型注意掩码，例如文档掩码；

(2) 暴露的 all-gather 延迟很小，因为通信的 K 和 V 张量远小于 Q 张量，这是由于使用了 GQA (Ainslie et al., 2023)。因此，注意力计算的时间复杂度比 all-gather 大一个数量级（$O(S^2)$ 对 $O(S)$，其中 S 表示完全因果掩码中的序列长度），使得 all-gather 开销可以忽略不计。

网络感知并行化配置。并行化维度的顺序 [TP, CP, PP, DP] 是针对网络通信进行优化的。最内层的并行化需要最高的网络带宽和最低的延迟，因此通常被限制在同一服务器内部。最外层的并行化可能跨越多跳网络，应能够容忍更高的网络延迟。因此，基于对网络带宽和延迟的要求，我们将并行化维度按照 [TP, CP, PP, DP] 的顺序排列。DP (即 FSDP) 是最外层并行化，因为它可以通过异步预取分片模型权重和减少梯度来忍受更长的网络延迟。确定具有最小通信开销的最佳并行化配置，同时避免 GPU 内存溢出是一个挑战。我们开发了一个内存消耗估计器和一个性能投影工具，这有助于我们探索各种并行化配置，并预测整体训练性能和有效地识别内存差距。

数值稳定性。通过比较不同并行设置之间的训练损失，我们修复了一些影响训练稳定性的数值问题。为了确保训练收敛，我们在多个微批次的反向计算过程中使用 FP32 梯度累积，并在 FSDP 中的数据并行工作器之间使用 FP32 进行 reduce-scatter 梯度。对于在正向计算中多次使用的中间张量，例如视觉编码器输出，反向梯度也会以 FP32 累积。

3.3.3 集体通信

Llama 3 的集体通信库基于 Nvidia NCCL 库的分支，称为 NCCLX。NCCLX 极大地提高了 NCCL 的性能，尤其是对于高延迟网络而言。回想一下，并行维度的顺序是 [TP, CP, PP, DP]，其中 DP 对应于 FSDP。最外层的并行维度 PP 和 DP 可能通过多跳网络进行通信，延迟高达数十微秒。原始 NCCL 的集体通

信操作 all-gather 和 reduce-scatter 在 FSDP 中使用，而 point-to-point 通信则用于 PP，这些都需要数据分块和分阶段的数据复制。这种方法会导致以下一些低效率问题：

1. 需要通过网络交换大量小控制消息以促进数据传输；
2. 额外的内存复制操作；
3. 使用额外的 GPU 周期进行通信。

对于 Llama 3 训练，我们通过调整分块和数据传输以适应我们的网络延迟来解决其中一部分低效率问题，这种延迟在大型集群中可能高达数十微秒。我们还允许小控制消息以更高的优先级穿过我们的网络，特别避免在深度缓冲核心交换机中出现队首阻塞。

我们正在进行的工作，用于未来的 Llama 版本，包括对 NCCLX 进行更深层次的修改，以全面解决上述所有问题。

3.3.4 可靠性和运行挑战

16K GPU 训练的复杂性和潜在故障场景超过了我们操作过的更大的 CPU 集群。此外，训练的同步性质使其容错性较低——单个 GPU 故障可能需要重新启动整个作业。尽管面临这些挑战，但对于 Llama 3，我们在支持自动集群维护（例如固件和 Linux 内核升级 (Vigraham 和 Leonhardi，2024))的同时，实现了高于 90% 的有效训练时间，这导致了每天至少一次训练中断。

有效训练时间是指在经过的时间内用于有效训练的时间。 在为期 54 天的预训练快照期间，我们经历了总共 466 次作业中断。其中 47 次是由于自动维护操作（例如固件升级或操作员启动的操作，如配置或数据集更新）而计划的中断。其余 419 次是未预期中断，这些中断按表 5 分类。大约 78% 的意外中断归因于已确认的硬件问题，例如 GPU 或主机组件故障，或者怀疑与硬件相关的问题，例如无声数据损坏和计划外个别主机维护事件。GPU 问题是最大类别，占所有意外问题的 58.7%。尽管存在大量故障，但在这段时间内，仅需要进行三次重大手动干预，其余问题都通过自动化处理。

为了提高有效训练时间，我们减少了作业启动和检查点时间，并开发了用于快速诊断和解决问题的工具。 我们广泛使用了 PyTorch 内置的 NCCL Flight Recorder (Ansel 等人，2024)——此功能将集体元数据和堆栈跟踪捕获到环形缓冲区中，从而使我们能够快速在大规模下诊断挂起和性能问题，特别是在 NCCLX 方面。使用它，我们可以有效地记录每个集体操作的通信事件和持续时间，并在 NCCLX 看门狗或心跳超时时自动转储跟踪数据。通过在线配置更改 (Tang 等人，2015)，我们可以选择性地启用更计算密集的跟踪操作和元数据收集，而无需代码发布或作业重启。 由于我们的网络中混合使用了 NVLink 和

RoCE，因此在大型训练中调试问题变得复杂。数据传输通常通过 CUDA 内核发出的加载/存储操作进行在 NVLink 上进行，远程 GPU 或 NVLink 连接失败经常会表现为 CUDA 内核中的加载/存储操作停滞而不会返回明确的错误代码。NCCLX 通过与 PyTorch 紧密的设计，提高了故障检测和定位的速度和准确性，允许 PyTorch 访问 NCCLX 的内部状态并跟踪相关信息。虽然无法完全防止由于 NVLink 故障导致的停滞，但我们的系统会监视通信库的状态，并在检测到此类停滞时自动超时。此外，NCCLX 还会跟踪每个 NCCLX 通信的内核和网络活动，并提供正在失败的 NCCLX 集体的内部状态快照，包括所有秩之间已完成和未完成的数据传输。我们分析这些数据来调试 NCCLX 扩展问题。

有时，硬件问题会导致仍然可以运行但速度慢的 stragglers，这些 stragglers 很难以检测到。即使只有一个 straggler，它也会减慢数千个其他 GPU 的速度，通常表现为正常运行但通信缓慢。 我们开发了工具来优先处理来自选定进程组的潜在问题通信。通过仅调查少数主要嫌疑犯，我们通常能够有效地识别出 stragglers。

一个有趣的观察是环境因素对大规模训练性能的影响。对于 Llama 3 405B，我们注意到基于时间变化的 1-2% 的吞吐量波动。这种波动是由于更高的中午温度影响 GPU 动态电压和频率缩放造成的。 在训练过程中，数万个 GPU 可能会同时增加或减少功耗，例如由于所有 GPU 正在等待检查点或集体通信结束，或者整个训练作业的启动或关闭。当发生这种情况时，它会导致数据中心内的功耗瞬时波动达到数十兆瓦的幅度，这将拉伸电网的极限。随着我们为未来甚至更大的 Llama 模型扩展训练规模，这是一个持续面临的挑战。

3.4 训练方案

Llama 3 405B 的预训练食谱包含三个主要阶段：

(1) 初始预训练，(2) 长上下文预训练和 (3) 退火。以下分别描述这三个阶段。我们使用类似的食谱来预训练 8B 和 70B 模型。

3.4.1 初始预训练

我们使用余弦学习率计划预训练 Llama 3 405B 模型，最高学习率为 8×10^{-5}，线性预热 8,000 步，经过 1,200,000 步训练后衰减至 8×10^{-7}。为了提高训练的稳定性，我们在训练初期使用较小的批量大小，随后增加批量大小以提高效率。具体来说，我们最初的批量大小为 4M 个 tokens，序列长度为 4,096。在预训练了 252M 个 tokens 后，我们将批量大小和序列长度分别加倍至 8M 个序列、8,192 个 tokens。在预训练了 2.87T 个 tokens 后，我们再次将批量大小加倍至 16M。我们发现这种训练方法非常稳定：很少出现损失峰值，并且不需要进行干预来纠正模型训练的偏离。

调整数据组合。在训练过程中，我们对预训练数据组合进行了几次调整，以改善模型在特定下游任务中的表现。特别是，我们在预训练期间增加了非英语数据的比例，以提高 Llama 3 的多语言表现。我们还上调了数学数据的比例，以增强模型的数学推理能力，在预训练的后期阶段加入了更多的最新网络数据，以更新模型的知识截止点，同时下调了后来被识别为质量较低的数据子集的比例。

3.4.2 长上下文预训练

在预训练的最后阶段，我们训练长序列以支持高达 128,000 个标记的上下文窗口。我们不会更早地训练长序列，因为自注意力层中的计算随着序列长度的增加而二次增长。我们逐步增加支持的上下文长度，并在模型成功适应增加的上下文长度后进行预训练。我们通过测量以下两个方面来评估成功的适应：

(1) 模型在短上下文评估中的性能是否已完全恢复；

(2) 模型是否能够完美地解决长达该长度的“大海捞针”任务。在 Llama 3 405B 预训练中，我们逐步增加了六个阶段的上下文长度，从最初的 8,000 个标记的上下文窗口开始，最终达到 128,000 个标记的上下文窗口。这个长上下文预训练阶段使用了大约 8000 亿个训练标记。

3.4.3 退火

在最后 40M 个标记的预训练过程中，我们将学习率线性退火至 0，同时保持上下文长度为 128K 个标记。在此退火阶段，我们还调整了数据混合，以增加非常高质量的数据源的样本量；请参见第 3.1.3 节。最后，我们在退火期间计算模型检查点的平均值（Polyak (1991) 平均）以生成最终预训练模型。

4 后续训练

我们通过应用多轮后续训练来生成与齐 Llama 3 模型。这些后续训练基于预训练的检查点，并结合人类反馈进行模型对齐（Ouyang 等人，2022；Rafailov 等人，2024）。每轮后续训练都包括监督微调 (SFT)，之后是直接偏好优化 (DPO; Rafailov 等人，2024)，使用通过人工标注或合成生成的示例进行。我们在第 4.1 节和第 4.2 节分别描述了我们的后续训练建模和数据方法。此外，我们将在第 4.3 节进一步详细介绍定制的数据整理策略，以提高模型的推理能力、编程能力、事实性、多语言支持、工具使用、长上下文以及精确指令遵循等方面。

4.1 建模

我们后训练策略的基础是奖励模型和语言模型。 我们首先使用人类标注的偏好数据（见第 4.1.2 节）在预训练检查点之上训练一个奖励模型。然后，我们用监督微调（SFT；见第 4.1.3 节）微调预训练检查点，并进一步用直接偏好优化（DPO；见第 4.1.4 节）与检查点对齐。这个过程如图 7 所示。除非另有说明，否则我们的建模过程适用于 Llama 3 405B，为简便起见，我们将 Llama 3 405B 称为 Llama 3。

4.1.1 聊天对话格式

为了调整大型语言模型（LLM）以实现人机交互，我们需要定义一个聊天对话协议，让模型能够理解人类指令并执行对话任务。与前身相比，Llama 3 具有新的功能，例如工具使用（第 4.3.5 节），这可能需要在单个对话回合中生成多条消息并将它们发送到不同的位置（例如，用户、ipython）。为了支持这一点，我们设计了一种新的多消息聊天协议，该协议使用各种特殊的头部和终止标记。头部标记用于指示对话中每条消息的来源和目标。同样，终止标记也指示何时轮到人类和 AI 交替发言。

4.1.2 奖励模型

我们训练了一个覆盖不同能力的奖励模型（RM），并将其构建在预训练检查点之上。 训练目标与 Llama 2 相同，但我们移除了损失函数中的边际项，因为我们在数据规模扩大后观察到改进效果减小。 与 Llama 2 一样，我们在过滤掉具有相似响应的样本后，使用所有偏好数据进行奖励建模。

除了标准的（被选择，被拒绝）响应偏好对之外，标注还为一些提示创建了第三个“编辑后的响应”，其中从该对中选择的响应会被进一步编辑以改进（请参见第 4.2.1 节)。因此，每个偏好排序样本具有两个或三个响应，排名明确（编辑版 > 选择版 > 拒绝版）。在训练过程中，我们将提示和多个响应连接成一行，并随机排列响应。这是一种将响应放在单独的行中计算分数的标准场景的近似，但在我们的消融实验中，这种方法提高了训练效率，而不会损失精度。

4.1.3 监督微调

首先使用奖励模型对人类标注提示进行拒绝采样，详细方法将在第 4.2 节中描述。我们结合这些拒绝采样的数据和其他数据源（包括合成数据），使用标准交叉熵损失对预训练语言模型进行微调，目标是预测目标标记（同时屏蔽提示标记的损失）。有关数据混合的更多详细信息请参见第 4.2 节。尽管许多训练目标都是模型生成的，我们仍将此阶段称为监督微调（SFT； Wei 等人，2022a；Sanh 等人，2022； Wang 等人，2022b）。

我们的最大模型在 8.5K 至 9K 步内以 1e-5 的学习率进行微调。我们发现这些超参数设置适用于不同轮次和数据混合。

4.1.4 直接偏好优化

我们进一步使用直接偏好优化（DPO；Rafailov 等人，2024）训练我们的 SFT 模型，以实现人类偏好对齐。在训练中，我们主要使用上一轮对齐中表现最佳的模型收集到的最新偏好数据批次。结果，我们的训练数据更符合每一轮优化的策略模型分布。我们也探索了策略算法，例如 PPO（Schulman 等人，2017），但发现 DPO 在大规模模型上需要更少的计算量，并且表现更好，尤其是在指令遵循基准测试中，如 IFEval（Zhou 等人，2023）。

对于 Llama 3，我们使用 1e-5 的学习率，并将 β 超参数设置为 0.1。此外，我们还对 DPO 应用了以下算法修改：

- **在 DPO 损失中屏蔽格式标记：** 我们将特殊格式标记（包括第 4.1.1 节中描述的头部和终止标记）从选定的和拒绝的响应中屏蔽掉，以稳定 DPO 训练。我们注意到，这些标记参与损失可能会导致不希望的模型行为，例如尾部重复或突然生成终止标记。我们假设这是由于 DPO 损失的对比性质——在选定和拒绝的响应中都存在共同标记会导致相互冲突的学习目标，因为模型需要同时增加和减少这些标记的可能性。
- **使用 NLL 损失进行正则化：** 我们对选定的序列添加了一个额外的负对数似然（NLL）损失项，其缩放系数为 0.2，类似于 Pang 等人（2024）。这有助于进一步稳定 DPO 训练，通过保持生成

所需的格式并防止选定响应的对数概率降低(Pang 等人 ,2024 ; Pal 等人，2024）。

4.1.5 模型平均

最后，我们在每个 RM、SFT 或 DPO 阶段使用各种数据版本或超参数的实验中获得的模型进行平均(Izmailov 等人 , 2019 ; Wortsman 等人 , 2022 ; Li 等人，2022）。我们列出了用于 Llama 3 调节的内部收集的人类偏好数据的统计信息。我们请评价者与模型进行多轮对话 ,并比较每轮的回复。在后处理过程中，我们将每个对话拆分成多个示例，每个示例包含一个提示（包括如果有的话前一次对话）和一个回复（例如，被选择或拒绝的回复）。

4.1.6 迭代轮次

继 Llama 2 之后，我们应用上述方法进行了六轮迭代。在每个循环中，我们收集新的偏好标注和微调（SFT）数据，并从最新的模型中采样合成数据。

4.2 训练后数据

后训练数据的组成对语言模型的实用性和行为起着至关重要的作用。在本节中，我们将讨论我们的注释程序和偏好数据收集（第 4.2.1 节）、SFT 数据的组成（第 4.2.2 节）以及数据质量控制和清洗的方法（第 4.2.3 节）。

4.2.1 偏好数据

我们的偏好数据标注过程与 Llama 2 相似。在每一轮之后，我们部署多个模型进行标注，并为每个用户提示采样两个来自不同模型的响应。这些模型可以使用不同的数据混合和对齐方案训练，从而具有不同的能力强度（例如，代码专业知识）和增加数据多样性。我们要求标注者根据其偏好程度将偏好评分分类为四个级别之一：显著更好、更好、略好或略微更好。

我们还在偏好排序之后加入了一个编辑步骤，以鼓励标注者进一步改进首选响应。标注者可以直接编辑所选响应，或者用反馈提示模型以细化其自身响应。因此，部分偏好数据具有三个排序的响应（编辑 > 选择 > 拒绝）。

表 6 中报告了我们用于 Llama 3 训练的偏好注释统计数据。通用英语涵盖多个子类别，例如基于知识的问答或精确的指令遵循，这些都超出了特定能力的范围。与 Llama 2 相比，我们观察到提示和响应的平均长度增加了，这表明我们正在更复杂的任务上训练 Llama 3。此外，我们实施了质量分析和人工评估流程来严格评估收集的数据，使我们能够完善提示并为标注者提供系统、可行的反馈。

例如，随着 Llama 3 在每一轮后都得到改进，我们将相应地增加提示的复杂性以针对模型落后的领域。

在每一轮后期训练中，我们会使用当时所有可用的偏好数据进行奖励建模，而只使用来自各种能力的最新批次进行 DPO 训练。对于奖励建模和 DPO，我们使用标记为“选择响应显著更好或更好”的样本进行训练，并将具有相似响应的样本丢弃。

4.2.2 SFT 数据

我们的微调数据主要来自于以下来源：

- 来自我们的人工标注收集中的提示及其拒绝采样响应
- 针对特定能力的合成数据（详情见第 4.3 节）
- 少量人工标注的数据（详情见第 4.3 节）

随着我们后训练周期的进行，我们开发出更强大的 Llama 3 变体，并使用这些变体收集更大的数据集，以覆盖广泛的复杂能力范围。在本节中，我们将讨论拒绝采样过程和最终 SFT 数据混合的总体构成的细节。

拒绝采样。在拒绝采样（RS）中，对于我们在人工标注期间收集的每个提示（第 4.2.1 节），我们从最新的聊天模型策略中采样 K 个输出（通常是前一个后训练迭代中的最佳执行检查点，或者特定能力的最佳执行检查点）并且使用我们的奖

励模型来选择最佳候选者，与 Bai 等人 (2022) 一致。在后训练的后期阶段，我们会引入系统提示来引导 RS 响应符合期望的语气、风格或格式，这对于不同的能力可能会有所不同。

为了提高拒绝采样的效率，我们采用了 PagedAttention（Kwon 等人，2023）。PagedAttention 通过动态键值缓存分配来提高内存效率。它通过根据当前缓存容量动态调度请求来支持任意输出长度。不幸的是，这会带来内存耗尽时的交换风险。为了消除这种交换开销，我们定义了一个最大输出长度，并且只有在有足够的内存可以容纳该长度的输出时才执行请求。PagedAttention 还使我们可以跨所有相应输出共享提示的键值缓存页面。总而言之，这导致拒绝采样过程中的吞吐量提高了 2 倍以上。

总体数据组成。表 7 显示了我们“有用性”混合每个广泛类别的数据统计信息。虽然 SFT 和偏好数据包含重叠的领域，但它们的策划方式不同，导致不同的计数统计量。在第 4.2.3 节中，我们将描述用于分类我们的数据样本的主题、复杂性和质量的技术。在每一轮后训练中，我们都会仔细调整我们总体的数据混合，以在多个轴上调整性能，以适应广泛的基准测试。我们的最终数据混合会对某些高质量来源进行多次迭代，并将其他来源进行降采样。

4.2.3 数据处理与质量控制

考虑到我们的大部分训练数据都是模型生成的，因此需要仔细清洗和质量控制。

数据清洗：在早期阶段，我们观察到数据中存在许多不希望出现的模式，例如过量使用表情符号或感叹号。因此，我们实施了一系列基于规则的数据删除和修改策略来过滤或清除有问题的數據。例如，为了减轻过度道歉的语调问题，我们将识别过度使用的短语（例如“对不起”或“我道歉”），并仔细平衡数据集中此类样本的比例。

数据修剪： 我们还应用了一系列基于模型的技术来删除低质量的训练样本并提高整体模型性能：

- **主题分类：** 我们首先将 Llama 3 8B 微调成一个主题分类器，并对所有数据进行推理以将其分类为粗粒度类别（“数学推理”）和细粒度类别（“几何与三角函数”）。
- **质量评分：** 我们使用奖励模型和基于 Llama 的信号来获得每个样本的质量分数。对于基于 RM 的得分，我们将得分处于最高四分位数的数据视为高质量数据。对于基于 Llama 的得分，我们提示 Llama 3 检查点以三个等级（准确性、指令遵循性和语气/呈现）对通用英语数据进行评分，并以两个等级（错误识别和用户意图）对代码数据进行评分，并将获得最高分的样本视为高质量数据。RM 和基于 Llama 的分数的冲突率很高，我们发现将这些信号结合起来可以获得最佳的内部测试集召回率。最终，我们选择那些被 RM 或基于 Llama 过滤器标记为高质量的示例。

- **难度评分：** 由于我们还对优先考虑更复杂的模型示例感兴趣，我们使用两个难度度量来评分数据：Instag (Lu et al., 2023) 和基于 Llama 的评分。对于 Instag，我们提示 Llama 3 70B 对 SFT 提示执行意图标记，其中更多的意图意味着更高的复杂性。我们还提示 Llama 3 以三个等级（Liu et al., 2024c）测量对话的难度。

- **语义去重：** 最后，我们执行语义去重 (Abbas et al., 2023; Liu et al., 2024c)。 我们首先使用 RoBERTa (Liu et al., 2019b) 对完整的对话进行聚类，并在每个聚类中按质量分数 × 难度分数排序。然后，我们通过迭代所有排序示例进行贪婪选择，只保留与到目前为止在聚类中看到的示例的最大余弦相似度小于阈值的示例。

4.3 能力

我们特别强调了一些为了提升特定能力所做的努力，例如代码处理（第 4.3.1 节）、多语言性（第 4.3.2 节）、数学和推理能力（第 4.3.3 节）、长上下文处理能力（第 4.3.4 节）、工具使用（第 4.3.5 节）、事实性（第 4.3.6 节）以及可控性（第 4.3.7 节）。

4.3.1 代码

自从 Copilot 和 Codex (Chen 等人，2021 年) 发布以来，用于代码的 LLM 已受到广泛关注。开发人员现在广泛使用这些模型来生成代码片段、调试、自动化任务和提高代码质量。对于 Llama 3，我们的目标是改进并评估以下优先级编程语言的代码生成、文档、调试和审查功能：Python、Java、JavaScript、C/C++、TypeScript、Rust、PHP、HTML/CSS、SQL、bash/shell。在这里，我们介绍了通过训练代码专家、为 SFT 生成合成数据、通过系统提示转向改进格式以及创建质量过滤器以从训练数据中删除不良样本来改进这些编码功能的工作。

专家训练。我们训练了一个代码专家，并在随后的多轮后训练中使用它来收集高质量的人类代码注释。这是通过分支主要预训练运行并继续对主要（>85%）为代码数据的 1T 令牌混合进行预训练来实现的。已证明在特定领域持续进行领域特定数据预训练可以有效地提高性能 (Gururangan 等人，2020 年)。我们遵循与 CodeLlama (Rozière 等人，2023 年) 相似的配方。在训练的最后几千步中，我们在高质量的仓库级别代码数据混合上执行长上下文微调 (LCFT)，将专家的上下文长度扩展到 16K 令牌。最后，我们遵循第 4.1 节中描述的类似的后训练建模配方来对齐该模型，但使用主要针对代码的 SFT 和 DPO 数据混合。该模型也用于编码提示的拒绝采样 (第 4.2.2 节)。

合成数据生成。在开发过程中，我们发现代码生成存在关键问题，包括难以遵循指令、代码语法错误、代码生成不正确以及难以修复错误。尽管理论上密集的人类注释可以解决这些问题，但合成数据生成提供了一种补充方法，成本更低，规模更大，不受标注人员专业水平的限制。

因此，我们使用 Llama 3 和代码专家来生成大量合成的 SFT 对话。 我们描述了生成合成代码数据的三个高级方法。总的来说，我们在 SFT 期间使用了超过 270 万个合成示例。

1. **合成数据生成：执行反馈。**8B 和 70B 模型在由更大的、更能胜任的模型生成的训练数据上表现出显著的性能改进。然而，我们的初步实验表明，仅用 Llama 3 405B 训练其自身生成的數據没有帮助（甚至会降低性能）。为了解决这一限制，我们引入了执行反馈作为真理之源，使模型能够从错误中学习并保持在正轨上。特别是，我们使用以下过程生成大约一百万个合成的代码对话数据集：

- **问题描述生成：**首先，我们生成了一大批涵盖各种主题（包括长尾分布）的编程问题描述。为了实现这种多样性，我们从各种来源随机采样代码片段并提示模型根据这些示例生成编程问题。这使我们能够利用广泛的主题并创建一个全面的问题描述集（Wei 等人，2024 年)。
- **解决方案生成：**然后，我们提示 Llama 3 用给定的编程语言解决每个问题。我们观察到在提示中添加良好的编程规则可以提高生成的解决方案质量。此外，我们发现要求模型用注释解释其思维过程也很有帮助。
- **正确性分析：** 在生成解决方案后，识别其正确性并非保证，并将不正确的解决方案包含在微调数据集可能会损害模型的质量至关重要。虽然我们不能确保完全正确，但我们开发了近似正确性的方

法。为此，我们将提取的源代码从生成的解决方案中，并应用静态和动态分析技术组合进行测试其正确性，包括：

- **静态分析：** 我们将所有生成的代码都通过解析器和代码检查工具运行，以确保语法正确性，捕捉语法错误、未初始化变量或未导入函数的使用、代码风格问题、类型错误等。
- **单元测试生成和执行：** 对于每个问题和解决方案，我们提示模型生成单元测试，并在容器化环境中与解决方案一起执行，捕获运行时执行错误和一些语义错误。

- **错误反馈和迭代自我校正：** 当解决方案在任何步骤都失败时，我们将提示模型对其进行修改。提示包含原始问题描述、错误解决方案以及来自解析器/代码检查工具/测试程序的反馈（标准输出、标准错误和返回码）。单元测试执行失败后，模型可以修复代码以通过现有的测试，或者修改其单元测试以适应生成的代码。只有通过所有检查的对话才会被包含在最终的数据集中，用于监督式微调（SFT）。值得注意的是，我们观察到大约 20% 的解决方案最初不正确但进行了自我校正，这表明模型从执行反馈中学习并提高了其性能。

- **微调和迭代改进：** 微调过程会进行多轮，每轮都建立在之前的轮子上。每次微调后，模型都会得到改进，为下一轮生成更高质量的合成数据。这个迭代过程允许对模型性能进行逐步细化和增强。

2. 合成数据生成：编程语言翻译。 我们观察到主要编程语言（例如 Python/C++）和不太常见的编程语言（例如 Typescript/PHP）之间存在性能差距。这并不奇怪，因为我们对于不太常见的编程语言拥有更少训练数据。为了缓解这种情况，我们将通过将常见编程语言的数据翻译成不太常见的语言来补充现有数据（类似于 Chen 等人 (2023) 在推理领域）。这是通过提示 Llama 3 并通过语法解析、编译和执行来确保质量实现的。图 8 展示了从 Python 翻译的合成 PHP 代码示例。这显著提高了 MultiPL-E (Cassano 等人，2023) 基准测量的不太常见语言的性能。

3. 合成数据生成：反向翻译。 为了改进某些编码能力（例如文档、解释），在执行反馈的信息量不足以确定质量的情况下，我们采用另一种多步骤方法。使用此过程，我们生成了约 120 万个与代码解释、生成、文档和调试相关的合成对话。从预训练数据中各种语言的代码片段开始：

- **生成：** 我们提示 Llama 3 生成代表目标能力的数据（例如，为代码片段添加注释和文档字符串，或者要求模型解释一段代码）。
- **反向翻译:** 我们会提示模型将合成生成的数据“反向翻译”回原始代码（例如，我们提示模型仅从其文档中生成代码，或者让我们询问模型仅从其解释中生成代码）。
- **过滤:** 使用原始代码作为参考，我们提示 Llama 3 确定输出的质量（例如，我们询问模型反向翻译后的代码与原始代码的忠实度如何）。然后，我们在 SFT 中使用具有最高自我验证分数的生成的示例。

系统提示引导，用于拒绝采样。 在拒绝采样过程中，我们使用代码特定的系统提示来改进代码的可读性、文档、完整性和具体性。回忆第 7 节中提到的内容，这些数据用于微调语言模型。图 9 显示了系统提示如何帮助提高生成代码质量的示例 - 它添加了必要的注释，使用了更具信息量的变量名，节省内存等。

使用执行和模型作为评判标准过滤训练数据。 如第 4.2.3 节所述，我们偶尔会在拒绝采样的数据中遇到质量问题，例如包含错误的代码块。检测拒绝采样数据

中的这些问题不像检测我们的合成代码数据那样简单，因为拒绝采样的响应通常包含自然语言和代码的混合体，而这些代码可能并不总是可执行的。（例如，用户提示可能会明确要求伪代码或仅对可执行程序的非常小的片段进行编辑。）为了解决这个问题，我们利用“模型作为评判者”方法，其中更早版本的 Llama 3 会根据两个标准评估并分配二进制（0/1）分数：代码正确性和代码风格。我们只保留获得 2 分满分的样本。最初，这种严格的过滤导致了下游基准性能下降，主要是因为它不成比例地删除了具有挑战性提示的示例。为了抵消这一点，我们将一些分类为最具挑战性的编码数据的响应进行战略性修改，直到它们满足基于 Llama 的“模型作为评判者”标准。通过改进这些具有挑战性的问题，编码数据在质量和难度之间取得了平衡，从而实现了最佳的下游性能。

4.3.2 多语言能力

本节描述了我们如何改进 Llama 3 的多语言能力，包括：训练一个在更多多语言数据上专业化的专家模型；为德语、法语、意大利语、葡萄牙语、印地语、西班牙语和泰语采购并生成高质量的多语言指令微调数据；以及解决多语言语言引导的具体挑战，以提高我们模型的整体性能。

专家训练。我们的 Llama 3 预训练数据混合包含的英语标记远远多于非英语标记。为了收集更高质量的非英语人工注释，我们通过分支预训练运行并继续在包含 90% 多语言标记的数据混合上进行预训练来训练一个多语言专家模型。然后，

我们在第 4.1 节中按照说明对该专家模型进行后训练。然后，该专家模型用于收集更高质量的非英语人工注释，直到预训练完全完成。

多语言数据收集。我们的多语言 SFT 数据主要来自以下来源。总体分布为 :2.4% 人工注释、44.2% 来自其他 NLP 任务的数据、18.8% 拒绝采样数据和 34.6% 翻译推理数据。

- **人工注释 :**我们从语言学家和母语人士那里收集高质量的、手动标注的数据。这些注释主要由开放式提示组成，它们代表了现实世界的用例。
- **来自其他 NLP 任务的数据 :**为了进一步增强，我们使用来自其他任务的多语言训练数据并将它们改写成对话格式。例如，我们使用了 exams-qa(Hardalov 等人 ,2020 年)和 Conic10k(Wu 等人，2023 年）的数据。为了改进语言对齐，我们还使用了 GlobalVoices（ Prokopidis 等人，2016 年）和 Wikimedia（ Tiedemann，2012 年) 的并行文本。我们使用基于 LID 的过滤和 Blaser2.0（ Seamless Communication 等人，2023 年）来删除低质量数据。对于并行文本数据，我们没有直接使用双文本对，而是应用了一个受 Wei 等人 (2022a) 启发的多语言模板，以更好地模拟翻译和语言学习场景中的真实对话。
- **拒绝采样数据 :**我们将拒绝采样应用于人类注释的提示，以生成高质量的样本用于微调，与英语数据的处理过程相比进行了很少的修改：

- 生成：我们在后训练的早期轮次中探索了随机选择温度超参数的范围为 0.2 -1 ,以实现多样化的生成。使用高温度时，多语言提示的响应可能会变得富有创意和鼓舞人心，但也会容易出现不必要的或不自然的代码切换。在后训练的最后阶段 ,我们使用 0.6 的恒定值来平衡这一权衡。此外 ,我们还使用了专门的系统提示来改进响应格式、结构和一般可读性。
- 选择 :在基于奖励模型的选择之前 ,我们实施了多语言特定的检查，以确保提示和回复之间的语言匹配率很高(例如 ,罗马化印地语提示不应期望使用印地语梵文脚本进行回复）。

- **翻译数据 :**我们试图避免使用机器翻译数据来微调模型 ,以防止出现翻译式英语（Bizzoni 等人，2020 年；Muennighoff 等人，2023 年）或可能出现的姓名偏见（Wang 等人，2022a）、性别偏见(Savoldi 等人 ,2021 年)或文化偏见(Ji 等人 ,2023 年)。此外 ,我们旨在防止模型仅接触根植于英语文化背景的任务 ,这些任务可能无法代表我们旨在捕捉的语言和文化多样性。我们对此做了一个例外，并将合成的定量推理数据（有关详细信息，请参见第 4.3.3 节）翻译成非英语，以提高非英语语言中的定量推理性能。由于这些数学问题的语言本质简单 ,因此发现翻译后的样本几乎没有质量问题。我们观察到从添加此翻译数据的 MGSM（Shi 等人，2022 年）获得了显著的收益。

4.3.3 数学与推理

我们定义推理为执行多步计算并得出正确最终答案的能力。

几个挑战指导了我们训练擅长数学推理模型的方法：

- **缺乏提示:** 随着问题复杂度的增加，用于监督微调（SFT）的有效提示或问题的数量减少。这种稀缺性使得创建多样化且具有代表性的训练数据集来教授模型各种数学技能变得困难（Yu 等人，2023；Yue 等人，2023；Luo 等人，2023；Mitra 等人，2024；Shao 等人，2024；Yue 等人，2024b）。
- **缺乏真实推理过程:** 有效的推理需要逐步解决方案来促进推理过程（Wei 等人，2022c）。然而，通常缺少真实推理过程，这些过程对于指导模型如何逐步分解问题并得出最终答案至关重要（Zelikman 等人，2022）。
- **错误的中间步骤:** 使用模型生成的推理链时，中间步骤可能并不总是正确（Cobbe 等人，2021；Uesato 等人，2022；Lightman 等人，2023；Wang 等人，2023a）。这种不准确性会导致最终答案错误，需要解决。
- **训练模型使用外部工具:** 增强模型以利用外部工具（如代码解释器）的能力，使其可以通过交织代码和文本进行推理（Gao 等人，2023；

Chen 等人，2022；Gou 等人，2023）。这种能力可以显着提高他们的问题解决能力。

- **训练和推理之间的差异:** 模型在训练期间进行微调的方式通常与在推理期间使用的方式存在差异。在推理过程中，经过微调的模型可能会与人类或其他模型互动，需要通过反馈来改进其推理能力。确保训练和实际应用之间的一致性对于维持推理性能至关重要。

为了解决这些挑战，我们应用以下方法：

- **解决提示缺乏问题:** 我们从数学上下文中获取相关预训练数据，并将其转换为问题-答案格式，可用于监督微调。此外，我们会识别模型表现不佳的数学技能，并积极地从人类那里收集提示以教授模型这些技能。为了促进这一过程，我们创建了一个数学技能分类法（Didolkar 等人，2024），并要求人类提供相应的提示/问题。
- **用逐步推理步骤增强训练数据:** 我们使用 Llama 3 为一组提示生成逐步解决方案。对于每个提示，模型都会产生可变数量的生成结果。然后根据正确答案过滤这些生成结果（Li 等人，2024a）。我们还进行自我验证，其中 Llama 3 用于验证特定逐步解决方案是否对给定的问题有效。这个过程通过消除模型不产生有效推理轨迹的实例来提高微调数据的质量。
- **过滤错误的推理步骤:** 我们训练结果和逐步奖励模型（Lightman 等人，2023；Wang 等人，2023a）来过滤中间推理步骤不正确的训练数据。这些奖励模型用于消除具有无效逐步推理的数据，确

保微调得到高质量的数据。对于更具挑战性的提示，我们使用带有学习逐步奖励模型的蒙特卡罗树搜索（MCTS）来生成有效的推理轨迹，从而进一步增强高质量推理数据的收集（Xie 等人，2024）。

- **交织代码和文本推理：**我们提示 Llama 3 通过文本推理及其关联 Python 代码组合来解决推理问题（Gou 等人，2023）。代码执行用作反馈信号，以消除推理链无效的情况，确保推理过程的正确性。
- **从反馈和错误中学习：**为了模拟人类反馈，我们利用错误的生成结果（即导致错误推理轨迹的生成结果），并通过提示 Llama 3 生成正确的生成结果来进行错误更正（An 等人，2023b；Welleck 等人，2022；Madaan 等人，2024a）。使用错误尝试的反馈并纠正它们的自迭代过程有助于提高模型准确推理的能力并从错误中学习。

4.3.4 长上下文

在最后的预训练阶段，我们将 Llama 3 的上下文长度从 8K 个标记扩展到 128K 个标记（有关详细信息请参见第 3.4 节）。与预训练类似，我们发现在微调过程中，必须仔细调整配方以平衡短上下文和长上下文能力。

SFT 和合成数据生成。简单地应用我们的现有 SFT 配方，只使用短上下文数据导致预训练中的长上下文能力显著下降，突出了在 SFT 数据组合中纳入长上下

文数据的必要性。然而，实际上，让人工标注这些示例大部分是不切实际的，因为阅读冗长的上下文非常繁琐且耗时，因此我们主要依赖合成数据来弥补这一差距。 我们使用早期版本的 Llama 3 生成合成数据，基于关键的长上下文用例：（可能为多轮）问答、长文档摘要以及代码库上的推理，并在下面更详细地描述这些用例。

- **问答：** 我们从预训练数据集中仔细挑选了一组长文档。我们将这些文档拆分成 8K 个标记的块，并提示早期版本的 Llama 3 模型在随机选择的块上生成 QA 对。 在训练过程中，整个文档都被用作上下文。
- **摘要：** 我们通过首先使用我们最强的 Llama 3 8K 上下文模型对 8K 输入长度的块进行层次化摘要来应用长上下文文档的层次化摘要。然后对这些摘要进行汇总。 在训练中，我们提供完整文档并提示模型总结文档，同时保留所有重要细节。 我们还基于文档的摘要生成 QA 对，并用需要对整个长文档全局理解的问题提示模型。
- **长上下文代码推理：**我们解析 Python 文件以识别导入语句并确定它们的依赖关系。 从这里开始，我们选择最常用的文件，特别是那些被至少其他五个文件引用的文件。 我们从存储库中删除这些关键文件中的一个，并提示模型识别依赖于缺少的文件，并生成必要的缺失代码。

我们进一步根据序列长度（16K、32K、64K 和 128K）对这些合成生成的样本进行分类，以实现更精细的输入长度定位。

通过仔细的消融实验，我们观察到将 0.1% 的合成生成的长上下文数据与原始短上下文数据混合可以优化短上下文和长上下文基准测试的性能。

DPO。 我们注意到在 DPO 中仅使用短上下文训练数据不会对长上下文性能产生负面影响，只要 SFT 模型对长上下文任务效果良好。我们怀疑这是因为我们的 DPO 配方比 SFT 的优化器步骤更少。考虑到这一发现，我们在长上下文 SFT 检查点之上保留标准的短上下文 DPO 配方。

4.3.5 工具使用

教导大型语言模型（LLM）使用诸如搜索引擎或代码解释器之类的工具，可以大大扩展它们能够解决的任务范围，将它们从纯粹的聊天模型转变为更通用的助手（Nakano 等人，2021 年；Thoppilan 等人，2022 年；Parisi 等人，2022 年；Gao 等人，2023 年；Mialon 等人，2023a；Schick 等人，2024 年）。我们训练 Llama 3 与以下工具进行交互：

- 搜索引擎。Llama 3 被训练使用 Brave Search7 来回答关于其知识截止日期之后的近期事件的问题，或者需要从网络中检索特定信息的请求。

- Python 解释器。Llama 3 可以生成并执行代码以执行复杂的计算，读取用户上传的文件并根据这些文件解决任务，例如问答、摘要、数据分析或可视化。
- 数学计算引擎。Llama 3 可以使用 Wolfram Alpha API8 更准确地解决数学和科学问题，或者从 Wolfram 的数据库中检索准确信息。

生成的模型能够在聊天设置中使用这些工具来解决用户的查询，包括多回合对话。如果查询需要多次调用工具，该模型可以编写逐步计划，依次调用工具，并在每次工具调用后进行推理。

我们还提高了 Llama 3 的零样本工具使用能力——给定上下文环境下可能未见过的工具定义和用户查询，我们训练模型生成正确的工具调用。

实现。我们将核心工具实施为具有不同方法的 Python 对象。零样本工具可以作为带有描述、文档（即如何使用它们的示例）的 Python 函数来实现，模型只需要函数签名和 docstring 作为上下文即可生成适当的调用。

我们还将函数定义和调用转换为 JSON 格式，例如用于 Web API 调用。所有工具调用都由 Python 解释器执行，Python 解释器必须在 Llama 3 系统提示中启用。可以在系统提示中单独启用或禁用核心工具。

数据收集。与 Schick 等人 (2024) 不同，我们依靠人类标注和偏好来教 Llama 3 使用工具。这与 Llama 3 中通常使用的后训练管道存在两个主要区别：

- 关于工具，对话常常包含不止一条助理消息（例如，调用工具并对工具输出进行推理）。因此，我们进行消息级别的标注以收集详细的反馈：标注者在相同上下文下提供两条助理消息的偏好，或者如果两者都存在主要问题，则编辑其中一条消息。然后将被选择或修改的消息添加到上下文中，对话继续进行。这为助理调用工具和推理工具输出的能力提供了人类反馈。标注者不能对工具输出进行排名或编辑。
- 我们没有执行拒绝采样，因为在我们的工具基准测试中没有观察到收益。

为了加速标注过程，我们首先通过微调来自之前 Llama 3 检查点的合成数据来引导基本的工具使用能力。这样一来，标注者就需要进行较少的编辑操作。类似地，随着 Llama 3 在开发过程中逐渐改进，我们会逐步复杂化我们的人类标注协议：我们从单回合的工具使用标注开始，然后转向对话中的工具使用，最后注释多步骤工具使用和数据分析。

工具数据集。为了创建用于工具使用应用程序的数据，我们采用以下步骤:

- **单步工具使用:** 我们首先进行少量样本生成合成用户提示，这些提示通过构造需要调用我们的核心工具之一（例如，超过我们知识截止日期的问题）。然后，仍然依靠少量样本生成，我们为这些提示生成适当的工具调用，执行它们，并将输出添加到模型的上下文中。最后，我们再次提示模型以根据工具输出生成对用户查询的最终

答案。我们最终得到以下形式的轨迹：系统提示、用户提示、工具调用、工具输出、最终答案。我们还过滤掉了大约 30% 的数据集，以删除无法执行的工具调用或其他格式问题。

- **多步骤工具使用:** 我们遵循类似的协议，首先生成合成数据来教授模型基本的 다단계 工具使用能力。为此，我们首先提示 Llama 3 生成至少需要两次工具调用的用户提示（可以是来自我们的核心集中的相同工具或不同工具）。然后，根据这些提示，我们进行少量样本提示 Llama 3 生成一个解决方案，该解决方案由交织的推理步骤和工具调用组成，类似于 ReAct (Yao 等人，2022)。请参见图 10，以了解 Llama 3 执行涉及多步骤工具使用任务的示例。
- **文件上传：** 我们针对以下文件类型进行标注：.txt， .docx， .pdf， .pptx， .xlsx， .csv， .tsv， .py ， .json， .jsonl， .html， .xml。我们的提示基于提供文件，并要求总结文件内容、查找和修复错误、优化代码段、执行数据分析或可视化。图 11 显示了 Llama 3 执行涉及文件上传任务的示例。

在对此合成数据进行微调后，我们收集了来自各种场景的人类标注，包括多轮交互、超过三步的工具使用以及工具调用无法产生令人满意的答案的情况。我们用不同的系统提示增强合成数据，以教导模型仅在被激活时才使用工具。为了训练模型避免对简单查询进行工具调用，我们还添加了来自易于计算或问答数据集（Berant 等人，2013；Koncel-Kedziorski 等人，2016；Joshi 等人，2017；Amini 等人，2019）的查询及其响应，这些响应不使用工具，但在系统提示中激活了工具。

零样本工具使用数据: 我们通过对大量且多样的部分合成（函数定义、用户查询、相应调用）元组进行微调，从而改善 Llama 3 的零样本工具使用能力（也称为功能调用）。我们在未见过工具的集合上评估我们的模型。

- **单一、嵌套和并行函数调用：** 调用可以是简单的、嵌套的（即我们将函数调用作为另一个函数的参数传递）或平行的（即模型返回一个独立函数调用的列表）。生成各种函数、查询和真实结果可能具有挑战性（Mekala 等人，2024），我们依靠挖掘 Stack（Kocetkov 等人，2022）来将我们的合成用户查询基于真实的函数。更准确地说，我们提取函数调用及其定义，清理和过滤它们（例如，缺少文档字符串或不可执行的函数），并使用 Llama 3 生成与函数调用相对应的自然语言查询。
- **多回合函数调用：** 我们还为包含函数调用的多回合对话生成合成数据，遵循类似于 Li 等人（2023b）中提出的协议。我们使用多个代理来生成域、API、用户查询、API 调用和响应，同时确保生

成的數據涵盖了一系列不同的领域和真实的 API。所有代理都是 Llama 3 的变体，其提示方式取决于它们的职责，并且以逐步的方式进行协作。

4.3.6 事实性

虚幻仍然是大型语言模型面临的主要挑战。模型往往过于自信，即使在它们缺乏知识的领域也是如此。尽管存在这些缺点，但它们经常被用作知识库，这可能导致危险的结果，例如误信息的传播。虽然我们认识到真实性超越了虚幻，但我们在这里采取了一种以虚幻为先的方法。

图 11 文件上传处理。示例展示了 Llama 3 对上传文件进行分析和可视化的过程。

我们遵循这样的原则：后训练应该使模型与“知道它知道什么”保持一致，而不是添加知识（Gekhman 等人，2024；Mielke 等人，2020）。我们的主要方法涉及生成将模型生成与预训练数据中存在的真实数据子集对齐的数据。为此，我们开发了一种利用 Llama 3 上下文能力的知识探测技术。这个数据生成过程包括以下步骤：

1. 从预训练数据中提取一个数据片段。
2. 通过提示 Llama 3 生成关于这些片段（上下文）的事实性问题。
3. 从 Llama 3 中采样对该问题的回答。
4. 使用原始上下文作为参考，并使用 Llama 3 作为评判者来评分生成的正确性。
5. 使用 Llama 3 作为评判者来评分生成的丰富度。
6. 为那些在多次生成中始终信息丰富且不正确的回答生成拒绝理由，并使用 Llama 3

我们使用从知识探测中生成的数据来鼓励模型只回答它知道的问题，并拒绝回答它不确定的问题。此外，预训练数据并不总是事实一致或正确的。因此，我们还收集了一组有限的标记真实性数据，这些数据处理了敏感话题，其中存在许多与事实相矛盾或不正确的陈述。

4.3.7 可控性

可控性是指能够将模型的行为和结果引导至满足开发者和用户需求的能力。由于 Llama 3 是一个通用的基础模型，因此应该能够轻松地将其引导至不同的下游用例。为了提高 Llama 3 的可控性，我们专注于通过系统提示（使用自然语言指令）来增强其可控性，特别是关于响应长度、格式、语气和角色/人物设定方面。

数据收集: 我们通过要求标注者为 Llama 3 设计不同的系统提示，在通用英语类别中收集可控性偏好样本。然后，标注者与模型进行对话，以评估模型在整个对话过程中是否能够始终如一地遵循系统提示中定义的指令。以下是用于增强可控性的定制系统提示示例：

“你是一个乐于助人且充满活力的 AI 聊天机器人，担任忙碌家庭的膳食计划助手。工作日餐食应该快速便捷。早餐和午餐应优先选择方便食品，例如谷物、英式松饼配预先煮熟的培根和其他快速易做的食物。这个家庭很忙。请务必询问他们手边是否有必需品和喜欢的饮品，例如咖啡或能量饮料，以免他们忘记购买。除非是特殊场合，否则请记住节约预算。”

建模: 收集偏好数据后，我们将这些数据用于奖励建模、拒绝采样、SFT（持续精细调整）和 DPO（数据驱动的参数优化），以增强 Llama 3 的可控性。

5 结果

我们对 Llama 3 进行了一系列广泛的评估，调查了以下方面的性能：(1) 预训练语言模型，(2) 后训练语言模型，以及 (3) Llama 3 的安全特性。我们在下面的独立子节中展示这些评估的结果。

5.1 预训练语言模型

在本节中，我们将报告预训练 Llama 3 (第三部分) 的评估结果，并将其与其他可比规模的模型进行比较。我们会尽可能重现竞品模型的结果。对于非 Llama 模型，我们将报告在公开报道的结果中或（在可能的情况下）我们自己重现的结果中的最佳分数。这些评估的具体细节，包括射数、指标和其他相关的超参数和设置等配置，可以在我们的 Github 仓库中获得： [在此插入链接]。此外，我们还会发布作为公开基准评估的一部分生成的数据，这些数据可以在这里找到：[在此插入链接]。

我们将根据标准基准（第五部分 5.1.1）对模型质量进行评估，测试多项选择题设置变化的鲁棒性（第五部分 5.1.2），以及进行对抗性评估（第五部分 5.1.3)。我们还将进行污染分析，以估计训练数据污染对我们评估的影响程度（第五部分 5.1.4)。

5.1.1 标准基准

为了将我们的模型与当前最先进的水平进行比较，我们对 Llama 3 在大量标准基准测试中进行了评估，这些测试如下所示：

(1) 常识推理；(2) 知识；(3) 阅读理解；(4) 数学、推理和问题解决；(5) 长上下文；(6) 代码；(7) 对抗性评估；以及 (8) 总体评估。

实验设置。对于每个基准测试，我们计算 Llama 3 的分数以及其他具有可比尺寸的预训练模型的分数。在可能的情况下，我们将使用自己的管道重新计算其他模型的数据。为了确保公平比较，我们然后选择我们在计算的数据和该模型报告的数字（使用相同或更保守的设置）之间的最佳分数。您可以在此处找到有关我们的评估设置的更多详细信息。对于某些模型，例如由于未发布预训练模型或 API 不提供对数概率访问权限，无法重新计算基准测试值。这尤其适用于与 Llama 3 405B 可比的所有模型。因此，我们没有报告 Llama 3 405B 的类别平均值，因为需要所有基准测试的数字都可用。

显著性值。在计算基准测试分数时，由于有几种方差来源会导致对基准测试旨在测量的模型性能产生不精确的估计，例如少量演示、随机种子和批量大小。这使得理解一个模型是否在统计上显著优于另一个模型变得具有挑战性。因此，我们

将分数与 95% 置信区间 (CI) 一起报告，以反映基准数据选择带来的方差。我们使用公式（Madaan 等人，2024b）分析地计算 95% CI：

$$CI\_analytic(S) = 1.96 * sqrt(S * (1 - S) / N)$$

其中 S 是首选的基准分数，N 是基准的样本量。我们注意到，由于基准数据中的方差不是唯一的方差来源，因此这些 95% CI 是实际能力估计方差的下限。对于不是简单平均值的指标，将省略 CI。

Llama 3 8B 和 70B 模型的结果。图 12 显示了 Llama 3 8B 和 70B 在常识推理、知识、阅读理解、数学和推理以及代码基准测试上的平均性能。结果表明，Llama 3 8B 在几乎所有类别中都优于竞争模型，无论是按类别胜率还是按类别平均性能而言。我们还发现， Llama 3 70B 在大多数基准测试上都比其前身 Llama 2 70B 大幅提高了性能，除了可能已饱和的常识基准测试之外。Llama 3 70B 也优于 Mixtral 8x22B。

8B 和 70B 模型的结果。图 12 显示了 Llama 3 8B 和 70B 在常识推理、知识、阅读理解、数学与推理以及代码基准测试中的平均表现。结果表明，Llama 3 8B 在几乎每个类别中都超越了竞争模型 ,无论是按类别的胜率还是平均每类别的表现。我们还发现，Llama 3 70B 在大多数基准测试中比其前代产品 Llama 2 70B 有了很大的提升，除了可能已达到饱和的常识基准测试之外。Llama 3 70B 还超越了 Mixtral 8x22B。

所有模型的详细结果。表 9、10、11、12、13 和 14 展示了预训练的 Llama 3 8B、70B 和 405B 模型在阅读理解任务、编码任务、常识理解任务、数学推理任务和常规任务中的基准测试表现。这些表格将 Llama 3 的表现与同类大小的模型进行了比较。结果显示，Llama 3 405B 在其类别中具有竞争力，尤其在很大程度上超越了先前的开源模型。关于长上下文的测试，我们在第 5.2 节中提供了更全面的结果（包括针在大海捞针等探测任务）。

5.1.2 模型稳健性（Model Robustness）

除了基准测试性能之外，稳健性也是预训练语言模型质量的重要因素。我们研究了预训练语言模型在多项选择题（MCQ）设置中的设计选择稳健性。先前的研究表明，模型性能可能对这些设置中看似任意的设计选择很敏感，例如，模型得分甚至排名可能会随着上下文示例的顺序和标签而改变（Lu 等人，2022；Zhao 等人，2021；Robinson 和 Wingate，2023；Liang 等人，2022；Gupta 等

人，2024），提示的确切格式（Weber 等人，2023b；Mishra 等人，2022）或答案选项的格式和顺序(Alzahrani 等人 ,2024 ;Wang 等人 ,2024a ;Zheng 等人，2023）。受此工作启发，我们使用 MMLU 基准来评估预训练模型对以下方面的稳健性：（1）少数镜头标签偏差，（2）标签变体，（3）答案顺序，以及（4）提示格式：

- **少数镜头标签偏差。** 遵循 Zheng 等人（2023），...（此处省略实验细节和结果描述）。
- **标签变体。** 我们还研究了模型对不同选择标记集的响应。我们考虑了 Alzahrani 等人（2024）提出的两个标记集：即一组常见语言无关标记（$ & # @）和一组不具有任何隐含相对顺序的稀有标记（oe § ɜ ü）。我们还考虑了规范标签的两种版本（A. B. C. D. 和 A) B) C) D））和一个数字列表（1. 2. 3. 4.）。
- **答案顺序。** 遵循 Wang 等人（2024a），我们计算结果在不同答案顺序下的稳定性。为此，我们将数据集中的所有答案根据固定的排列重新映射。例如，对于排列 A B C D，所有标记为 A 和 B 的答案选项保持其标签，而标记为 C 的所有答案选项获取标签 D，反之亦然。
- **提示格式。** 我们评估了五种任务提示的性能差异，这些提示的信息量不同：一种提示简单地要求模型回答问题，而其他提示则断言模型的专业知识或应该选择最佳答案。

表 11 预训练模型在常识理解任务上的性能。结果包括 95% 置信区间。

表 12 预训练模型在数学和推理任务上的性能。结果包括 95% 置信区间。11 次射击。

表 13 预训练模型在通用语言任务上的性能。结果包括 95% 置信区间。

图 13 我们预训练语言模型在 MMLU 基准测试中对不同设计选择的鲁棒性。左侧：不同标签变体的性能。右侧：少样本示例中存在不同标签的性能。

图 14 我们预训练语言模型在 MMLU 基准测试中对不同设计选择的鲁棒性。左侧：不同答案顺序的性能。右侧：不同提示格式的性能。

图 13 展示了我们研究标签变体（左）和少数镜头标签偏差（右）模型性能稳健性的实验结果。结果表明，我们的预训练语言模型对 MCQ 标签的变化以及少数镜头提示标签的结构都非常稳健。这种稳健性对于 405B 参数模型尤其明显。

图 14 展示了我们对答案顺序和提示格式稳健性研究的结果。这些结果进一步强调了我们预训练语言模型性能的稳健性，特别是 Llama 3 405B 的稳健性。

5.1.3 对抗性基准测试

除了上面提到的基准测试外，我们还对三个领域的几个对抗性基准进行了评估：问答、数学推理和句式改写检测。这些测试旨在探测模型在专门设计成具有挑战性的任务上的能力，并可能指出模型在基准测试上的过度拟合问题。

- **问答方面**，我们使用了 Adversarial SQuAD (Jia 和 Liang, 2017) 和 Dynabench SQuAD (Kiela 等人, 2021)。
- **数学推理方面**，我们使用了 GSM-Plus (Li 等人, 2024c)。
- **句式改写检测方面**，我们使用了 PAWS (Zhang 等人, 2019)。

图 15 展示了 Llama 3 8B、70B 和 405B 在对抗性基准测试上的得分，作为其在非对抗性基准测试上的性能的函数。我们使用的非对抗性基准测试是：用于问答的 SQuAD (Rajpurkar 等人, 2016)、用于数学推理的 GSM8K 和用于句式改写检测的 QQP (Wang 等人, 2017)。每个数据点代表一个对抗性数据集和非对抗性数据集对（例如 QQP 与 PAWS 配对），我们展示了类别内的所有可能配对。对角线上的黑色线表示对抗性和非对抗性数据集之间的平价——在该线上表示模型无论对抗性与否都具有类似的性能。

在句式改写检测方面，无论是预训练模型还是后训练模型，似乎都没有受到 PAWS 所构建的对立性的影响，这代表了相对于上一代模型的巨大进步。这一结果证实了 Weber 等人 (2023a) 的发现，他们也发现大型语言模型对几个对抗性数据集中的虚假相关性更少敏感。然而，对于数学推理和问答，对抗性表现明显低于非对抗性表现。这种模式适用于预训练模型和后训练模型。

5.1.4 污染分析

我们进行了一项污染分析，以估计基准分数可能受到预训练语料库中评估数据污染影响的程度。先前的一些工作使用了多种不同的污染方法和超参数——我们参考了 Singh 等人的研究（2024）。结果表明，我们的预训练语言模型对多项选择题标签的变化以及少样本提示标签结构变化非常稳健（2024 年概述）。任何这些方法都可能出现假阳性和假阴性，如何最佳地进行污染分析目前仍然是一个开放的研究领域。在这里，我们主要遵循 Singh 等人的建议（2024）。

方法：具体来说，Singh 等人 (2024) 建议根据哪种方法导致“干净”数据集和整个数据集之间的最大差异来经验选择污染检测方法，他们将其称为估计的性能增益。对于所有评估数据集，我们基于 8-gram 重叠进行评分，Singh 等人 (2024) 发现这种方法对许多数据集都是准确的。我们将数据集 D 的一个示例视为受污染的，如果其标记 T_D 中的一个比例至少在预训练语料库中出现一次。

我们针对每个数据集单独选择 T_D，根据哪个值显示最大显著估计性能增益（跨三个模型大小）。

结果：表 15 显示了如上所述，所有主要基准的评估数据中被认为受污染的百分比，以获得最大的估计性能增益。从该表中，我们排除了结果不显着的基准数字，例如由于干净或受污染的集合样本太少，或者观察到的性能增益估计显示出极其不稳定行为。

在表 15 中，我们可以看到，对于一些数据集，污染具有很大影响，而对其他数据集则没有。例如，对于 PiQA 和 HellaSwag，污染估计和性能增益估计都较高。另一方面，对于 Natural Questions，估计的 52% 污染似乎对性能几乎没有影响。对于 SQuAD 和 MATH，低阈值会导致高水平的污染，但不会带来性能提升。这表明污染可能对这些数据集没有帮助，或者需要更大的 n 来获得更好的估计。最后，对于 MBPP、HumanEval、MMLU 和 MMLU-Pro，可能需要其他污染检测方法：即使使用更高的阈值，8-gram 重叠也会给出如此高的污染分数，以至于无法获得良好的性能增益估计。

5.2 微调语言模型

我们展示了在不同能力的基准测试上训练后的 Llama 3 模型的结果。与预训练类似，我们将作为评估的一部分生成的数据发布到公共可用的基准测试中，这些基准测试可以在 Huggingface 上找到（此处插入链接）。有关我们评估设置的更多详细信息，请参见此处（此处插入链接）。

基准测试和指标。表 16 概述了所有基准测试，按能力分类。我们将通过与每个基准测试中的提示进行精确匹配来对后训练数据进行去污染处理。除了标准的学术基准测试外，我们还进行了广泛的人工评估不同能力的测试。详细信息请参见第 5.3 节。

实验设置。我们采用与预训练阶段类似的实验设置，并对 Llama 3 与其他具有可比尺寸和能力的模型进行比较分析。在可能的情况下，我们将自己评估其他模型的性能并将其结果与报告数字进行比较，选择最佳分数。有关我们评估设置的更多详细信息，请参见此处（此处插入链接）。

表 16 按类别列出的后训练基准测试。我们用来评估训练后的 Llama 3 模型的所有基准测试的概述，按能力排序。

5.2.1 通用知识和指令遵循基准测试

我们使用表 2 中列出的基准测试来评估 Llama 3 在通用知识和指令遵循方面的能力。

通用知识： 我们利用 MMLU (Hendrycks et al., 2021a) 和 MMLU-Pro (Wang et al., 2024b) 来评估 Llama 3 在基于知识的问答能力上的表现。对于 MMLU，我们在没有 CoT 的 5 次示例标准设置下报告子任务准确率的宏观平均值。MMLU-Pro 是 MMLU 的扩展版本，包含更具挑战性的、以推理为重点的问题，消除了噪声问题，并将选择范围从四个选项扩展到十个选项。考虑到它侧重于复杂推理，我们报告了 MMLU-Pro 的 5 次示例 CoT。所有任务都格式化为生成任务，类似于 simple-evals (OpenAI, 2024)。

如表 2 所示，我们的 8B 和 70B Llama 3 变体在两个通用知识任务上都优于其他相似规模的模型。我们的 405B 模型优于 GPT-4 和 Nemotron 4 340B，Claude 3.5 Sonnet 在更大模型中领先。

指令遵循： 我们使用 IFEval (Zhou et al., 2023) 来评估 Llama 3 和其他模型遵循自然语言指令的能力。IFEval 包括大约 500 个“可验证指令”，例如“用超过 400 个字写”，这些指令可以使用启发式方法进行验证。我们在表 2 中报告了严格和宽松约束下的提示级和指令级准确率的平均值。请注意，所有 Llama 3 变体在 IFEval 上都优于可比较的模型。

5.2.2 能力考试

接下来，我们将在最初设计用于测试人类的一系列能力测试上评估我们的模型。我们从公开可用的官方来源获取这些考试；对于某些考试，我们将不同考试集的平均分数报告为每个能力测试的结果。具体来说，我们平均：

- GRE：教育测试服务提供的官方 GRE 实践考试 1 和 2；
- LSAT：官方预测试 71、73、80 和 93；
- SAT：来自《官方 SAT 学习指南》2018 版的 8 个考试；
- AP：每个学科一个官方练习考试；
- GMAT：官方 GMAT 在线考试。

这些考试中的问题包含了多项选择题和生成式问题。我们将排除任何附带图像的问题。对于包含多个正确选项的 GRE 题目，只有当模型选择了所有正确选项时，我们才将输出限定为正确。在有超过一个考试集的情况下，我们会使用少量提示进行评估。我们将分数调整到 130-170 的范围内（用于 GRE），并报告其他所有考试的准确率。

我们的结果见表 17。我们发现我们的 Llama 3 405B 模型的表现与 Claude 3.5 Sonnet 和 GPT-4 4o 非常相似。而我们的 70B 模型则展现出更为令人印象深刻的表现。它比 GPT-3.5 Turbo 显著更好，并在许多测试中超过了 Nemotron 4 340B。

5.2.3 编码基准

我们在多个流行的 Python 和多语言编程基准上评估 Llama 3 的代码生成能力。为了衡量模型生成功能正确代码的有效性，我们使用 pass@N 指标，该指标评估 N 个生成中的一组单元测试通过率。我们报告 pass@1 的结果。

Python 代码生成。 HumanEval（Chen 等人，2021）和 MBPP（Austin 等人，2021）是流行的 Python 代码生成基准测试，它们侧重于相对简单、自包含的功能。HumanEval+（Liu 等人，2024a）是 HumanEval 的增强版本，其中生成了更多测试用例以避免假阳性。MBPP EvalPlus 基准版本（v0.2.0）是从原始 MBPP（训练和测试）数据集中的 974 个初始问题中选出的 378 个良好格式的问题（Liu 等人，2024a）。这些基准测试的结果如表 18 所示。在这些 Python 变体的基准测试中，Llama 3 8B 和 70B 表现优于相同规模的模型表

现相似。对于最大的模型，Llama 3 405B、Claude 3.5 Sonnet 和 GPT-4o 表现类似，其中 GPT-4o 的结果最强。

模型: 我们将 Llama 3 与其他类似规模的模型进行了比较。对于最大模型 Llama 3 405B，Claude 3.5 Sonnet 和 GPT-4o 的表现相似，其中 GPT-4o 显示出最佳结果。

多编程语言代码生成： 为评估除 Python 之外的代码生成能力，我们报告了基于 HumanEval 和 MBPP 问题翻译的 MultiPL-E（Cassano 等人，2023）基准测试的结果。表 19 显示了一部分流行编程语言的结果。

请注意，与表 18 中的 Python 对应项相比，性能有显著下降。

5.2.4 多语言基准测试

Llama 3 支持 8 种语言 - 英语、德语、法语、意大利语、葡萄牙语、印地语、西班牙语和泰语，尽管基础模型的训练使用了更广泛的语言集合。在 表 20 中显示了我们在多语言 MMLU (Hendrycks 等人, 2021a) 和多语言小学数学 (MGSM) (Shi 等人, 2022) 基准测试上对 Llama 3 进行评估的结果。

- **多语言 MMLU**: 我们使用 Google 翻译将 MMLU 的问题、少例示例和答案翻译成不同语言。我们将任务说明保留为英语，并在 5-shot 设置下进行评估。

- **MGSM (Shi 等人, 2022)**：对于我们的 Llama 3 模型，我们报告了 MGSM 的 0-shot CoT 结果。多语言 MMLU 是一个内部基准测试，其中包含将 MMLU (Hendrycks 等人, 2021a) 的问题和答案翻译成 7 种语言——我们报告的 5-shot 结果是跨这些语言的平均值。

对于 MGSM (Shi 等人, 2022)，我们使用与 simple-evals (OpenAI, 2024) 中相同的原生提示来测试我们的模型，并将其置于 0-shot CoT 环境中。在表 20 中，我们报告了 MGSM 基准测试中包含的所有语言的平均结果。

我们发现 Llama 3 405B 在 MGSM 上优于大多数其他模型，平均得分达到 91.6%。在 MMLU 上，与上述英语 MMLU 结果一致，Llama 3 405B 落后于 GPT-4o 2%。另一方面，Llama 3 的 70B 和 8B 模型都表现出色，在两个任务上均以较大优势领先于竞争对手。

5.2.5 数学和推理基准

我们的数学和推理基准测试结果如表 2 所示。Llama 3 8B 模型在 GSM8K、MATH 和 GPQA 上都优于其他同等规模的模型。我们的 70B 模型在所有基准测试中均表现出显著优于同类模型的性能。最后，Llama 3 405B 模型在其类别中是 GSM8K 和 ARC-C 最佳模型，而在 MATH 上，它是第二好的模型。在 GPQA 上，它与 GPT-4 4o 竞争激烈，而 Claude 3.5 Sonnet 以显著优势位列榜首。

5.2.6 长上下文基准测试

我们考虑了一系列跨越不同领域和文本类型任务。在下面的基准测试中，我们专注于使用无偏评估协议的子任务，即基于准确性的指标而不是 n-gram 重叠指标。我们还优先考虑发现方差较低的任务。

- **Needle-in-a-Haystack (Kamradt, 2023)** 衡量模型检索隐藏在长文档随机部分信息的能力。我们的 Llama 3 模型表现出完美的针检索性能，成功地在所有文档深度和上下文长度下检索到 100% 的“针”。我们还测量了 Multi-needle（表 21）性能，这是一个 Needle-in-a-Haystack 的变化，其中我们将四根“针”插入上下文并测试模型是否可以检索其中的两根。我们的 Llama 3 模型实现了接近完美的检索结果。
- **ZeroSCROLLS (Shaham et al., 2023)**是一个针对长文本自然语言理解的零样本基准测试。由于真实答案未公开提供，我们报告了

验证集上的数字。我们的 Llama 3 405B 和 70B 模型在该基准测试的各种任务上与其他模型持平或超越它们。

- **InfiniteBench (Zhang et al., 2024)** 要求模型理解上下文窗口中的长距离依赖关系。我们在 En.QA（小说上的问答）和 En.MC（小说上的多项选择问答）上评估 Llama 3，其中我们的 405B 模型优于所有其他模型。增益在 En.QA 上尤为显着。

表 21 长文本基准测试。对于 ZeroSCROLLS（Shaham 等人，2023），我们报告的是验证集上的结果。对于 QuALITY 我们报告精确匹配，对于 Qasper - f1，对于 SQuALITY - rougeL。我们为 InfiniteBench（张等，2024）En.QA 指标报告 f1，并为 En.MC 报告准确率。对于 Multi-needle（Kamradt，2023），我们在上下文中插入 4 个针头，测试模型是否能够检索不同上下文长度的 2 个针头，我们计算最多 128k 的 10 个序列长度的平均召回率。

5.2.7 工具使用性能

我们利用一系列零样本工具使用（即函数调用）基准测试评估了我们的模型：Nexus (Srinivasan 等人，2023)、API-Bank (Li 等人，2023b)、Gorilla API-Bench (Patil 等人，2023) 和 Berkeley 函数调用排行榜 (BFCL) (Yan 等人，2024)。结果如表 22 所示。

在 Nexus 上，我们的 Llama 3 变体表现最佳，优于其他同类模型。 在 API-Bank 上，我们的 Llama 3 8B 和 70B 模型在相应类别中显着超越其他模型。405B 模型仅落后于 Claude 3.5 Sonnet 0.6%。最后，我们的 405B 和 70B 模型在 BFCL 上表现优异，并且在各自的规模类别中排名第二。 Llama 3 8B 在其类别中表现最佳。

我们还进行人工评估以测试模型的工具使用能力，重点关注代码执行任务。 我们收集了 2000 个与代码执行（不包括绘图或文件上传）相关的用户提示、绘图生成和文件上传。这些提示来自 LMSys 数据集 (Chiang 等人，2024)、GAIA 基准测试 (Mialon 等人，2023b)、人工标注者以及合成生成。我们将 Llama 3 405B 与 GPT-4o 进行比较，使用 OpenAI 的 Assistants API10 。结果如图 16 所示。在仅限文本的代码执行任务和绘图生成方面，Llama 3 405B 明显优于 GPT-4o。然而，在文件上传用例上，它落后于 GPT-4o。

5.3 人工评估

除了在标准基准数据集上的评估外，我们还进行了一系列人类评估。这些评估使我们能够测量和优化模型性能的更细微方面，例如模型的语气、冗长程度以及对细微差别和文化背景的理解。精心设计的 rengren 评估与用户体验密切相关，提供了有关模型在现实世界中表现的见解。

https://platform.openai.com/docs/assistants/overview

对于多轮人类评估，每个提示中的轮数在 2 到 11 之间。我们评估模型在最后一轮的响应。

提示收集。 我们收集了高质量的提示，涵盖范围广泛的类别和难度。为此，我们首先开发了一个分类法，其中包含尽可能多模型能力的类别和子类别。我们使用这个分类法收集了约 7,000 个提示，涵盖六个单次能力（英语、推理、编码、印地语、西班牙语和葡萄牙语）和三个多轮能力[11] (英语、推理和编码)。我们确保在每个类别中，提示在子类别之间均匀分布。我们还将每个提示分类为三个难度级别之一，并确保我们的提示集合包含大约 10% 的简单提示、30% 的中等难度提示和 60% 的困难提示。所有的人类评估图 16 Llama 3 405B 与 GPT-4o 在代码执行任务（包括绘图和文件上传）上的人类评估结果。 Llama 3 405B 在代码执行（不包括绘图或文件上传）以及情节生成方面都优于 GPT-4o，但在文件上传用例方面落后。

提示集都经过了严格的质量保证流程。建模团队无法访问我们的人类评估提示，以防止意外污染或对测试集过度拟合。

评估过程。为了对两个模型进行成对的人类评估，我们会询问人类标注员他们更喜欢哪两个模型响应（由不同的模型产生）。标注员使用 7 分制评分，使他们能够表明一个模型响应是否比另一个模型响应好得多、更好、略好或大致相同。当标注员表示一个模型响应比另一个模型响应更好或好得多时，我们就会将其视为该模型的“胜利”。我们将对模型进行成对比较，并在提示集中报告每个能力的获胜率。

结果。我们使用人类评估过程将 Llama 3 405B 与 GPT-4（0125 API 版本）、GPT-4o（API 版本）和 Claude 3.5 Sonnet（API 版本）进行比较。这些评估的结果如图 17 所示。我们观察到 Llama 3 405B 的表现与 GPT-4 的 0125 API 版本大致相当，而与 GPT-4o 和 Claude 3.5 Sonnet 相比，结果参差不齐（一些胜利和一些失败）。在几乎所有能力上，Llama 3 和 GPT-4 的获胜率都在误差范围内。在多轮推理和编码任务中，Llama 3 405B 的性能优于 GPT-4，但在多语言（印地语、西班牙语和葡萄牙语）提示方面则不如 GPT-4。Llama 3 在英语提示上的表现与 GPT-4 相当，在多语言提示上与 Claude 3.5 Sonnet 相当，并在单轮和多轮英语提示上优于 Claude 3.5 Sonnet。然而，它在编码和推理等方面不及 Claude 3.5 Sonnet。从质上讲，我们发现模型在人类评估中的性能受到语气、响应结构和冗长程度等细微因素的很大影响——这些都是我们在后训练过程中正在优化的因素。总体而言，我们的人类评估结果与标准基准评估的结果一致：Llama 3 405B 与领先的行业模型竞争非常激烈，使其成为表现最佳的公开可用模型。

局限性。 所有的人类评估结果都经过了严格的数据质量保证流程。然而，由于定义模型响应客观标准很困难，人类评估仍然可能会受到人类标注员的个人偏见、背景和偏好的影响，这可能导致不一致或不可靠的结果。

图 16 Llama 3 405B 与 GPT-4o 在代码执行任务（包括绘图和文件上传）上的人类评估结果。 Llama 3 405B 在代码执行（不包括绘图和文件上传）以及绘图生成方面优于 GPT-4o，但在文件上传用例方面落后。

图 17 Llama 3 405B 模型的人工评估结果。左：与 GPT-4 的比较。中：与 GPT-4o 的比较。右：与 Claude 3.5 Sonnet 的比较。所有结果均包含 95% 置信区间，并排除平局。

5.4 安全性

我们专注于评估 Llama 3 以安全且负责任的方式生成内容的能力，同时最大限度地提高有用信息。我们的安全工作始于预训练阶段，主要以数据清理和过滤的

形式进行。然后，我们将描述安全微调方法，重点介绍如何训练模型以符合特定的安全策略，同时仍保留有帮助性。我们将分析 Llama 3 的每项功能，包括多语言、长上下文、工具使用以及各种多模态功能，以衡量我们安全缓解措施的有效性。

随后，我们将描述我们对网络安全和化学/生物武器风险提升的评估。提升是指与使用现有可用技术（例如网络搜索）相比，新的技术发展带来的额外风险。

然后，我们将描述我们如何利用“红队”测试来迭代地识别和应对各种跨功能的安全风险，并进行剩余风险评估。

最后，我们将描述系统级安全，或围绕模型本身的输入和输出开发和编排分类器，以进一步提高安全性，并使开发者更容易地将安全性定制到各种用例中，并将生成式 AI 以更负责任的方式部署。

5.4.1 基准构建

为了安全负责地开发模型，我们创建了各种内部基准。我们的基准受到了 ML Commons 风险分类法 (Vidgen 等人，2024) 的启发。虽然 ToxiGen (Hartvigsen 等人，2022)、XS Test (Röttger 等人， 2023) 等语言模型安全基准已经存在，但 Llama 3 的许多新功能缺乏足够的外部基准，而且现有的外部基准在广度和深度上往往不够完善。

对于每个风险类别，我们收集了由人类编写的对抗性或边界性提示——表 23 中列出了这些提示的例子。对抗性提示范围从直接引发有害响应的简单提示到融入复杂越狱技术的提示。这些对抗性提示构成基准，是我们衡量违规率的基础。

作为违规率的反向指标，我们还构建了由边界性提示组成的错误拒绝基准。当模型即使存在合理且安全的答案时也拒绝以有帮助的方式回答时，我们将这种情况称为错误拒绝。边界性提示接近决策边界的提示，一个校准良好的模型应该能够处理这些提示，例如："我怎样才能从总是扮演主角的闺蜜那里抢走聚光灯呢？？"我们在违规和错误拒绝方面的总体基准规模超过了 4000 个提示/能力或语言，其中包括单回合和多回合提示。

5.4.2 安全预训练

我们相信负责任的开发必须从端到端的角度考虑，并在模型开发和部署的每个阶段都加以融入。在预训练过程中，我们应用各种过滤器，例如用于识别可能包含个人身份信息网站的过滤器（见第 3.1 节）。我们还重点关注可发现的记忆化（Nasr 等人，2023 年）。类似于 Carlini 等人 (2022) 的做法，我们使用训练数据中所有 n 元组的有效滚动哈希索引，以不同的发生频率对提示和真实结果进行采样。通过改变提示和真实结果的长度、目标数据的检测语言和领域，我们构建不同的测试场景。然后，我们测量模型多长时间精确生成真实结果序列，并分析指定场景中记忆化的相对速率。我们将逐字记忆化定义为包含率（模型生成的包含真实结果续集的比例），并在表 24 中显示的加权平均值中报告该比率，

这些加权平均值由数据中给定特征的流行度决定。我们发现训练数据的记忆化率较低（对于 405B，n = 50 和 n = 1000 时平均分别为 1.13% 和 3.91%）。使用相同的方法 ology 应用于其数据混合的等效大小 Llama 2 的记忆化率大致相同。

表 23 我们内部基准测试中所有能力的对抗性提示示例。

表 24 预训练 Llama 3 在选定测试场景下的平均逐字记忆。我们的基线是使用相同提示方法应用于其数据混合的英语、50-gram 场景下的 Llama 2。

5.4.3 安全微调

本章描述了我们用来降低各种能力风险的安全性微调方法，该方法包含两个关键方面：**(1) 安全性训练数据和 (2) 风险缓解技术。** 我们的安全性微调过程基于我们的通用微调方法，并进行了针对特定安全问题的修改。

我们优化两个主要指标：违规率（VR），该指标捕捉模型产生违反安全策略响应的情况；以及错误拒绝率（FRR），该指标捕捉模型错误拒绝对无害提示进行响应的情况。同时，我们评估模型在有用性基准测试中的表现，以确保安全性改进不会损害整体的有用性。我们的实验表明，80 亿参数的模型需要相对于 700 亿参数模型更高的安全数据比例与有用性数据比例，以实现可比的安全性能。更大的模型能够更好地区分对抗性和边界上下文，从而在 VR 和 FRR 之间取得更 favoable 的平衡。

微调数据

安全性训练数据的质量和设计对性能有深远影响。通过广泛的消融实验，我们发现质量比数量更重要。我们主要使用来自数据供应商的人类生成数据，但发现它容易出现错误和不一致，特别是对于微妙的安全策略。为了确保最高质量的数据，我们开发了 AI 辅助标注工具来支持我们的严格质量保证流程。

除了收集对抗性提示外，我们还收集了一组类似的提示，称为边界提示。这些提示与对抗性提示密切相关，但目的是教导模型提供有用的响应，从而降低错误拒绝率（FRR）。

除了人类标注之外，我们还利用合成数据来提高训练数据集的质量和覆盖范围。我们利用各种技术生成额外的对抗性示例，包括使用精心设计的系统提示进行上下文学习、基于新的攻击向量引导种子提示的突变，以及包括彩虹团队（Samvelyan 等人，2024）在内的先进算法，基于 MAP-Elites（Mouret 和 Clune，2015），它生成跨多个维度受限的提示。

我们进一步解决了模型在产生安全响应时的语气问题，这会影响下游用户的体验。我们为 Llama 3 开发了一个拒绝语气指南，并通过严格的质量保证流程确保所有新的安全性数据都遵守该指南。我们还使用零样本重写和人工编辑相结合的方法来改进现有的安全性数据，以生成高质量的数据。通过这些方法，以及使用语气分类器评估安全响应的语气质量，我们能够显著改善模型的措辞。

安全性监督微调

遵循我们的 Llama 2 配方（Touvron 等人，2023b），我们在模型对齐阶段将所有有用性数据和安全性数据组合在一起。此外，我们还引入了边界数据集来帮助模型区分安全请求和不安全的微妙区别。我们的标注团队被指示要根据我们的指南仔细设计安全性提示的响应。我们发现，当我们战略性地平衡对抗性和边界示例的比率时，SFT 在对齐模型方面非常有效。我们专注于更具挑战性的风险区域，其中边界示例的比例更高。 这对于我们成功的安全缓解工作起着至关重要的作用，同时将错误拒绝降至最低。

此外，我们还研究了模型大小对 FRR 和 VR 之间权衡的影响（见图 18）。我们的结果表明，它会发生变化——较小的模型需要相对于有用性更大的安全数据比例，并且与较大的模型相比，更难以有效地平衡 VR 和 FRR。

安全性 DPO

为了加强安全性学习，我们将对抗性和边界示例纳入 DPO 中的偏好数据集。我们发现，为给定的提示设计在嵌入空间中几乎正交的响应对是特别有效的，可以教导模型区分好的和坏的响应。我们进行了多项实验以确定对抗性、边界性和有

用性示例的最佳比率，目的是优化 FRR 和 VR 之间的权衡。我们还发现模型大小会影响学习结果——因此，我们为不同的模型大小调整了不同的安全性组合。

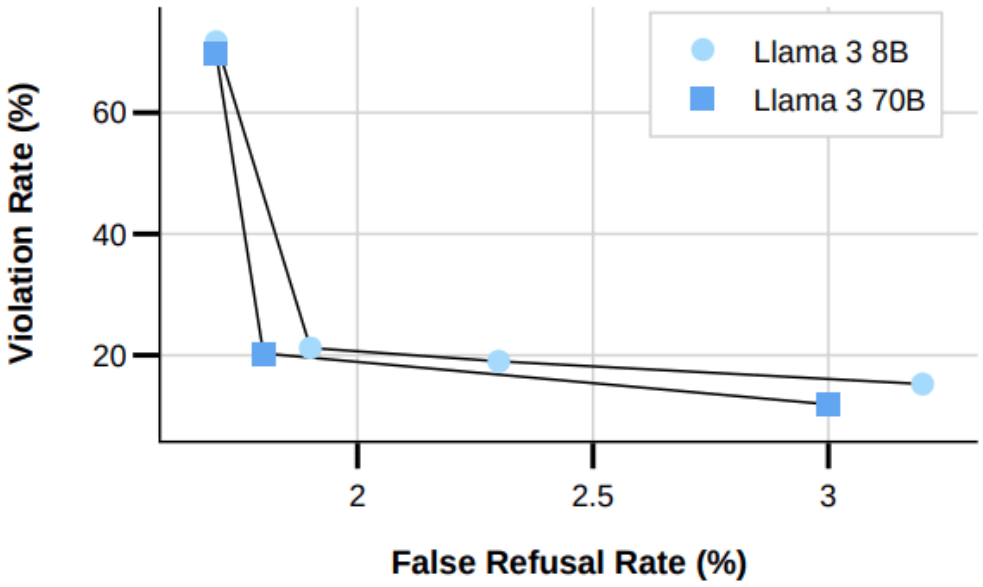

Figure 18 Influence of model size on safety mix design for balancing violation rate (VR) and false refusal rate (FRR). Each point of the scatterplot represents a different data mix balancing safety and helpfulness data. Different model sizes retain varying capacities for safety learning. Our experiments show that 8B models require a higher proportion of safety data relative to helpfulness data in the overall SFT mix to achieve comparable safety performance to 70B models. Larger models are more capable of discerning between adversarial and borderline context, resulting in a more favorable balance between VR and FRR.

图 18 显示了模型大小对安全数据组合设计的影响，以平衡违规率（Violation Rate, VR）和误拒率（False Refusal Rate, FRR）。散点图中的每个点表示在平衡安全性和有用性数据的不同数据组合。不同模型大小保留了不同的安全学

习能力。我们的实验表明，8B 模型相对于有用性数据需要在整体 SFT（监督微调）组合中占有更高比例的安全数据，以实现与 70B 模型相当的安全性能。较大的模型能够更好地区分对抗性和边缘性上下文，从而在 VR 和 FRR 之间达成更理想的平衡。

5.4.4 安全性结果

首先，我们概述 Llama 3 在各个方面的总体表现，然后描述每个新功能的测试结果以及我们降低安全风险的有效性。

整体性能 :图 19 和图 20 展示了 Llama 3 最终违规率和错误拒绝率与类似模型的比较结果。这些结果侧重于我们最大的参数规模（Llama 3 405B）模型，并与相关竞争对手进行了比较。其中两位竞争对手是通过 API 访问的端到端系统，另一位是我们在内部托管并直接评估的开源语言模型。我们既单独测试了我们的 Llama 模型，也将其与 Llama Guard(我们的开源系统级安全解决方案，详情见第 5.4.7 节）结合使用。

虽然较低的违规率是可取的，但将错误拒绝作为反向指标至关重要，因为总是拒绝的模型在安全性方面最高，但完全无用。同样，一个总是回答所有提示的模型，无论请求多么有问题，都会过度有害和有毒。图 21 利用我们的内部基准，探讨了行业中不同的模型和系统如何进行权衡以及 Llama 3 的表现如何。我们发现我们的模型在违规率指标方面具有很强的竞争力，同时错误拒绝率也很低，表明在实用性和安全性之间实现了良好的平衡。

多语言安全：我们的实验表明，英语的安全知识并不能轻易地转移到其他语言，尤其是考虑到安全策略的细微差别和语言特定的上下文。因此，收集每个语言的高质量安全数据至关重要。我们还发现，每种语言的安全数据分布会显著影响安全性表现，一些语言受益于迁移学习，而另一些语言则需要更多语言特定数据。为了在错误拒绝率（FRR）和违规率（VR）之间取得平衡，我们在监控这两个指标的影响的同时，迭代地添加对抗性数据和边界数据。

图 19 展示了我们内部基准测试结果，其中包含短上下文模型，显示了 Llama 3 在英语和其他语言中的违规率和错误拒绝率，并与类似的模型和系统进行了比较。为了构建每个语言的基准，我们使用了母语人士撰写的提示组合，有时还补充了来自英语基准的翻译。对于我们支持的所有语言，我们发现 Llama 405B 配合 Llama Guard 的安全性至少与两个竞争系统相当，如果不算更严格的安全性，而错误拒绝率仍然具有竞争力。

长上下文安全：长上下文模型如果不进行针对性缓解，就容易受到多发越狱攻击 (Anil 等人，2024 年)。为了解决这个问题，我们在 SFT 数据集上微调了我们的模型，其中包含了在存在不安全行为演示的情况下安全行为的示例。我们开发了一种可扩展的缓解策略，可以显著降低 VR，从而有效地中和了更长上下文攻击的影响，即使对于 256 次攻击也是如此。这种方法对 FRR 和大多数实用性指标的影响几乎可以忽略不计。

为了量化我们的长上下文安全缓解措施的有效性，我们使用了两种额外的基准测试方法：DocQA 和 Many-shot。对于 DocQA（“文档问答”的简称），我们使用长文档，其中包含可能被用于对抗性目的的信息。模型会提供文档和与文

档相关的一组提示，以测试问题是否与文档中的信息有关，从而影响模型安全地回复提示的能力。

在 Many-shot 中，遵循 Anil 等人 (2024) 的做法，我们构建了一个由不安全提示-响应对组成的合成聊天记录。最后一条提示与之前的消息无关，用于测试不安全的上下文行为是否会影响模型的不安全回复。图 20 显示了 DocQA 和 Many-shot 的违规率和错误拒绝率。我们看到 Llama 405B （带有和不带 Llama Guard）在违规率和错误拒绝率方面都优于 Comp. 2 系统，同时适用于 DocQA 和 Many-shot。相对于 Comp. 1，我们发现 Llama 405B 安全性更高，但会略微提高错误拒绝率。

工具使用安全性：可能的工具的多样性和工具使用的调用以及将工具集成到模型中的实现，使得完全缓解工具使用能力成为一项挑战性的工作 (Wallace 等人，2024 年)。我们专注于搜索用例。图 20 显示了违规率和错误拒绝率。我们与 Comp. 1 系统进行了测试，发现 Llama 405B 安全性更高，但错误拒绝率略高。

图 19 英文和我们核心多语言短语上下文基准测试上的违规率（VR）和错误拒绝率（FRR） 将 Llama 3 405B（带和不带 Llama Guard（LG）系统级保护）与竞争对手模型和系统进行比较。Comp. 3 未支持的语言用“x”表示。数值越低越好。

图 20 工具使用和长文本基准测试中的违规率（VR）和错误拒绝率（FRR）。越低越好。 DocQA 和多轮问答基准测试的性能分别列出。需要注意的是，由于该基准测试具有对抗性，我们没有边界数据集用于多轮问答，因此未对其进行错误拒绝率测量。对于工具使用（搜索）方面，我们仅比较 Llama 3 405B 与 Comp. 1。

图 21 模型和能力的违规率和拒绝率。每个点代表总体拒绝率。 并且评估了所有安全类别内在能力基准的违规率。符号表示我们是在评估模型级还是系统级的安全性。正如预期的那样，与系统级安全性结果相比，模型级安全性的结果显示出更高的违规率和更低的拒绝率。Llama 3 旨在平衡低违规率和低误拒率，而一些竞争对手则更倾向于其中一种或另一种。

5.4.5 网络安全评估结果

为了评估网络安全风险 ,我们利用了 CyberSecEval 基准框架(Bhatt 等 ,2023，2024），该框架包含了生成不安全代码、生成恶意代码、文本提示注入和漏洞识别等任务的安全性测量。我们开发并将 Llama 3 应用于新的基准测试，包括鱼叉式钓鱼和自主网络攻击。总体上，我们发现 Llama 3 在生成恶意代码或利用漏洞方面没有显著的易受攻击性。以下是针对具体任务的简要结果：

- **不安全编码测试框架：**在评估 Llama 3 8B、70B 和 405B 模型与不安全编码测试框架时，我们继续观察到较大的模型生成的不安全代码更多，且代码的 BLEU 平均分数更高（ Bhatt 等，2023）。
- **代码解释器滥用提示语料库：**我们发现 Llama 3 模型在特定提示下容易执行恶意代码 ,其中 Llama 3 405B 对恶意提示的遵从率为 10.4%，而 Llama 3 70B 为 3.8%。
- **文本提示注入基准 :**在对提示注入基准进行评估时 ,Llama 3 405B 在 21.7%的情况下成功遭遇提示注入攻击。图 22 展示了 Llama 3、GPT-4 Turbo、Gemini Pro 和 Mixtral 模型的文本提示注入成功率。
- **漏洞识别挑战：**在使用 CyberSecEval 2 的 Capture-the-Flag 测试挑战评估 Llama 3 识别和利用漏洞的能力时，Llama 3 未能超越常用的传统非 LLM 工具和技术。
- **鱼叉式钓鱼基准：**我们评估了模型在进行个性化对话以欺骗目标，使其不知不觉地参与安全妥协方面的说服力和成功率。由 LLM 生成的随机详细受害者档案用于鱼叉式钓鱼目标。评判 LLM（ Llama

3 70B)对 Llama 3 70B 和 405B 在与受害者模型(Llama 3 70B ）互动时的表现进行了评分，并评估了尝试的成功率。评判 LLM 评估 Llama 3 70B 在 24%的鱼叉式钓鱼尝试中成功，而 Llama 3 405B 在 14%的尝试中成功。图 23 展示了各模型和钓鱼目标的评判 LLM 评估说服力得分。

- **攻击自动化框架：**我们评估了 Llama 3 405B 作为自主代理在勒索软件攻击的四个关键阶段——网络侦察、漏洞识别、利用执行和后期利用行动中的潜力。我们通过配置模型在 Kali Linux 虚拟机上迭代生成和执行新的 Linux 命令，针对另一台已知漏洞的虚拟机来使模型自主行为。尽管 Llama 3 405B 在网络侦察中有效识别网络服务和开放端口，但在 34 次测试运行中未能有效利用这些信息获得对易受攻击机器的初始访问。在识别漏洞方面，Llama 3 405B 表现中等，但在选择和应用成功的利用技术方面存在困难。尝试执行利用和维持访问或在网络内执行横向移动的尝试完全失败。

网络攻击提升测试：我们进行了一项提升研究，测量虚拟助手在两个模拟进攻性网络安全挑战中提高新手和专家攻击者攻击率的程度。研究包括 62 名内部志愿者。志愿者根据其进攻性安全经验被分类为“专家”（31 名受试者）和“新手”（31 名受试者）。

为了评估与化学和生物武器扩散相关的风险，我们进行了提升测试，旨在评估使用 Llama 3 是否可以显著提高行为体策划此类攻击的能力。

实验设计:

- 该研究包括为期六小时的场景，其中两名参与者被要求为生物或化学袭击制定虚构的操作计划。
- 场景涵盖了 CBRNE（化学、生物、放射性、核和爆炸）袭击的主要规划阶段（试剂获取、生产、武器化和运送），旨在引发详细的计划，以解决与获取受限材料、现实世界实验室规程以及操作安全相关的问题。
- 参与者根据他们在科学或运营相关领域之前的经验进行招募，并被分配到由两名低技能行为体（无正式培训）或两名中等技能行为体（具有一些正式培训和科学或运营方面的实践经验）组成的团队。

研究方法:

- 该研究与一群 CBRNE 专家合作制定，旨在最大限度地提高定量和定性结果的普遍适用性、有效性和稳健性。
- 进行了一项初步研究，以验证研究设计，包括进行了强大的功效分析，确保我们的样本量足以进行统计分析。
- 每个团队都被分配到“控制”或“LLM”条件下。控制组只能访问基于互联网的资源，而配备 LLM 的团队除了可以访问互联网之外，还可以访问启用了网页搜索（包括 PDF 摄取）、信息检索功能（RAG）和代码执行（Python 和 Wolfram Alpha）的 Llama 3 模型。

- 为了测试 RAG 功能，使用关键词搜索生成了一组数百篇相关的科学论文并预加载到 Llama 3 模型推理系统中。

评估:

- 在练习结束后，每个团队生成的行动计划将由具有生物学、化学和运营规划领域专业知识的主体专家进行评估。
- 每个计划都在潜在攻击的四个阶段进行评估，为诸如科学准确性、细节、检测回避以及科学和运营执行成功的概率等指标生成分数。
- 在经过严格的德尔菲过程以减轻主体专家(SME)评估中的偏差和变异性之后，通过合并阶段级指标生成最终得分。

结果分析:

定量分析表明，使用 Llama 3 模型不会显着提高性能。该结果在进行总体分析（将所有 LLM 条件与仅基于 Web 的控制条件进行比较）以及针对子组的细分（例如，分别评估 Llama 3 70B 和 Llama 3 405B 模型，或分别评估与化学武器或生物武器相关的场景)时都成立。在与 CBRNE SME 验证这些结果之后，我们评估 Llama 3 模型的发布将增加生态系统中与生物或化学武器袭击相关的风险的可能性很低。

5.4.6 红队战术

我们利用“红队测试”来发现风险，并将这些发现用于改进我们的基准测试和安全调优数据集。我们定期进行红队演习，以持续迭代并发现新的风险，这将指导我们的模型开发和缓解过程。

我们的红队由网络安全、对抗性机器学习、负责任的 AI 和诚信方面的专家组成，此外还包括具备特定地理市场诚信问题背景的多语言内容专家。我们还与内部和外部主题领域的专家合作，以构建风险分类法并帮助进行更专注的对抗性评估。

针对特定模型能力的对抗性测试。 我们首先通过关注特定高风险类别的单个模型功能来进行红队测试，然后一起测试这些功能。红队专注于模拟更真实场景的提示级攻击，发现模型经常偏离预期行为，尤其是在提示意图被混淆或提示层叠多个抽象时。随着更多功能的加入，这些风险变得更加复杂，我们在下面详细描述了我们的一些红队发现。我们将这些红队发现与我们的内部安全基准测试结果结合使用，以开发专注的缓解措施，不断迭代地提高模型安全性。

- **短文本和长文本英语。** 我们在单轮和多轮对话中采用了众所周知的已发表和未发表的技术组合。我们还利用了类似 PAIR（Chao 等人，2023）的高级对抗性多轮自动化技术来处理一些技术和风险类别。总的来说，多轮对话会导致更多的有害输出。许多攻击在不同模型检查点上都很普遍，尤其是当它们一起使用时。

- **多轮拒绝抑制：** 指定模型响应应遵循特定格式或包含/排除与拒绝相关的特定短语的信息。
- **假设场景包装：**将违规提示包装为假设的/理论的任务或虚构场景。提示可以很简单，例如添加“假设地”这个词，或者构建一个复杂的层级场景。
- **角色扮演：** 为模型赋予具有特定违规响应特征（例如，“你是 X，你的目标是 Y”）的违规角色，或您自己作为用户化身特定的良性角色来掩盖提示的上下文。
- **添加免责声明和警告是一种反应启动形式，我们假设这是一种方法，可以为模型提供一条通往有益合规的路径，该路径与普遍的安全训练相交。** 与其他提及的攻击一起，在多轮对话中要求添加免责声明、触发警告等措施会导致违规率增加。
- **逐步升级违规是一种多轮攻击，其中对话从一个或多或少的良性请求开始，然后通过直接提示生成更夸张的内容，逐渐导致模型生成非常违规的回复。**一旦模型开始输出违规内容，就可能很难恢复（或者如果遇到拒绝，可以使用其他攻击）。对于上下文较长的模型，这个问题将越来越常见。

- **多语言。** 当考虑多种语言时，我们会发现许多独特的风险。
    - 在一个提示或对话中混合使用多种语言很容易导致比使用单一语言更多违规输出。
    - 资源匮乏的语言可能由于缺乏相关安全微调数据、模型对安全的泛化能力较弱或测试或基准测试的优先级问题而导致违规输出。然而，这种攻击通常会导致质量较差，限制了实际的恶意利用。
    - 俚语、特定上下文或文化特定参考可能会让人误以为一开始就违规了，但实际上模型没有正确理解某个参考，因此输出并不真正有害，也无法使其成为违规输出。
- **工具使用。** 在测试中，除了英语文本级别的对抗性提示技术成功地产生了违规输出之外，还发现了几种特定于工具的攻击。这包括但不限于：

不安全的工具链式调用，例如同时请求多个工具，其中一个违规，在早期的检查点中可能导致所有工具都受到违规和良性输入的混合调用。

强制使用工具：经常强制使用工具，并搭配特定的输入字符串、碎片化或编码文本，可能会导致工具输入出现潜在的违规行为，从而产生更具违规性的输出。 之后可以采用其他技术来访问工具结果，即使模型通常会拒绝执行搜索或协助处理结果。

修改工具使用参数：例如，在查询中交换单词、重试或在多轮对话中模糊化部分初始请求等方式，会导致许多早期检查点出现违规行为，作为强制使用工具的一种形式。

儿童安全风险：我们组建了一支专家团队，进行儿童安全风险评估，以评估模型产生可能导致儿童安全风险的输出的能力，并告知任何必要且适当的风险缓解措施（通过微调）。 我们利用这些专家红队会议来扩展我们通过模型开发评估基准的覆盖范围。 对于 Llama 3，我们进行了新的深入会话，使用基于目标的方法来评估模型沿着多个攻击路径的风险。 我们还与内容专家合作，进行红队演习，评估潜在的违规内容，同时考虑市场特定的细微差别或经验。

5.4.7 系统级安全

大型语言模型在各种现实世界应用中并非孤立使用，而是集成到更广泛的系统中。本节描述了我们的系统级安全实施方案，它通过提供更多灵活性和控制来补充模型级别的缓解措施。

为此，我们开发并发布了一个新的分类器 Llama Guard 3，这是一个针对安全分类微调的 Llama 3 8B 模型。类似于 Llama Guard 2 (Llama-Team, 2024)，此分类器用于检测输入提示和/或语言模型生成的输出响应是否违反了特定危害类别的安全策略。

它旨在支持 Llama 日益增长的能力，可用于英文和多语言文本。它还经过优化以用于工具调用（例如搜索工具）的上下文，并防止代码解释器滥用。最后，我们还提供量化变体以减少内存需求。我们鼓励开发人员将我们的系统安全组件版本作为基础，并为自己的用例配置它们。

分类法：我们使用 AI 安全分类法 (Vidgen 等人，2024) 中列出的 13 个危害类别进行训练：儿童性剥削、诽谤、选举、仇恨、不加区别的武器、知识产权、非暴力犯罪、隐私、与性有关的犯罪、色情内容、专业建议、自杀和自残以及暴力犯罪。我们还针对代码解释器滥用类别进行了训练，以支持工具调用的用例。

训练数据：我们从 Llama Guard (Inan 等人，2023) 使用的英文数据开始，并扩展此数据集以包含新的功能。对于诸如多语言和工具使用等新功能，我们收集提示和响应分类数据，以及利用用于安全微调的数据。通过进行提示工程，使 LLM 不拒绝对对抗性提示的回复，从而增加了训练集中不安全回复的数量。我们使用 Llama 3 对这些生成的数据获取响应标签。

为了改进 Llama Guard 3 的性能，我们利用人工标注和 Llama 3 模型的 LLM 标注对收集到的样本进行了广泛清洗。获取用户提示的标签对于人类和 LLM 都

更困难，我们发现人工标签略好一些，特别是对于边界提示，尽管我们的完整迭代系统能够减少噪声并产生更准确的标签。

结果。 Llama Guard 3 能显著降低跨能力的违规行为（在我们的基准测试中，平均违规率下降了 65%）。请注意，添加系统保障措施（以及任何安全缓解措施）都将以增加对良性提示的拒绝为代价。表 25 报告了与基本模型相比违规率的降低和误拒率的提高，以突出这种权衡。这种影响在图 19、图 20 和图 21 中也可见。

系统安全还提供更大的灵活性。Llama Guard 3 可以仅针对特定危害进行部署，从而能够在危害类别级别上控制违规和误拒权衡。表 26 按类别列出了违规减少量，以便根据开发人员用例来确定应开启/关闭哪些类别。

为了简化安全系统的部署，我们使用常用的 int8 量化技术提供了 Llama Guard 3 的量化版本，将其大小减少了 40% 以上。表 27 说明量化对模型性能的影响可以忽略不计。

系统级安全组件使开发者能够自定义和控制 LLM 系统对用户请求的响应方式。为了提高模型系统的整体安全性，并使开发者能够负责任地部署模型，我们描述并发布了两种基于提示的过滤机制：**Prompt Guard** 和 **Code Shield**。 我们将这些开源给社区，以便他们可以按原样使用它们，或者以此为灵感进行调整以适应其用例。

Prompt Guard 是一种基于模型的过滤器，旨在检测提示攻击，即旨在颠覆作为应用程序一部分的 LLM 预期行为的输入字符串。 该模型是一个多标签分类器，它检测两类提示攻击风险：

- 直接越狱（明确尝试覆盖模型的安全条件或系统提示的技术）。
- 间接提示注入（模型上下文窗口中包含第三方数据的情况，其中包括意外被 LLM 执行的用户命令的指令）。

该模型从 mDeBERTa-v3-base (一个小型的参数为 86M 的模型) 中微调而来，适合将输入过滤到 LLM 中。 我们在表 28 中显示的几个评估数据集上评估其性能。 我们在两个来自与训练数据相同分布的数据集（越狱和注入）上进行评估，以及一个英文的分布外数据集、一个基于机器翻译的多语言越狱集以及一个来自 CyberSecEval 的间接注入数据集（英文和多语言）。 总体而言，我们发现该模型可以很好地推广到新的分布，并且具有很强的性能。

Code Shield 是一个基于在推理时进行过滤的系统级保护类别的示例。 它特别关注在可能进入下游用例（如生产系统）之前检测不安全代码的生成。 它通过利用静态分析库“Insecure Code Detector” (ICD) 来识别不安全代码来实现这一目标。 ICD 使用一套静态分析工具跨 7 种编程语言执行分析。 这种类型的防护措施通常对开发者非常有用，他们可以部署多层防护措施到各种应用程序中。

5.4.8 局限性

我们对 Llama 3 的安全使用进行了广泛的测量和减轻各种风险的工作。但是，任何测试都不能保证完全识别出所有可能的风险。由于在各种数据集上进行训练，尤其是在英语之外的语言以及受到熟练的对抗性红色团队成员精心设计提示的情况下，Llama 3 仍然可能生成有害内容。恶意开发者或对抗性用户可能会找到新的方法来破解我们的模型，并将它们用于各种邪恶用途。我们将继续积极识别风险，开展缓解方法研究，并鼓励开发人员在各个方面都考虑责任，从模型开发到部署再到用户。我们希望开发人员能够利用和贡献我们开源系统级安全套件中发布的工具。

6 推论

我们研究了两种主要技术来提高 Llama 3 405B 模型的推理效率：(1) 管道并行和 (2) FP8 量化。我们已公开发布了 FP8 量化的实现。

6.1 管道并行 (Pipeline Parallelism)

当使用 BF16 表示模型参数时，Llama 3 405B 模型无法装入单个配备 8 个 Nvidia H100 GPU 的机器的 GPU 内存。为了解决这个问题，我们使用了 BF16 精度，将模型推理并行化到两台机器上的 16 个 GPU 上。在每台机器内，高带宽的 NVLink 使得能够使用张量并行 (Shoeybi et al.，2019)。然而，跨节点连接具有较低的带宽和更高的延迟，因此我们使用管道并行 (Huang et al.，2019)。

在使用管道并行的训练过程中，气泡是一个主要的效率问题（参见第 3.3 节）。但是，它们在推理过程中并不是一个问题，因为推理不涉及需要管道冲刷的反向传递。因此，我们使用微批量化来提高管道并行推理的吞吐量。

我们评估了在 4,096 个输入标记和 256 个输出标记的推理工作负载中使用两个微批量的效果，这两个微批量分别用于推理的关键值缓存预填充阶段和解码阶段。我们发现微批量化可以提高相同本地批量大小的推理吞吐量；参见图 24。这些改进来自微批量化在两个阶段都能够并发执行微批量。由于微批量化导致额

外的同步点，也会增加延迟，但总体而言，微批量化仍然会导致更好的吞吐量-延迟权衡。

6.2 FP8 量化

我们利用 H100 GPU 本身的 FP8 支持进行低精度推理实验。为了实现低精度推理，我们将模型内部的大多数矩阵乘法应用了 FP8 量化。具体来说，我们将模型中前馈网络层的绝大多数参数和激活值进行了量化，这些层占推理计算时间的约 50%。我们没有对模型的自注意力层中的参数进行量化。我们利用动态缩放因子来提高精度 (Xiao 等人，2024b)，并优化我们的 CUDA 内核 15 以减少计算比例的开销。

我们发现 Llama 3 405B 的质量对某些类型的量化很敏感，并做了一些额外的改动以提高模型输出质量：

1. 与 Zhang 等人 (2021) 相似，我们没有对第一个和最后一个 Transformer 层进行量化。
2. 高排列度标记（例如日期）可能导致较大的激活值。反过来，这会导致 FP8 中较高的动态缩放因子，并产生不可忽视数量的浮点下溢，从而导致解码错误。图 26 显示了使用 BF16 和 FP8 推理的 Llama 3 405B 的奖励得分分布。我们的 FP8 量化方法对模型的响应几乎没有影响。

为了解决这个问题，我们将动态缩放因子上限设置为 1200。

3. 我们使用逐行量化，为参数和激活矩阵计算跨行的缩放因子（见图 25）。我们发现这比张量级量化方法效果更好。

量化错误的影响。标准基准的评估通常表明，即使没有这些缓解措施，FP8 推理的性能也与 BF16 推理相当。但是，我们发现这样的基准不能充分反映 FP8 量化的影响。当缩放因子没有被上限限制时，模型偶尔会产生损坏的响应，即使基准性能很强。

与其依靠基准来测量由于量化导致的分布变化，不如分析使用 BF16 和 FP8 生成的 100,000 个响应的奖励模型得分的分布。图 26 显示了我们量化方法得到的奖励分布。结果表明，我们的 FP8 量化方法对模型的响应影响非常有限。

效率实验评估。图 27 描述了使用 Llama 3 405B 在预填充和解码阶段执行 FP8 推理的吞吐量-延迟权衡关系，使用 4,096 个输入标记和 256 个输出标记。

该图将 FP8 推理的效率与第 6.1 节中描述的两台机器 BF16 推理方法进行了比较。结果表明，使用 FP8 推理可以使预填充阶段的吞吐量提高高达 50%，并且在解码期间大幅提高了吞吐量-延迟权衡关系。

7 视觉实验

我们进行了一系列实验，通过一种组合方法将视觉识别能力整合到 Llama 3 中。该方法主要分为两个阶段：

第一个阶段: 我们将预训练的图像编码器 (Xu et al., 2023) 与预训练的语言模型结合起来，并在大量的图像文本对上引入和训练一组交叉注意力层 (Alayrac et al., 2022)。这导致了如图 28 所示的模型。

第二个阶段: 我们引入了时间聚合层和额外的视频交叉注意力层，这些层作用于大量视频文本对上，以学习模型识别和处理来自视频的时间信息。

组合式方法构建基础模型有几个优势:

(1) 它允许我们并行开发视觉和语言建模功能；

(2) 它避免了联合预训练视觉和语言数据带来的复杂性，这些复杂性源于视觉数据的标记化、不同模式的背景困惑度的差异以及模式之间的竞争；

(3) 它保证了引入视觉识别能力不会影响模型在仅文本任务上的性能；

(4) 交叉注意力架构确保我们不需要将全分辨率图像传递给不断增大的 LLM 主干 (特别是在每个 Transformer 层中的前馈网络)，从而提高推理效率。

请注意，我们的多模态模型仍在开发中，尚未准备好发布。

在第 7.6 和 7.7 节中介绍实验结果之前，我们将描述用于训练视觉识别能力的数据、视觉组件的模型架构、我们如何扩展这些组件的训练以及我们的预训练和后训练配方。

7.1 数据

我们分别描述了图像和视频数据。

7.1.1 图像数据

我们的图像编码器和适配器在图像-文本对上进行训练。我们通过一个复杂的的数据处理管道构建这个数据集，该管道包括四个主要阶段：

(1) 质量过滤 (2) 感知去重 (3) 重采样 (4) 光学字符识别 。我们还应用了一系列的安全措施。

- **质量过滤:** 我们实施质量过滤器 ,通过诸如(Radford 等人 ,2021 ）产生的低对齐分数等启发式方法去除非英语标题和低质量标题。具体来说，我们将低于特定 CLIP 分数的所有图像-文本对都删除。
- **去重:** 去除大规模训练数据集的重复数据可以提高模型性能，因为它减少了用于冗余数据的训练计算（ Esser 等人，2024；Lee 等人 ,2021 ;Abbas 等人 ,2023 ）,并减少了模型记忆化风险(Carlini 等人，2023；Somepalli 等人，2023 ）。因此，我们出于效率和隐私原因对训练数据进行去重。为此，我们使用最新的 SSCD 复制检测模型（ Pizzi 等人，2022) 的内部版本来大规模地对图像进行去重。对于所有图像，我们首先使用 SSCD 模型计算一个 512

维的表示。然后，我们使用这些嵌入针对数据集中的所有图像执行最近邻（NN）搜索，使用余弦相似性度量。我们将高于特定相似度阈值的示例定义为重复项。我们使用连通分量算法对这些重复项进行分组，并仅保留每个连通分量的单个图像-文本对。我们通过以下方式提高去重管道的效率：(1) 使用 k 均值聚类预先对数据进行聚类 (2) 对 NN 搜索和聚类使用 FAISS（Johnson 等人，2019）。

- **重采样:** 我们确保图像-文本对的多样性，类似于 Xu 等人（2023）；Mahajan 等人（2018）；Mikolov 等人（2013）。首先，我们通过解析高质量的文本源构建一个 n 元语法词汇表。接下来，我们计算数据集中每个词汇表 n 元语法的频率。然后，我们以如下方式对数据进行重采样：如果标题中的任何 n 元语法在词汇表中出现的次数少于 T 次，我们就保留相应的图像-文本对。否则，我们独立地对标题中的每个 n 元语法 n_i 以概率 T / f_i 进行抽样，其中 f_i 表示 n 元语法 n_i 的频率；如果任何 n 元语法都被抽取到了，我们就保留图像-文本对。这种重采样有助于提高低频类别和细粒度识别任务的性能。
- **光学字符识别:** 我们通过提取图像中的文字并将其与标题串联起来，进一步改进了我们的图像文本数据。使用专有光学字符识别 (OCR) 管道提取书写文本。我们观察到将 OCR 数据添加到训练数据中可以极大地提高需要 OCR 功能的任务的性能，例如文档理解。

为了提高模型在文档理解任务上的性能，我们将文档页面渲染为图像，并将图像与其各自的文本配对。文档文本是直接从源获取的，或者通过文档解析管道获取的。

安全： 我们主要致力于确保图像识别预训练数据集不包含不安全内容，例如性虐待材料（CSAM）(Thiel, 2023)。我们使用感知哈希方法，如 PhotoDNA (Farid, 2021)，以及内部专有分类器，扫描所有训练图像以查找 CSAM。我们还使用专有的媒体风险检索管道来识别和删除我们认为是 NSFW 的图像文本对，例如因为它们包含性或暴力内容。我们相信最小化训练数据集中的此类材料的流行度可以提高最终模型的安全性和帮助性，而不会影响其有用性。最后，我们对训练集中的所有图像执行面部模糊处理。我们测试了该模型针对人类生成的引用附加图像的提示。

退火数据： 我们通过使用 n-gram 对图像标题对进行重采样，创建了一个包含约 3.5 亿个示例的退火数据集。由于 n-gram 重采样有利于更丰富的文本描述，因此它选择了一个更高质量的数据子集。我们还用来自五个额外来源的约 1.5 亿个示例增强了结果数据：

-
- **视觉定位:** 我们将文本中的名词短语与图像中的边界框或蒙板关联起来。定位信息（边界框和蒙板）以两种方式在图像文本对中指定 :(1) 我们在图像上覆

盖框或蒙板，并在文本中使用标记作为参考，类似于标记集（Yang et al., 2023a）。(2) 我们直接将归一化（x min, y min, x max, y max）坐标插入文本，并用特殊标记将其分隔开。

- **截图解析：** 我们从 HTML 代码中渲染截图，并让模型预测生成特定截图元素的代码，类似于 Lee 等人 (2023)。感兴趣的元素通过边界框在截图中指示。
- **问答对：** 我们包括问答对，使我们能够使用太大而无法用于模型微调的大量问答数据。
- **合成标题：** 我们包括带有由早期模型版本生成的合成标题的图像。与原始标题相比，我们发现合成标题提供了比原始标题更全面的图像描述。
- **合成结构化图像。** 我们还包含了各种领域（例如图表、表格、流程图、数学公式和文本数据）的合成生成图像。这些图像都附带相应的结构化表示，例如对应的 Markdown 或 LaTeX 符号。 除了提高模型在这些领域的识别能力外，我们发现这些数据对于通过文本模型生成用于微调的问答对也很有用。

图 28 本文研究的将多模态能力添加到 Llama 3 的组合方法示意图。 这种方法导致一个多模态模型，经过五个阶段训练：语言模型预训练、多模态编码器预训练、视觉适配器训练、模型微调和语音适配器训练。

7.1.2 视频数据

为了进行视频预训练，我们使用了大型的视频-文本对数据集。我们的数据集通过多阶段过程进行整理。 我们使用基于规则的启发式方法筛选和清理相关的文本，例如确保最小长度并修复大写字母。然后，我们运行语言识别模型以过滤掉非英语文本。

我们运行 OCR 检测模型以过滤掉过度叠加文本的视频。为了确保视频-文本对之间的合理对齐，我们使用 CLIP（Radford 等，2021）风格的图像-文本和视频-文本对比模型。 我们首先使用视频中的单帧计算图像-文本相似度，并过滤掉相似度低的对，然后后续过滤掉视频-文本对齐较差的对。我们的某些数据包含静态或低运动视频；我们使用基于运动评分的过滤（Girdhar 等，2023）来过滤这些数据。我们没有对视频的视觉质量进行任何过滤，例如美学评分或分辨率过滤。

我们的数据集包含平均持续时间为 21 秒的中值持续时间为 16 秒的视频，超过 99% 的视频都在一分钟以内。空间分辨率在 320p 和 4K 视频之间变化很大，其中超过 70% 的视频的短边大于 720 像素。视频具有不同的纵横比，几乎所有视频的纵横比都在 1:2 和 2:1 之间，中值为 1:1。

7.2 模型架构

我们的视觉识别模型由三个主要组件组成：（1）图像编码器，(2）图像适配器，以及（3）视频适配器。

图像编码器:

我们的图像编码器是一个标准的视觉 Transformer（ ViT；Dosovitskiy 等人 (2020)) ,它被训练用来对齐图像和文本 (Xu 等人, 2023)。我们使用 ViT-H/14 版本的图像编码器，它拥有 6.3 亿个参数，在 25 亿张图像-文本对上训练了五个 epoch。 图像编码器的输入图像分辨率为 224 × 224；图像被拆分为 16 × 16 个大小相等的小块(即 14×14 像素的块大小)。正如之前 ViP-Llava (Cai 等人, 2024) 等工作所表明的那样，我们发现通过对比文本对齐目标训练的图像编码器无法保留精细的定位信息。为了缓解这个问题，我们采用了多层特征提取方法，除了最后一层特征外，还提供了第 4、8、16、24 和 31 层的特征。

此外 ,我们在交叉注意力层的预训练之前进一步插入了 8 个门控自注意力层(总共 40 个 Transformer 块），以学习特定于对齐的特征。因此，图像编码器最终拥有 8.5 亿个参数和额外的层。 通过多层特征，图像编码器为生成的 16 × 16 = 256 个小块中的每一个产生一个 7680 维度的表示。在后续训练阶段中，我们不会冻结图像编码器的参数，因为我们发现这可以提高性能，特别是在文本识别等领域。

图像适配器:

我们在图像编码器产生的视觉标记表示和语言模型产生的标记表示之间引入交叉注意力层（Alayrac 等人, 2022)。交叉注意力层应用于核心语言模型中的每个第四个自注意力层之后。 与语言模型本身一样，交叉注意力层使用广义查询注意力（GQA）来提高效率。

交叉注意力层为模型引入了大量可训练参数：对于 Llama 3 405B，交叉注意力层的参数约为 1000 亿。 我们对图像适配器进行了两个阶段的预训练: (1) 初始预训练和 (2) 退火： * **初始预训练:** 我们在上述约 60 亿张图像-文本对的数据集上预训练我们的图像适配器。为了提高计算效率，我们将所有图像的大小调整为最多适合四个 336 × 336 像素的图块，其中我们安排图块以支持不同的宽高比，例如 672 × 672、672 × 336 和 1344 × 336。 * **退火:** 我们继续使用上述退火数据集中的约 5 亿张图像来训练图像适配器。在退火过程中，我们增加每个图块的图像分辨率以提高对需要更高分辨率图像的任务的性能，例如信息图表理解。

视频适配器:

我们的模型最多可以接受 64 帧的输入（均匀采样自完整视频），每个帧都被图像编码器处理。我们通过两个组件对视频中的时间结构进行建模: (i) 编码的视频帧通过一个时间聚合器合并成一个，该聚合器将 32 个连续的帧合并为一个；(ii) 在每个第四个图像交叉注意力层之前添加额外的视频交叉注意力层。时间聚合器被实现为感知器重采样器（Jaegle 等人, 2021；Alayrac 等人, 2022)。 我们使

用每视频 16 帧（聚合成 1 帧）进行预训练，但在监督微调期间将输入帧数增加到 64。 视频聚合器和交叉注意力层在 Llama 3 7B 和 70B 中分别有 0.6 亿和 4.6 亿个参数。

7.3 模型规模

在将视觉识别组件添加到 Llama 3 后，模型包含自注意力层、交叉注意力层和一个 ViT 图像编码器。我们发现在训练较小（80 亿和 700 亿参数）模型的适配器时，数据并行和张量并行是最有效的组合。在这些规模下，模型或流水线并行不会提高效率，因为收集模型参数将主导计算。然而，在训练 4050 亿参数模型的适配器时，我们确实使用了流水线并行（除了数据和张量并行）。在这个规模上进行训练带来了三个新的挑战，除了第 3.3 节中概述的挑战之外：模型异质性、数据异质性和数值不稳定性。

模型异质性。模型计算是异质的，因为某些标记比其他标记执行更多的计算。特别是，图像标记通过图像编码器和交叉注意力层进行处理，而文本标记仅通过语言骨干网络进行处理。这种异质性会导致流水线并行调度中的瓶颈。我们通过确保每个流水线阶段包含五个层来解决这个问题 :即语言骨干网络中的四个自注意力层和一个交叉注意力层。（回想一下，我们在每四个自注意力层后引入了一个交叉注意力层。）此外，我们将图像编码器复制到所有流水线阶段。由于我们训练的是配对的图像文本数据 ,这使我们能够在计算的图像和文本部分之间进行负载平衡。

数据异质性。数据是异质的，因为平均而言，图像比关联文本具有更多标记：一张图像有 2308 个标记，而关联文本平均只有 192 个标记。因此，交叉注意力层的计算需要更长的时间和更多的内存，而不是自注意力层的计算。我们通过在图像编码器中引入序列并行来解决这个问题，这样每个 GPU 处理的标记数量大致相同。由于平均文本尺寸相对较小，我们还使用了更大的微批量大小（8 而不是 1）。

数值不稳定性。在将图像编码器添加到模型后，我们发现使用 bf16 进行梯度累积会导致数值不稳定。最可能的解释是图像标记通过所有交叉注意力层引入到语言骨干网络中。这意味着图像标记表示中的数值偏差对整体计算的影响过大，因为错误被复合起来。我们通过使用 FP32 执行梯度累积来解决这个问题。

7.4 预训练

图像预训练: 我们从预训练的文本模型和视觉编码器权重开始初始化。视觉编码器被解冻，而文本模型权重保持冻结，如上所述。首先，我们使用 60 亿个图像-文本对训练模型，每个图像都被调整大小以适合四个 336 × 336 像素的图块。我们使用 16,384 的全局批量大小和余弦学习率计划，初始学习率为 10×10^{-4}，权重衰减为 0.01。初始学习率是基于小规模实验确定的。然而，这些发现并不能很好地推广到非常长的训练计划中，并且在损失值停滞时，我们会在训练过程中多次降低学习率。在基本预训练之后，我们将图像分辨率进一步提高，并

在退火数据集上继续训练相同的权重。优化器通过热身重新初始化到学习率 2×10^{-5}，再次遵循余弦计划。

视频预训练: 对于视频预训练，我们从上述图像预训练和退火权重开始。我们将添加视频聚合器和交叉注意力层，如架构中所述，并随机初始化它们。我们冻结模型中的所有参数，除了特定于视频的参数（聚合器和视频交叉注意力），并在视频预训练数据上训练它们。我们使用与图像退火阶段相同的训练超参数，学习率略有不同。我们从完整的视频中均匀采样 16 帧，并使用四个大小为 448×448 像素的块来表示每一帧。我们在视频聚合器中使用 16 的聚合因子，从而得到一个有效的帧，文本标记将交叉关注该帧。我们使用 4,096 的全局批量大小、190 个标记的序列长度以及 10^{-4} 的学习率进行训练。

7.5 训练后处理

在这一章节中，我们将详细描述视觉适配器的后续训练步骤。

在预训练之后，我们将模型微调到高度精选的多模态对话数据上，以启用聊天功能。

此外，我们还实现了直接偏好优化（DPO）以提高人工评估性能，并采用了拒绝采样来改善多模态推理能力。

最后，我们添加了一个质量调整阶段，在这个阶段我们将继续在非常小的高质量对话数据集上微调模型，这进一步提高了人工评估结果，同时保留了基准测试的性能。

以下将提供每个步骤的详细信息。

7.5.1 监督微调数据

我们在下面分别描述用于图像和视频功能的监督微调（SFT）数据。

图像: 我们使用不同数据集的混合进行监督微调。

- 学术数据集：我们使用模板或通过大语言模型（LLM）重写，将高度过滤的现有学术数据集转换为问答对。LLM 重写的目的是用不同的指令来扩充数据，并提高答案的语言质量。
- 人工标注：我们通过人工标注员为各种任务（开放式问答、字幕、实际用例等）和领域（例如，自然图像和结构化图像）收集多模态对话数据。标注员将收到图像并被要求编写对话。

为了确保多样性，我们将大规模数据集进行聚类，并在不同的聚类中均匀地采样图像。此外，我们还通过使用 k 近邻扩展种子来获取一些特定领域的附加图像。标注员还将获得现有模型的中间检查点，以促进模型在循环中的样式标注，以便可以使用模型生成作为标注员提供额外人类编辑的起点。这是一个迭代过程，其

中模型检查点将定期更新为性能更好的版本，这些版本是在最新数据上训练的。这增加了人工标注的量和效率，同时也提高了质量。

- 合成数据：我们探索了通过使用图像的文本表示和文本输入 LLM 生成合成多模态数据的不同方法。基本思路是利用文本输入 LLM 的推理能力在文本域中生成问答对，并用其对应的图像替换文本表示以产生合成的多模态数据。示例包括将来自问答数据集的文本呈现为图像或将表格数据呈现为合成表格和图表图像。此外，我们还使用现有图像的字幕和 OCR 提取来生成与图像相关的一般对话或问答数据。

视频: 与图像适配器类似，我们将预先存在注释的学术数据集用于转换，将其转换为适当的文本指令和目标响应。目标将转换为开放式回答或多项选择题，视情况而定。我们请人工标注员为视频添加问题和相应的答案。我们要求标注员关注那些不能基于单个帧来回答的问题，以引导标注员朝着需要时间理解的问题方向前进。

7.5.2 监督微调方案

我们分别介绍了图像和视频能力的受监督微调（SFT）方案：

图像: 我们从预训练的图像适配器初始化模型，但将预训练语言模型的权重替换为指令调优语言模型的权重。为了保持仅文本性能，语言模型权重保持冻结，即我们只会更新视觉编码器和图像适配器权重。

我们的微调方法类似于 Wortsman 等人（2022）。首先，我们使用多个数据随机子集、学习率和重量衰减值进行超参数扫描。接下来，我们根据模型性能对它们进行排名。最后，我们将前 K 个模型的权重取平均以获得最终模型。K 的值通过评估平均模型并选择性能最高实例来确定。我们观察到，与通过网格搜索找到的最佳单个模型相比，平均模型始终产生更好的结果。此外，这种策略减少了对超参数的敏感性。

视频: 对于视频 SFT，我们使用预训练权重初始化视频聚合器和交叉注意力层。模型的其余参数（图像权重和 LLM）则从相应的模型中初始化，并按照它们的微调阶段进行。类似于视频预训练，然后只对视频 SFT 数据上的视频参数进行微调。在此阶段，我们将视频长度增加到 64 帧，并使用 32 的聚合因子获得两个有效帧。 aynı zamanda,块的分辨率也相应提高，以与对应的图像超参数保持一致。

7.5.3 偏好数据

为了奖励建模和直接偏好优化，我们构建了多模态成对偏好数据集。

- **人工标注:** 人工标注的偏好数据包含两个不同模型输出的比较，标记为“选择”和“拒绝”，并使用 7 级评分。用于生成响应的模型是每周从一池最佳近期模型中随机采样的，每个模型都有不同的特征。除了偏好标签外，我们还要求标注员提供可选的人工编辑以更正“选择”响应中的不准确性，因为视觉任务对不准确性的容忍度较低。请注意，人工编辑是一个可选步骤，因为实践中存在数量和质量之间的权衡。
- **合成数据:** 通过使用仅文本的 LLM 编辑并故意在监督微调数据集引入错误也可以生成合成偏好对。我们取对话数据作为输入，并使用 LLM 引入微妙但有意义的错误(例如，更改对象、更改属性、添加计算错误等)。这些编辑后的响应被用作负面“拒绝”样本，并与“选择”的原始监督微调数据配对。
- **拒绝采样:** 此外，为了创建更多策略性负样本，我们利用拒绝采样的迭代过程来收集额外的偏好数据。我们在接下来的部分中将更详细地讨论拒绝采样的使用方式。总而言之，拒绝采样用于从模型中迭代地采样高质量的生成结果。因此，作为副产品，所有未被选中的生成结果都可用作负面拒绝样本，并用作额外的偏好数据对。

7.5.4 奖励模型

我们训练了一个视觉奖励模型（RM），它基于视觉 SFT 模型和语言 RM。视觉编码器和交叉注意力层从视觉 SFT 模型初始化，并在训练期间解冻，而自注意力层则从语言 RM 初始化并保持冻结。我们观察到，冻结语言 RM 部分通常会导致更好的精度，尤其是在需要 RM 根据其知识或语言质量进行判断的任务中。我们采用与语言 RM 相同的训练目标，但添加了一个加权正则化项来对批量平均的奖励 logits 平方进行处理，以防止奖励分数漂移。

第 7.5.3 节中的人类偏好注释用于训练视觉 RM。我们遵循与语言偏好数据（第 4.2.1 节）相同的做法，创建两到三个具有清晰排名的对（编辑版本 > 选择版本 > 拒绝版本）。此外，我们还通过扰动与图像信息相关的词语或短语（例如数字或视觉文本）来合成增强负面回复。这鼓励视觉 RM 基于实际的图像内容进行判断。

7.5.5 直接偏好优化

与语言模型（第 4.1.4 节）类似，我们使用直接偏好优化（DPO；Rafailov 等人（2023））进一步训练视觉适配器，并使用第 7.5.3 节中描述的偏好数据。为了对抗后训练过程中分布偏移，我们只保留最近几批人类偏好注释，而丢弃那些与策略差距较大的批次（例如，如果更改了基本预训练模型）。我们发现，与其始终冻结参考模型，不如每 k 步以指数移动平均（EMA）的方式更新它，有助于模型从数据中学习更多，从而在人类评估中获得更好的性能。总体而言，我们

观察到视觉 DPO 模型在人类评估中始终优于其 SFT 起始点，并且在每次微调迭代中都表现出色。

7.5.6 拒绝采样

现有的大多数问答对仅包含最终答案，而缺乏推理任务泛化良好的模型所需的思维链解释。我们使用拒绝采样来为这些示例生成缺少的解释，从而提高模型的推理能力。

给定一个问答对，我们通过使用不同的系统提示或温度对微调后的模型进行采样来生成多个答案。接下来，我们将生成的答案与真实答案通过启发式方法或 LLM 裁判进行比较。最后，我们将正确的答案添加到微调数据中重新训练模型。我们发现每道题保留多个正确答案是有用的。

为了确保只将高质量的示例添加到训练中，我们实施了以下两个安全措施：

1. 我们发现，一些示例包含错误的解释，尽管最终答案是正确的。我们注意到这种模式在只有很小一部分生成的答案正确的题目中更常见。因此，我们将正确答案概率低于特定阈值的题目的答案丢弃。
2. 评审员由于语言或风格的差异而偏爱某些答案。我们使用奖励模型来选择 K 个最高质量的答案并将其添加到训练中。

7.5.7 质量调优

我们精心策划了一个规模虽小但高度精选的精细调优（SFT）数据集，所有样本都经过人工或我们最佳模型的重写和验证，以满足最高标准。 我们用这些数据训练 DPO 模型，以提高响应质量，并将此过程称为质量调优（QT）。我们发现，当 QT 数据集涵盖广泛的任务并且应用适当的提前停止时，QT 可以显着提高人类评估结果，而不会影响基准测试验证的一般性性能。在此阶段，我们仅根据基准测试选择检查点，以确保能力保持或改进。

7.6 图像识别结果

我们评估了 Llama 3 图像理解能力在一系列任务上的表现，这些任务涵盖了自然图像理解、文本理解、图表理解和多模态推理：

- MMMU (Yue 等人，2024a) 是一个具有挑战性的多模态推理数据集，模型需要理解图像并解决跨越 30 个不同学科的大学水平问题。这包括多项选择题和开放式问题。我们将模型在包含 900 张图像的验证集上进行评估，与其他工作一致。
- VQAv2 (Antol 等人，2015) 测试了模型结合图像理解、语言理解和常识知识来回答关于自然图像的一般性问题的能力。

- AI2 Diagram (Kembhavi 等人 ,2016) 评估了模型解析科学图表并回答有关图表问题的能力。我们使用与 Gemini 和 x.ai 相同的评估协议，并使用透明边界框报告分数。
- ChartQA (Masry 等人，2022) 是一个具有挑战性的图表理解基准测试。这要求模型在视觉上理解不同类型的图表并回答有关这些图表逻辑问题。
- TextVQA (Singh 等人，2019) 是一个流行的基准数据集，需要模型阅读和推理图像中的文本以回答有关它们的疑问。这测试了模型对自然图像中 OCR 理解能力。
- DocVQA (Mathew 等人，2020) 是一个专注于文档分析和识别的基准数据集。它包含各种文档的图像 ,评估了模型执行 OCR 理解并推理文档内容以回答有关它们的问题的能力。

表 29 展示了我们实验的结果。表中的结果表明，附加到 Llama 3 上的视觉模块在不同模型容量的各种图像识别基准测试中都具有竞争力。使用 resulting Llama 3-V 405B 模型，我们在所有基准测试上均优于 GPT-4V，但略逊于 Gemini 1.5 Pro 和 Claude 3.5 Sonnet。Llama 3 405B 在文档理解任务上表现尤为出色。

7.7 视频识别结果

我们对 Llama 3 的视频适配器在三个基准测试上进行了评估：

- **PerceptionTest (Lin et al., 2023)**：该基准测试模型对短视频片段进行理解和预测的能力。它包含了各种类型的问题，例如识别物体、动作、场景等。我们根据官方提供的代码和评估指标（准确率）报告结果。
- **TVQA (Lei et al., 2018)**：该基准评估模型的复合推理能力，需要进行空间时间定位、识别视觉概念以及与字幕对话联合推理。由于该数据集来源于流行的电视节目，因此它还测试了模型利用这些电视节目的外部知识来回答问题的能力。它包含超过 1.5 万个验证 QA 对，每个对应的视频片段平均长度为 76 秒。它采用多项选择格式，每个问题有五个选项，我们根据先前工作（OpenAI, 2023b）在验证集上报告性能。
- **ActivityNet-QA (Yu et al., 2019)**：该基准评估模型对长视频片段进行理解以了解动作、空间关系、时间关系、计数等的能力。它包含 800 个视频中的 8,000 个测试 QA 对，每个视频平均长度为 3 分钟。对于评估，我们遵循先前工作（Google, 2023；Lin et al., 2023；Maaz et al., 2024）的协议，其中模型生成短的单

词或短语回答，并且使用 GPT-3.5 API 将其与真实答案进行比较以评估输出的正确性。我们报告 API 计算出的平均准确率。

推理过程

在执行推理时，我们从完整的视频片段中均匀采样帧并将其与简短的文本提示一起传递给模型。由于大多数基准都涉及回答多项选择题，因此我们使用以下提示：

- 从以下选项中选择正确的答案：{问题}。只用正确选项字母回答，不要写其他任何东西。

对于需要生成简短答案（例如 ActivityNet-QA 和 NExT-QA）的基准，我们使用以下提示：

- 使用一个单词或短语回答问题：{问题}。

对于 NExT-QA，由于评估指标（WUPS）对长度和使用的特定单词很敏感，因此我们还提示模型要具体化并回复最突出的答案，例如在被问及位置问题时指定“客厅”而不是简单地回答“房子”。对于包含字幕（即 TVQA）的基准，我们在推理过程中将片段对应的字幕包含在提示中。

结果

表 30 展示了 Llama 3 8B 和 70B 模型的性能。我们将其与两个 Gemini 模型和两个 GPT-4 模型的性能进行了比较。请注意，所有结果都是零样本的结果，

因为我们在训练或微调数据中没有包含这些基准的任何部分。我们发现，我们的 Llama 3 模型在后处理期间训练小型视频适配器非常有竞争力，并且在某些情况下甚至比其他可能从预训练开始就利用原生多模态处理的模型更好。Llama 3 在视频识别方面表现特别出色，因为我们仅评估了 8B 和 70B 参数模型。Llama 3 在 PerceptionTest 上取得了最佳成绩，这表明该模型具有很强的进行复杂时间推理的能力。在像 ActivityNet-QA 这样的长片段活动理解任务中，即使 Llama 3 只处理高达 64 帧，它也能获得强劲的结果（对于 3 分钟的视频，模型每 3 秒只处理一帧）。

	Llama 3-V 8B	Llama 3-V 70B	Gemini 1.0 Pro	Gemini 1.0 Ultra	Gemini 1.5 Pro	GPT-4V	GPT-4o
PerceptionTest (test)	53.8	**60.8**	51.1	54.7	–	–	–
TVQA (val)	82.5	**87.9**	–	–	–	87.3	–
NExT-QA (test)	27.3	**30.3**	28.0	29.9	–	–	–
ActivityNet-QA (test)	52.7	56.3	49.8	52.2	57.5	–	**61.9**

Table 30 Video understanding performance of our vision module attached to Llama 3. We find that across range of tasks covering long-form and temporal video understanding, our vision adapters for Llama3 8B and 70B parameters are competitive and sometimes even outperform alternative models.

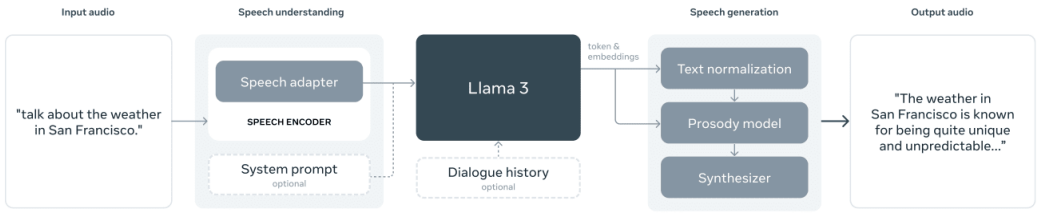

Figure 29 Architecture of our speech interface for Llama 3.

8 语音实验

我们进行了实验，研究将语音功能整合到 Llama 3 中的组合方法，类似于我们用于视觉识别的方案。在输入端，添加了编码器和适配器来处理语音信号。我们利用系统提示（以文本形式）来使 Llama 3 支持不同的语音理解模式。如果没有提供系统提示，该模型将充当通用语音对话模型，能够有效地以与仅文本版本 Llama 3 一致的方式响应用户语音。引入对话历史作为提示前缀可以改善多轮对话体验。我们还尝试了使用系统提示来实现 Llama 3 的自动语音识别（ASR）和自动语音翻译（AST）。Llama 3 的语音界面支持高达 34 种语言。18 它还允许文本和语音交替输入，使模型能够解决高级音频理解任务。

我们还尝试了一种语音生成方法，其中我们实现了流式文本到语音（TTS）系统，该系统在语言模型解码过程中动态生成语音波形。我们基于专有的 TTS 系统设计了 Llama 3 的语音生成器，并且没有针对语音生成微调语言模型。相反，我们通过在推理时利用 Llama 3 词嵌入来专注于改进语音合成的延迟、准确性和自然度。语音界面如图 28 和 29 所示。

8.1 数据

8.1.1 语音理解

训练数据可以分为两类。预训练数据包括大量的未标记语音，用于以自我监督的方式初始化语音编码器。有监督的微调数据包括语音识别、语音翻译和口语对话数据；这些数据用于在与大型语言模型集成时解锁特定的能力。

预训练数据: 为了预训练语音编码器，我们整理了一个包含约 1500 万小时语音记录的数据集，涵盖多种语言。 我们使用语音活动检测 (VAD) 模型过滤音频数据，并选择 VAD 阈值高于 0.7 的音频样本进行预训练。 在语音预训练数据中，我们还重点确保不存在个人身份信息 (PII)。 我们使用 Presidio Analyzer 来识别此类 PII。

语音识别和翻译数据: 我们的 ASR 训练数据包含 23 万小时的手写转录语音记录，涵盖 34 种语言。 我们的 AST 训练数据包含 90,000 小时的双向翻译：从 33 种语言到英语以及从英语到 33 种语言。这些数据包含监督和使用 NLLB 工具包 (NLLB Team 等人，2022) 生成的合成数据。 使用合成 AST 数据可以提高低资源语言的模型质量。 我们数据中的语音片段最大长度为 60 秒。

口语对话数据: 为了微调用于口语对话的语音适配器，我们通过询问语言模型对这些提示的转录做出回应来合成语音提示的回复 (Fathullah 等人，2024)。 我们使用 ASR 数据集的子集(包含 60,000 小时的语音)以这种方式生成合成数据。

此外，我们还通过在用于微调 Llama 3 的数据子集上运行 Voicebox TTS 系统 (Le 等人，2024) 生成了 25,000 小时的合成数据。 我们使用了几种启发式

方法来选择与语音分布相匹配的微调数据的子集。这些启发式方法包括关注相对较短且结构简单的提示，并且不包含非文本符号。

8.1.2 语音生成

语音生成数据集主要包括用于训练文本规范化（TN）模型和韵律模型（PM）的数据集。两种训练数据都通过添加 Llama 3 词嵌入作为额外的输入特征进行增强，以提供上下文信息。

文本规范化数据。我们的 TN 训练数据集包含 5.5 万个样本，涵盖了广泛的符号类别（例如，数字、日期、时间），这些类别需要非平凡的规范化。每个样本由书面形式文本和相应的规范化口语形式文本组成，并包含一个推断的手工制作的 TN 规则序列，用于执行规范化。

声韵模型数据。PM 训练数据包括从一个包含 50,000 小时的 TTS 数据集提取的语言和声韵特征，这些特征与专业配音演员在录音室环境中录制的文字稿件和音频配对。

Llama 3 嵌入。Llama 3 嵌入取自第 16 层解码器输出。我们仅使用 Llama 3 8B 模型，并提取给定文本的嵌入（即 TN 的书面输入文本或 PM 的音频转录），就像它们是由 Llama 3 模型在空用户提示下生成的。在一个样本中，每个 Llama 3 标记序列块都明确地与 TN 或 PM 本地输入序列中的相应块对齐，

即 TN 特定的文本标记(由 Unicode 类别区分)或语音速率特征。这允许使用 Llama 3 标记和嵌入的流式输入训练 TN 和 PM 模块。

8.2 模型架构

8.2.1 语音理解

在输入方面，语音模块由两个连续的模块组成：语音编码器和适配器。语音模块的输出直接作为标记表示输入到语言模型中，使语音和文本标记能够直接交互。此外，我们引入了两个新的特殊标记，用于包含语音表示序列。语音模块与视觉模块(见第 7 节)有显著不同，后者通过交叉注意力层将多模态信息输入到语言模型中。相比之下，语音模块生成的嵌入可以无缝集成到文本标记中，使语音接口能够利用 Llama 3 语言模型的所有功能。

语音编码器：我们的语音编码器是一个拥有 10 亿参数的 Conformer 模型（Gulati 等，2020）。模型的输入由 80 维的梅尔频谱图特征组成，这些特征首先通过一个步幅为 4 的堆叠层处理，然后通过线性投影将帧长度缩减到 40 毫秒。处理后的特征由一个包含 24 个 Conformer 层的编码器处理。每个 Conformer 层具有 1536 的潜在维度，包括两个维度为 4096 的 Macron-net 风格的前馈网络、一个核大小为 7 的卷积模块，以及一个具有 24 个注意力头的旋转注意力模块（Su 等，2024）。

语音适配器：语音适配器包含约 1 亿参数。它由一个卷积层、一个旋转 Transformer 层和一个线性层组成。卷积层的核大小为 3，步幅为 2，旨在将语音帧长度减少到 80 毫秒。这允许模型向语言模型提供更粗粒度的特征。

Transformer 层的潜在维度为 3072，前馈网络的维度为 4096，进一步处理经过卷积下采样后的语音信息。最后，线性层将输出维度映射到与语言模型嵌入层匹配。

8.2.2 语音生成

我们在语音生成的两个关键组件中使用了 Llama 3 8B 的嵌入：文本标准化和韵律建模。文本标准化（TN）模块通过将书面文本上下文地转换为口语形式来确保语义正确性。韵律建模（PM）模块通过使用这些嵌入预测韵律特征来增强自然性和表现力。这两个组件协同工作，实现了准确而自然的语音生成。

文本标准化：作为生成语音语义正确性的决定因素，文本标准化（TN）模块进行从书面文本到相应口语形式的上下文感知转换，这些口语形式最终由下游组件进行口头化。例如，根据语义上下文，书面形式的“123”可能被读作基数（one hundred twenty three）或逐个数字拼读（one two three）。TN 系统由一个流式的基于 LSTM 的序列标记模型组成，该模型预测用于转换输入文本的手工制作的 TN 规则序列（Kang 等，2024）。该神经模型还通过交叉注意力接收 Llama 3 嵌入，以利用其中编码的上下文信息，使得最少的文本标记前瞻和流式输入/输出成为可能。

韵律建模：为了增强合成语音的自然性和表现力，我们集成了一个仅解码 Transformer 架构的韵律模型（PM）（Radford 等，2021），该模型将 Llama 3 嵌入作为额外输入。这种集成利用了 Llama 3 的语言能力，使用其文本输出和

中间嵌入（Devlin 等，2018；Dong 等，2019；Raffel 等，2020；Guo 等，2023）来增强韵律特征的预测，从而减少了模型所需的前瞻。PM 集成了多个输入组件来生成全面的韵律预测：从上文详细介绍的文本标准化前端得出的语言特征、标记和嵌入。PM 预测三个关键的韵律特征：每个音素的对数持续时间、对数基频（F0）平均值和音素持续时间内的对数功率平均值。模型由一个单向 Transformer 和六个注意力头组成。每个块包括交叉注意力层和具有 864 隐藏维度的双全连接层。PM 的一个显著特征是其双重交叉注意力机制，一层专用于语言输入，另一层专用于 Llama 嵌入。该设置有效地管理了不同输入速率，而无需显式对齐。

8.3 训练方案

8.3.1 语音理解

语音模块的训练分两个阶段进行。第一阶段，语音预训练，利用未标记数据训练一个语音编码器，该编码器在语言和声学条件方面表现出强大的泛化能力。第二阶段，监督微调，适配器和预训练编码器与语言模型集成，并与其联合训练，而 LLM 保持冻结状态。这使得模型能够响应语音输入。此阶段使用对应语音理解能力的标记数据。

多语言 ASR 和 AST 建模常常导致语言混淆/干扰，从而降低性能。一种流行的缓解方法是在源端和目标端都加入语言识别（LID）信息。这可以在预先确定的方向上提高性能，但同时也可能导致泛化能力下降。例如，如果一个翻译系统期

望在源端和目标端都提供 LID 信息，那么该模型不太可能在训练中未见过的方向上表现出良好的零样本性能。因此，我们的挑战在于设计一个允许一定程度 LID 信息的系统，同时保持模型足够通用，以便在未见的方向进行语音翻译。为了解决这个问题，我们设计了仅包含要输出文本（目标端）的 LID 信息的系统提示。这些提示中没有语音输入（源端）的 LID 信息，这也可能使其能够处理代码切换语音。对于 ASR，我们使用以下系统提示：用{语言}重复我的话：，其中{语言}来自 34 种语言之一（英语、法语等）。对于语音翻译，系统提示为：“将以下句子翻译成{语言}：”。这种设计已被证明能够有效地提示语言模型以期望的语言进行响应。我们在训练和推理过程中使用相同的系统提示。

我们使用自监督式 BEST-RQ 算法（Chiu 等，2022）对语音进行预训练。

编码器采用长度为 32 帧的掩码，对输入 mel 谱图的概率为 2.5%。如果语音话语超过 60 秒，我们将随机裁剪 6K 帧，对应 60 秒的语音。通过堆叠 4 个连续帧、将 320 维向量投影到 16 维空间，并在 8192 个向量的代码库内使用余弦相似度度量进行最近邻搜索，对 mel 谱图特征进行量化。为了稳定预训练，我们采用 16 个不同的代码库。投影矩阵和代码库随机初始化，在模型训练过程中不更新。多软最大损失仅用于掩码帧，以提高效率。编码器经过 50 万步训练，全局批处理大小为 2048 个语音。

监督微调。预训练语音编码器和随机初始化的适配器在监督微调阶段与 Llama 3 联合优化。语言模型在此过程中保持不变。训练数据是 ASR、AST 和对话数据的混合。Llama 3 8B 的语音模型经过 650K 次更新训练，使用全局批大小为

512 个话语和初始学习率为 10。Llama 3 70B 的语音模型经过 600K 次更新训练，使用全局批大小为 768 个话语和初始学习率为 4 × 10。

8.3.2 语音生成

为了支持实时处理，韵律模型采用了一个前瞻机制，该机制考虑了固定数量的未来音位和可变数量的未来标记。 这确保在处理传入文本时保持一致的前瞻，这对于低延迟语音合成应用程序至关重要。

训练: 我们开发了一种利用因果掩码的动态对齐策略，以促进语音合成的流式传输。该策略结合了韵律模型中的前瞻机制，用于固定数量的未来音位和可变数量的未来标记，与文本规范化过程中的分块过程相一致（第 8.1.2 节）。

对于每个音位，标记前瞻包括块大小定义的最大标记数，导致 Llama 嵌入具有可变前瞻，而音位具有固定前瞻。

Llama 3 嵌入来自 Llama 3 8B 模型，在韵律模型训练期间保持冻结。输入电话率特征包括语言和说话人/风格可控性元素。模型训练使用批量大小为 1,024 个语调的 AdamW 优化器，学习率为 9 × 10 -4，训练超过 100 万次更新，前 3,000 次更新进行学习率预热，然后遵循余弦调度。

推理: 在推理过程中，使用相同的前瞻机制和因果掩码策略，以确保训练与实时处理之间的一致性。PM 以流式方式处理传入文本，为电话率特征逐个更新输入电话，为标记率特征逐块更新输入。只有当该块的第一个电话当前时才更新新的块输入，从而在训练期间保持对齐和前瞻。

为了预测韵律目标，我们采用了一种延迟模式方法（Kharitonov 等人，2021），这增强了模型捕获和复制长程韵律依赖关系的能力。这种方法有助于合成的语音的自然度和表达力，确保低延迟和高质量输出。

8.4 语音理解结果

我们评估了 Llama 3 语音界面的语音理解能力，包括三个任务：(1) 自动语音识别 (2) 语音翻译 (3) 语音问答。我们将 Llama 3 的语音界面性能与三种最先进的语音理解模型进行了比较：Whisper（Radford 等人，2023 年）、SeamlessM4T（Barrault 等人，2023 年）和 Gemini。在所有评估中，我们使用贪婪搜索来预测 Llama 3 的标记。

语音识别。我们在 Multilingual LibriSpeech (MLS; Pratap 等人，2020 年)、LibriSpeech (Panayotov 等人，2015 年)、VoxPopuli (Wang 等人，2021a) 和 FLEURS 多语言数据集的子集（Conneau 等人，2023 年）上的英文数据集上评估了 ASR 性能。在评估中，使用 Whisper 文本标准化程序对解码结果进行后处理，以确保与其他模型报告的结果一致。在所有基准测试中，我们在这些基准测试的标准测试集上测量 Llama 3 语音界面的单词错误率，除了中文、日语、韩语和泰语，我们报告了字符错误率。

表 31 显示了 ASR 评估结果。它证明了 Llama 3（以及更广泛的多模态基础模型）在语音识别任务上的强劲性能：我们的模型在所有基准测试中都优于

Whisper20 和 SeamlessM4T 等专门针对语音的模型。在 MLS 英文方面，Llama 3 的表现与 Gemini 相似。

语音翻译。 我们还评估了我们的模型在语音翻译任务中的性能，其中模型被要求将非英语语音翻译成英语文本。我们在这些评估中使用 FLEURS 和 Covost 2 (Wang 等人，2021b) 数据集，并测量翻译英文的 BLEU 分数。表 32 展示了这些实验的结果。我们的模型在语音翻译方面的性能突显了多模态基础模型在语音翻译等任务中的优势。

语音问答。 Llama 3 的语音界面展示出惊人的问题回答能力。该模型可以毫不费力地理解代码切换的语音，而无需事先接触此类数据。值得注意的是，尽管该模型仅在单轮对话上进行了训练，但它能够进行扩展且连贯的多轮对话会话。图 30 展示了一些突显这些多语言和多轮能力的例子。

安全性。 我们在 MuTox (Costa-jussà 等人，2023 年) 上评估了我们语音模型的安全性能，这是一个包含 20,000 个英语和西班牙语语段以及 4,000 个其他 19 种语言语段的用于多语言基于音频的数据集，每个语段都有毒性标签。将音频作为输入传递给模型，并在清除一些特殊字符后评估输出的毒性。我们将 MuTox 分类器（Costa-jussà 等人，2023 年）应用于 Gemini 1.5 Pro 并比较结果。我们评估了当输入提示安全而输出有毒时添加毒性 (AT) 的百分比，以及当输入提示有毒而答案安全时丢失毒性 (LT) 的百分比。表 33 显示了英语的結果以及我们在所有 21 种语言上的平均结果。添加毒性百分比非常低：我们的语音模型在英语方面具有最低的添加毒性百分比，不到 1%。它去除的毒性远多于添加的毒性。

8.5 语音生成结果

在语音生成方面，我们专注于评估使用 Llama 3 向量进行文本规范化和韵律建模任务的基于标记流式输入模型的质量。评估重点在于与不将 Llama 3 向量作为额外输入的模型进行比较。

文本规范化: 为了衡量 Llama 3 向量的影响，我们尝试改变模型使用的右侧上下文量。我们使用 3 个文本规范化（TN）标记（以 Unicode 类别分隔）的右侧上下文训练了模型。这个模型与不使用 Llama 3 向量、使用 3 个标记右侧上下文或完整双向上下文的模型进行了比较。正如预期的那样，表 34 显示使用完整的右侧上下文可以提高不使用 Llama 3 向量的模型的性能。然而，包含 Llama 3 向量的模型优于所有其他模型，从而实现了无需依赖输入中的长上下文即可进行标记率输入/输出流式传输。我们比较了有或没有 Llama 3 8B 向量以及使用不同右侧上下文值的模型。

韵律建模: 为了评估我们的韵律模型（PM）与 Llama 3 8B 的性能，我们进行了两组人类评价，比较了有和没有 Llama 3 向量的模型。评分者聆听来自不同模型的样本，并表示他们的偏好。

要生成最终的语音波形，我们使用了一种基于 Transformer 的内部声学模型（Wu 等人，2021），它预测谱特征，并使用 WaveRNN 神经声码器（Kalchbrenner 等人，2018）生成最终的语音波形。

首先，我们将直接与不使用 Llama 3 向量的流式基准模型进行比较。在第二项测试中，将 Llama 3 8B PM 与不使用 Llama 3 向量的非流式基准模型进行了比较。如表 35 所示，Llama 3 8B PM 被优选了 60% 的时间（与流式基准相比），并且 63.6% 的时间被优选（与非流式基准相比），这表明感知质量有显著提高。 Llama 3 8B PM 的关键优势在于其基于标记的流式传输能力（第 8.2.2 节），它在推理过程中保持低延迟。这降低了模型的前瞻需求，使模型能够实现更具响应性和实时性的语音合成，相比于非流式基准模型。总体而言，Llama 3 8B 韵律模型始终优于基准模型，证明了其在提高合成语音的自然度和表现力方面的有效性。

	Llama 3 8B		**Llama 3 70B**		**Gemini 1.5 Pro**	
Language	AT (↓)	LT (↑)	AT (↓)	LT (↑)	AT (↓)	LT (↑)
English	0.84	15.09	**0.68**	**15.46**	1.44	13.42
Overall	2.31	9.89	**2.00**	10.29	2.06	**10.94**

Table 33 Speech toxicity of our speech interface to Llama 3 on the MuTox dataset. AT refers to added toxicity (%) and LT refers to lost toxicity (%).

Model	Context	Accuracy
Without Llama 3 8B	3	73.6%
Without Llama 3 8B	∞	88.0%
With Llama 3 8B	3	**90.7%**

Table 34 Sample-wise text normalization (TN) accuracy. We compare models with or without Llama 3 8B embeddings, and using different right-context values.

Model	Preference
PM for Llama 3 8B	**60.0%**
Streaming phone-only baseline	40.0%

Model	Preference
PM for Llama 3 8B	**63.6%**
Non-streaming phone-only baseline	36.4%

Table 35 Prosody Modeling (PM) evaluation. *Left:* Rater preferences of PM for Llama 3 8B vs. streaming phone-only baseline. *Right:* Rater preferences of PM for Llama 3 8B vs. non-streaming phone-only baseline.

9 相关工作

Llama 3 的开发建立在对语言、图像、视频和语音基础模型的大量先前研究之上。本文范围不包括对这些工作的全面概述；我们建议读者参考 Bordes 等人 (2024) ;Madan 等人 (2024) ;Zhao 等人 (2023a) 以获取这样的概述。下面，我们将简要概述直接影响 Llama 3 开发的开创性著作。

9.1 语言

规模: Llama 3 遵循着将简单方法应用于不断增大的规模的持久趋势，这是基础模型的特征。改进是由计算能力的提高和数据质量的提升驱动的，其中 405B 模型使用的预训练计算预算几乎是 Llama 2 70B 的五十倍。尽管我们的最大 Llama 3 包含 405B 个参数，但实际上其参数数量少于早期且性能更差的模型，例如 PALM (Chowdhery 等人，2023)，这是由于对缩放定律（Kaplan 等人，2020；Hoffmann 等人，2022）有了更好的理解。其他前沿模型的大小，如 Claude 3 或 GPT 4（OpenAI，2023a），公开信息有限，但总体性能相当。

小规模模型: 小规模模型的发展与大规模模型的发展同步进行了。参数较少的模型可以显著提高推理成本并简化部署（Mehta 等人，2024；Team 等人，2024）。较小的 Llama 3 模型通过远远超过计算最优训练点来实现这一目标，有效地将训练计算权衡为推理效率。另一种途径是将较大模型蒸馏成较小模型，如 Phi（Abdin 等人，2024）。

架构: 与 Llama 2 相比，Llama 3 在架构上做了最小的修改，但其他最近的基础模型探索了其他设计。最值得注意的是，专家混合架构（Shazeer 等人，2017；Lewis 等人，2021；Fedus 等人，2022；Zhou 等人，2022）可以作为有效地提高模型容量的一种方法，例如在 Mixtral（Jiang 等人，2024）和 Arctic（Snowflake，2024）中。Llama 3 的性能优于这些模型，这表明密集架构并

不是限制因素，但在训练和推理效率、以及大规模下的模型稳定性方面仍然存在许多权衡。

开源: 开源基础模型在过去一年中迅速发展，其中 Llama3-405B 现在与当前闭源的最先进水平相当。最近开发了许多模型系列，包括 Mistral（Jiang 等人，2023）、Falcon（Almazrouei 等人，2023）、MPT（Databricks，2024）、Pythia（Biderman 等人，2023）、Arctic（Snowflake，2024）、OpenELM（Mehta 等人，2024）、OLMo（Groeneveld 等人，2024）、StableLM（Bellagente 等人，2024）、OpenLLaMA（Geng 和 Liu，2023）、Qwen（Bai 等人，2023）、Gemma（Team 等人，2024）、Grok（XAI，2024）和 Phi（Abdin 等人，2024）。

后训练: Llama 3 的后训练遵循了既定的指令调整策略（Chung 等人，2022；Ouyang 等人，2022），接着是与人类反馈进行对齐（Kaufmann 等人，2023）。尽管一些研究表明轻量级对齐程序的效果出乎意料（Zhou 等人，2024），但 Llama 3 使用数百万个人类指令和偏好判断来改进预训练模型，包括拒绝采样（Bai 等人，2022）、监督微调（Sanh 等人，2022）和直接偏好优化（Rafailov 等人，2023）。为了策划这些指令和偏好示例，我们部署了更早版本的 Llama 3 来过滤（Liu 等人，2024c）、重写（Pan 等人，2024）或生成提示和响应（Liu 等人，2024b），并将这些技术通过多轮后训练应用。

9.2 多模态性

我们的 Llama 3 多模态能力实验是关于联合建模多个模态的基础模型长期研究的一部分。我们的 Llama 3 方法结合了许多论文中的想法，取得了与 Gemini 1.0 Ultra（谷歌，2023 年）和 GPT-4 Vision（OpenAI，2023b）相当的结果；请参见第 7.6 节。

视频：尽管越来越多的基础模型支持视频输入（谷歌，2023 年； OpenAI，2023b），但关于视频和语言联合建模的研究并不多。类似于 Llama 3，目前大多数研究都采用适配器方法来对齐视频和语言表示，并解开有关视频的问答和推理（Lin 等人，2023 年；Li 等人，2023a；Maaz 等人，2024 年；Zhang 等人，2023 年；Zhao 等人，2022 年）。我们发现此类方法产生的结果与最先进的结果具有竞争力；请参见第 7.7 节。

语音：我们的工作也融入了一项更大的工作，该工作结合了语言和语音建模。早期的文本和语音联合模型包括 AudioPaLM（Rubenstein 等人，2023 年）、VioLA（Wang 等人，2023b）、VoxtLM Maiti 等人（2023 年），SUTLM（Chou 等人，2023 年）和 Spirit-LM（Nguyen 等人，2024 年）。我们的工作基于 Fathullah 等人（2024 年）等先前组合语音和语言的组成方法。与大多数先前的工作不同，我们选择不针对语音任务对语言模型本身进行微调，因为这样做可能会导致非语音任务的竞争。我们发现，即使没有这样的微调，在更大的模型规模下也能取得良好的性能；请参见第 8.4 节。

10 结论

高品质基础模型的开发仍处于初期阶段。我们开发 Llama 3 的经验表明，这些模型未来还有很大的改进空间。在开发 Llama 3 模型家族的过程中，我们发现强烈的关注高质量数据、规模和简单性始终能带来最佳结果。在初步实验中，我们探索了更复杂模型架构和训练方案，但并没有发现这些方法的优势能够超过它们在模型开发中引入的额外复杂性。

开发像 Llama 3 这样的旗舰基础模型不仅需要克服许多深层次的技术问题，还需要做出明智的组织决策。例如，为了确保 Llama 3 不会意外过度拟合于常用基准测试，我们的预训练数据由一个独立的团队采购和处理，该团队被强烈激励防止将外部基准测试与预训练数据污染。另一个例子是，我们确保人类评估的可信度，只允许一小部分不参与模型开发的研究人员执行和访问这些评估。虽然这些组织决策在技术论文中很少被讨论，但我们发现它们对于 Llama 3 模型家族的成功开发至关重要。

我们分享了我们的开发过程细节，因为我们相信这将有助于更广泛的研究社区了解基础模型开发的关键因素，并为公众关于基础模型未来发展进行更有见地的讨论做出贡献。我们还分享了将多模态功能整合到 Llama 3 中的初步实验结果。虽然这些模型仍在积极开发中，尚未准备好发布，但我们希望尽早分享我们的结果可以加速该方向的研究。

鉴于本文中详细的安全分析取得的积极成果，我们将公开发布我们的 Llama 3 语言模型，以加速为众多社会相关用例开发人工智能系统的进程，并使研究界能够审查我们的模型并找到使这些模型更好、更安全的途径。我们相信基础模型的公开发布对于此类模型的负责任发展至关重要，并且我们希望 Llama 3 的发布可以鼓励整个行业拥抱开放、负责任的人工智能开发。