# llama3技术报告

## 概述

**成功要素：**

1. 采用15T tokens的高质量多语言数据
2. 通过大模型提升小模型的质量，实现同类最佳效果（知识蒸馏？）
3. 采用transformer框架而非MOE架构，后训练采用DPO、SFT和RS，不用更复杂的强化学习算法。

**开发流程**

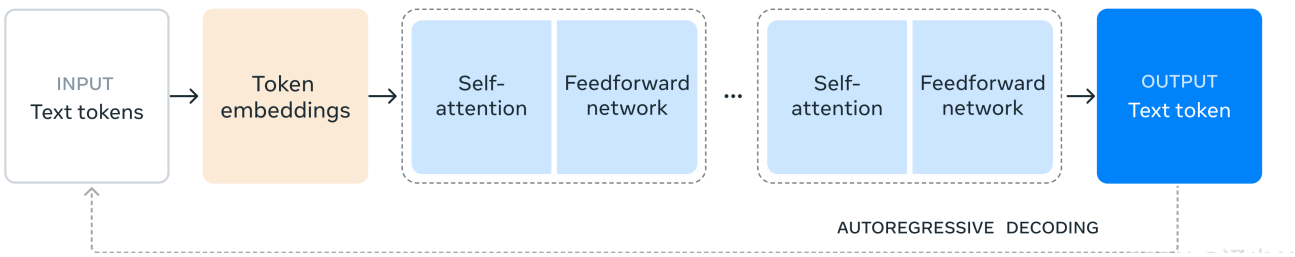
- 预训练：在15T的token上预训练405B的模型，上下文扩展到128k 个token。
- 后训练：通过多轮人类反馈与模型对齐，每轮进行SFT和DPO。后训练阶段还整合了工具使用等新能力，并在编码和推理等领域取得显著进展。

**多模态能力：**

1.**多模态编码器预训练**：分别对图像和语音编码器进行训练，学习视觉和语音信号的表示。

2.**视觉适配器训练**：将图像编码器融入预训练的语言模型，实现图像表示与语言表示的对齐。在此基础上，训练视频适配器，实现跨帧信息聚合。

3.**语音适配器训练**：将语音编码器整合到模型中，实现高质量的语音理解。




## 预训练

### 预训练数据

**网络数据整理**

- 个人隐私（PII）和安全过滤：通过过滤器从包含不安全内容或大量 PII 的网站、根据各种 Meta 安全标准被列为有害的域以及已知包含成人内容的域中删除数据。

- 自定义HTML解析器: 对非截断网页文档的原始HTML 内容进行处理，以提取高质量的多样化文本。维护数学和代码内容的结构
- 去重：URL级重复数据删除，只保留最新的页面。文档级重复数据删除，删除重复的文档。行级重复数据删除，3000 万文档作为一个桶，删除出现6次的行，可以去除导航栏，cookie警告等信息。
- 启发式过滤：使用启发式过滤(n-gram算法、Kullback-Leibler)，去除额外的低质量文档、异常值和重复过多的文档（清洗)。
- 基于模型的质量筛选：各种基于模型的质量分类器来分选高质量的token，比如说基于 fasttext 训练其识别的给定文本是否会被维基百科引用；还有基于Llama 2 的质量分类器，创建了一个经过清理的网络文档训练集，描述了质量要求，并指示 Llama 2 的聊天模型确定文档是否符合这些要求。

**数据混合比例**

- 知识分类：首先对网络数据的信息类型进行分类，并对数据类型进行下采样，保障数据类型的均衡。
- 在每个数据混合上训练几个小模型，并以此来预测一个大模型在该数据组合上的性能（根据 scaling law），在选出的候选数据混合上训练更大的模型。
- 数据混合比例：50% 通用数据, 25% 数理数据, 17% 代码数据, 8%多语言数据。

**退火数据**

在大模型训练的最后的学习率退火阶段，用高质量的代码和数学数据学习能提高性能。退火数据中不包含常用的baseline训练集，能提升在few-shot学习能力和泛化能力。在8B的模型上性能提升明显，在405B模型上没有提升，说明该模型具有强大的上下文学习和推理能力，不需要特定的领域内训练样本就能获得强大的性能。

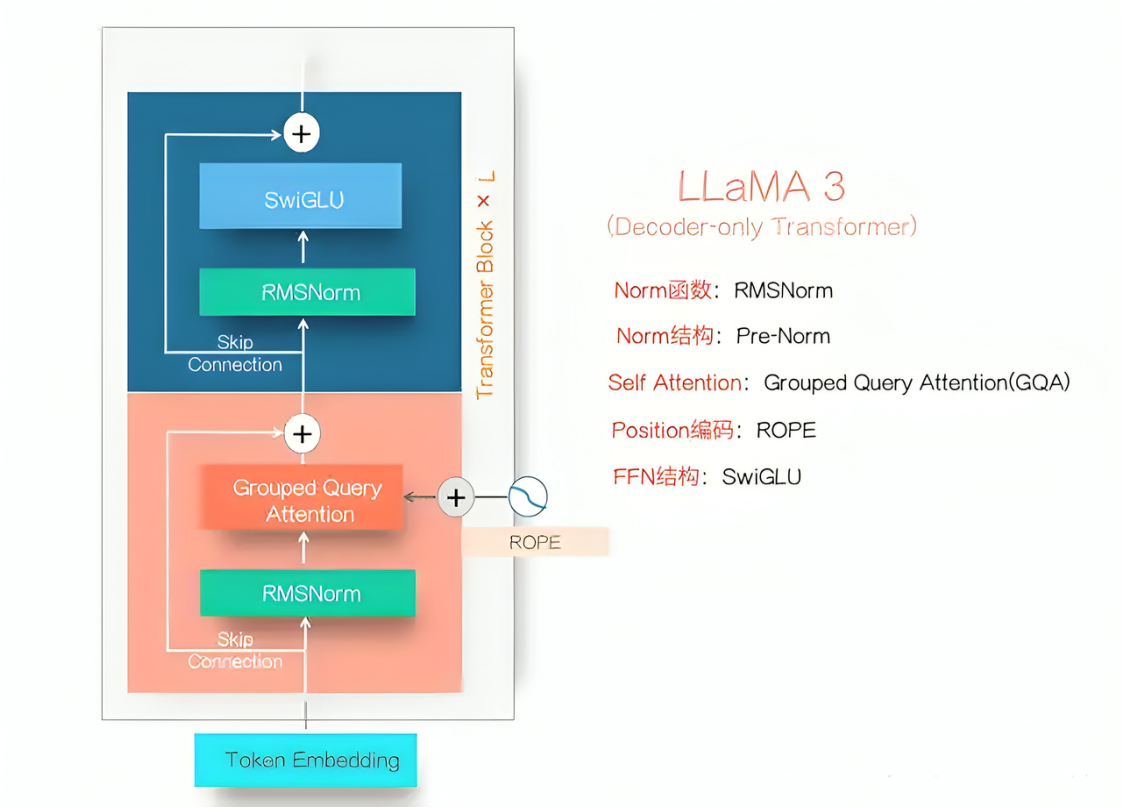
## 模型架构

性能提升主要是由于数据质量和多样性的提高以及训练规模的增加，模型层面做了以下修改：

- 采用 8 KV 头分组查询注意力（GQA）技术，提高了推理的可扩展性
- 注意力掩码可防止同一序列中不同文档之间的自注意力。
- 词表扩大到128k，100K 来自 tiktoken，28K 额外词汇用于除英语以外的语言，提高了压缩比和下游性能。
- 新tokenizer在一组英语数据上的压缩率从每个token的3.17个字符提高到3.94个字符，模型能够在相同的训练计算量下“读取”更多文本
- RoPE 超参增加至 500000

| | 8B | 70B | 405B |
|---|---|---|---|
| Layers | 32 | 80 | 126 |
| Model Dimension | 4,096 | 8192 | 16,384 |
| FFN Dimension | 14,336 | 28,672 | 53,248 |
| Attention Heads | 32 | 64 | 128 |
| Key/Value Heads | 8 | 8 | 8 |
| Peak Learning Rate | $3 \times 10^{-4}$ | $1.5 \times 10^{-4}$ | $8 \times 10^{-5}$ |
| Activation Function | SwiGLU | | |
| Vocabulary Size | 128,000 | | |
| Positional Embeddings | RoPE ($\theta = 500,000$) | | |

别的层面采用Llama系列的惯用结构，包含RMSNorm，SwiGLU，RoPE，PreNorm等：




### Scaling Law

根据scaling law确定最优的模型大小，并预测在下游baseline任务上的性能。

1. 首先建立计算最优模型在下游任务上的负对数似然（NLL）与训练FLOPs之间的相关性。FLOPs关注的是模型的计算需求，而Normalized NLL per Character关注的是模型在特定任务上的性能表现。如果一个模型需要大量的FLOPs但NLL降低不多，这可能意味着进一步增加计算资源的边际效益在减少
2. 将下游任务上的负对数似然与任务准确性相关联，利用scaling law和用更高计算FLOPs训练的旧模型。

### 并行训练

采用张量并行TP、上下文并行CP、流水线并行PP、数据并行DP。

张量并行将单个权重张量分割成不同设备上的多块。流水线并行将模型按层纵向划分为多个阶段，这样不同的设备可以并行处理整个模型流水线的不同阶段。上下文并行将输入上下文分为若干段，从而减少了超长序列长度输入的内存瓶颈。数据并行将模型、优化器和梯度分片，同时实现数据并行，在多个 GPU上并行处理数据，并在每个训练步骤后同步。在Llama 3中使用FSDP对优化器状态和梯度进行分片，但对于模型分片，在前向计算后我们不重新分片，以避免在反向传递期间产生额外的全收集通信。

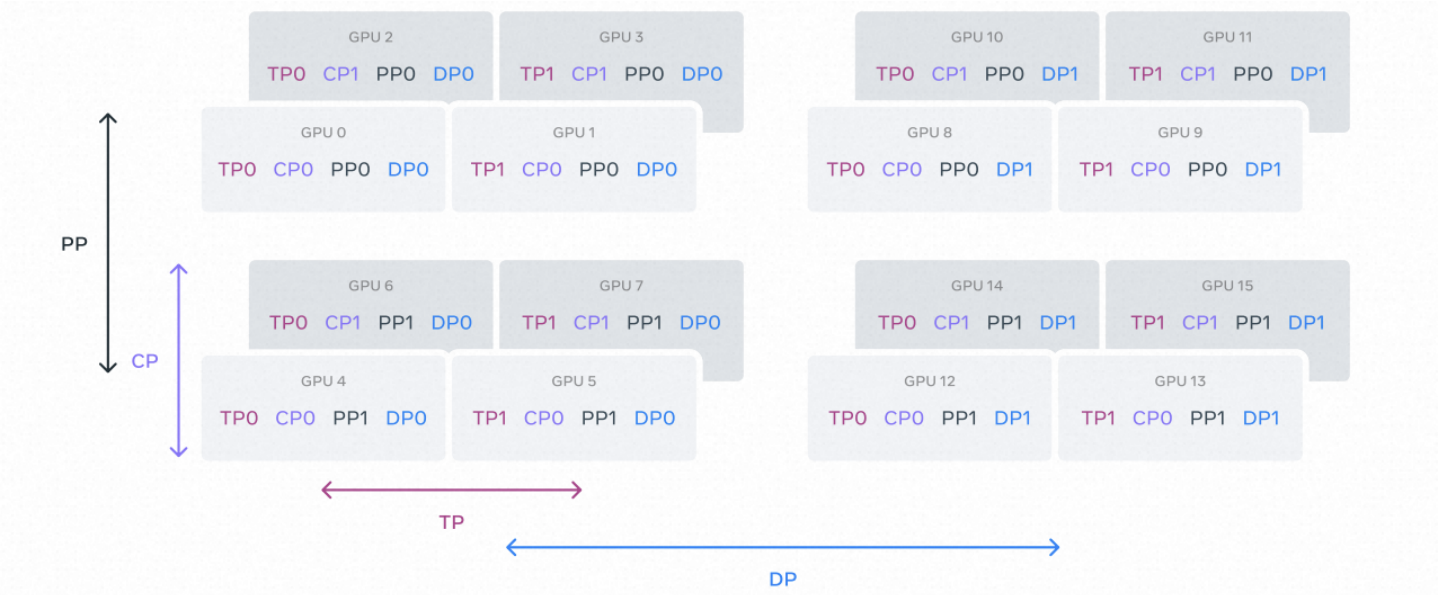



**Figure 5 Illustration of 4D parallelism.** GPUs are divided into parallelism groups in the order of [TP, CP, PP, DP], where DP stands for FSDP. In this example, 16 GPUs are configured with a group size of |TP|=2, |CP|=2, |PP|=2, and |DP|=2. A GPU's position in 4D parallelism is represented as a vector, $[D_1, D_2, D_3, D_4]$, 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.

**训练流程**

1. **初始预训练**：采用warm up + 余弦学习率，最大为8 × 10^－5，预热8000步，1200000步后衰减到8x10^-7。逐步提升训练的批量大小，以提升训练稳定性。初始batch4M，序列长度4096；训练252M和2.87T个token后加倍。

在预训练期间增加了非英语数据的百分比，以提高 Llama 3 的多语言性能；对数学数据进行上采样以提高模　型的数学推理性能；在预训练的后期添加了更多最新的网络数据以推进模型的知识截止日期（即避免对最近的知识不了解），对预训练数据的子集进行下采样，这些数据后来被识别为质量较低。

2. **长上下文预训练**：从最初的8K上下文窗口开始，逐步增加了上下文长度，经过六个阶段，最终达到128K上下文窗口。消耗800B个token。在每个长度的上下文，都要衡量是否能在短上下文任务恢复性能，以及能否完美解决该长度的最难的任务。

3. **退火**：在40M的token上，将学习率线性退火到0，保持上下文长度为128K个token。此外在退火阶段对高质量数据进行上采样。最后计算模型检查点的平均值，产生最终的预训练模型。

## 后训练

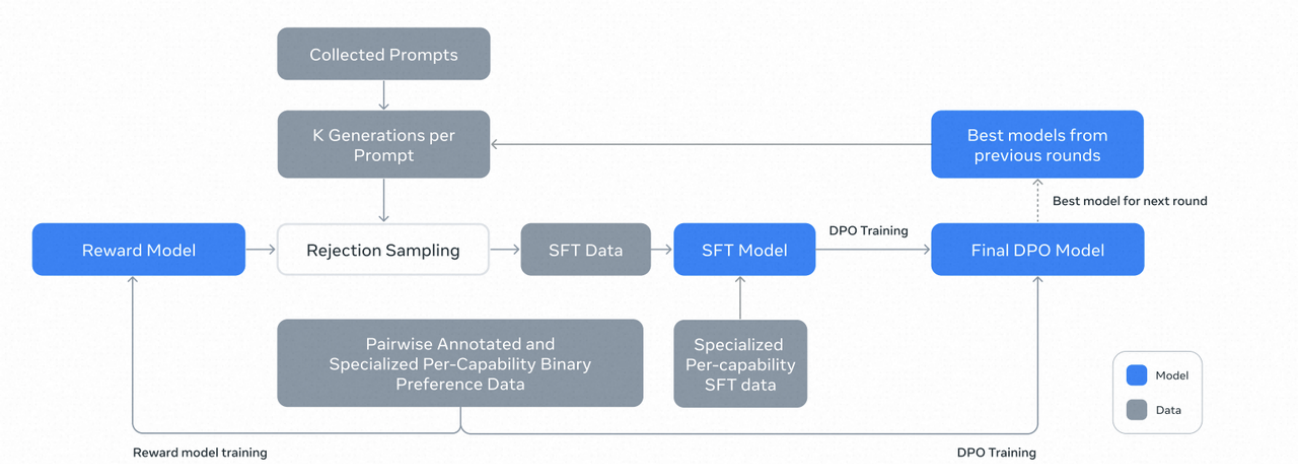



**Figure 7 Illustration of the overall post-training approach for Llama 3.** Our post-training strategy involves rejection sampling, supervised finetuning, and direct preference optimization. See text for details.

每轮后训练需要在通过人工注释或合成生成的示例上进行SFT和DPO。首先在预训练模型的基础上训练奖励模型，然后用SFT在预训练模型进行微调，并通过DPO与预训练模型对齐。

## 方法建模

1. **聊天对话格式**：为了训练LLM与人类的交互，为模型定义一个聊天对话协议，以理解人类指令并执行对话任务。此外Llama3支持工具的使用 (在一个对话回合中，可能生成多个消息并被发送到不同的位置（如用户、ipython)，定义了一个聊天对话协议，使用各种特殊的头标记和终止标记。头标记用于指示对话中每条信息的来源和目的地。终止标记用于指示何时人类和AI交替发言的时间。
2. **奖励模型**：在预训练模型的基础上训练一个奖励模型，训练目标与llama2相同，只是去掉了损失中的边际项，因为观察到data scaling后的改进效果递减。首先过滤掉偏好数据中响应相似的样本，每个偏好数据都包含2-3个回复（edited>chosen>rejected），然后将prompt和多个回复拼在一起，回复的顺序随机。
3. **SFT**：使用奖励模型对人类注释的prompt进行拒绝采样，连同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（同时屏蔽prompt上的损失）对预训练语言模型进行SFT。在405B模型 8.5K 到 9K 步的过程中以 1e-5 的学习率进行微调。
4. **DPO**：用DPO对SFT模型进行训练，与人类偏好对齐(DPO本质是二分类问题，调整模型参数鼓励模型输出Good Response，不输出Bad Response)，使用之前对齐中表现最好的模型收集的最新一批偏好数据。相比于PPO，DPO对计算量更低，且表现更好，学习率1e-5，将β超参数设置为0.1。此外去除头标记和尾标记，这些token与学习目标冲突；对所选序列应用额外的负对数似然 (NLL) 损失项，其缩放系数为 0.2，保持所需的生成格式并防止所选响应的对数概率降低。
5. **模型平均**：对从不同数据和超参数训练的RM，SFT和DPO模型进行聚合。

## 后训练数据

### 偏好数据

在每轮之后对部署多个模型进行标记，针对每个用户prompt利用两个模型生成两个回复。这些模型是基于不同数据混合和对齐方法训练的，以保障数据的多样性。偏好排名过程中，标注者以四个评级表

示其对prefer相较于rejected的评价：明显更好、更好、稍微更好或稍微更好。此外还在排名后，允许注释者编辑回答，因此部分偏好数据中，有一部分有三个响应排序（编辑 > 选择 > 拒绝）。

与llama2相比，prompt和回复的平均长度增加，说明能处理更复杂的任务。对于奖励建模和DPO，只采用明显更好和更好的偏好数据并删除掉回复相似的数据，奖励模型训练使用所有的偏好数据，DPO只用最新的batch。

基于偏好数据可以训练出奖励模型。

| Dataset | % of comparisons | Avg. # turns per dialog | Avg. # tokens per example | Avg. # tokens in prompt | Avg. # tokens in response |
|---|---|---|---|---|---|
| General English | 81.99% | 4.1 | 1,000.4 | 36.4 | 271.2 |
| Coding | 6.93% | 3.2 | 1,621.0 | 113.8 | 462.9 |
| Multilingual | 5.19% | 1.8 | 1,299.4 | 77.1 | 420.9 |
| Reasoning and tools | 5.89% | 1.6 | 707.7 | 46.6 | 129.9 |
| Total | 100% | 3.8 | 1,041.6 | 44.5 | 284.0 |

## SFT数据

拒绝采样：对每个人类标注过的prompt，采样10-30个输出，利用奖励模型选择最好的一个。在后训练的后期轮次中，我们引入系统提示，以引导拒绝采样响应符合不同能力所需的期望语调、风格或格式。 每个 SFT 示例由一个上下文(即除最后一轮外的所有对话轮次）和一个最终response组成。SFT和偏好数据有重复的领域，但是按照不同方法产生的数据。

### 数据质量控制

- 数据清洗：实施了一系列基于规则的数据移除和修改策略，例如，为了缓解过于道歉的语气问题，我们识别出过度使用的短语（如“I’ m sorry”或“I apologize”），并仔细平衡我们数据集中这类样本的比例。
- 主题分类：微调8B的模型为主题分类器，进行粗粒度分类（“数学推理”）和细粒度分类（“几何和三角学”）
- 质量评分：分别采用RM和llama信号进行质量评分，这两个评分不一致性很高，选取二者的并集。
- 难度评分：分别采用Instag和llama评分，选取最复杂的数据。
- 语意去重：对完整对话进行聚类，每个聚类内按质量分数×难度分数排序，采用贪心的思想，只保留与迄今为止聚类中看到的示例的最大余弦相似度小于阈值的示例。