deepseek

DeepSeek-V3 技術レポート

ディープシークAI

お問い合わせ

抽象的な

我々は、合計 671Bのパラメータを持ち、トークンごとに 37B がアクティブ化された強力な Mixture-of-Experts (MoE) 言語モデルである DeepSeek-V3 を紹介します。効率的な推論とコスト効率の高いトレーニングを実現するために、DeepSeek-V3は、DeepSeek-V2 で徹底的に検証されたMulti-head Latent Attention (MLA) と DeepSeekMoE アーキテクチャを採用しています。さらに、DeepSeek-V3 は、負荷分散のための補助損失のない戦略を開拓し、より強力なパフォーマンスのためにマルチトークン予測トレーニング目標を設定しています。我々は、14.8 兆個の多様で高品質のトークンで DeepSeek-V3 を事前トレーニングし、その後、教師ありファインチューニングと強化学習の段階を経て、その機能を最大限に活用します。包括的な評価により、DeepSeek-V3 は他のオープンソース モデルよりも優れており、主要なクローズドソースモデルに匹敵するパフォーマンスを達成することが明らかになりました。優れたパフォーマンスにもかかわらず、DeepSeek-V3 は完全なトレーニングに 2.788M H800 GPU 時間しか必要としません。さらに、トレーニング プロセスは驚くほど安定しています。トレーニング プロセス全体を通じて、回復不可能な損失スパイクは発生せず、ロールバックも実行されませんでした。

モデル チェックポイントはhttps://github.com/deepseek-ai/DeepSeek-V3 で入手できます。

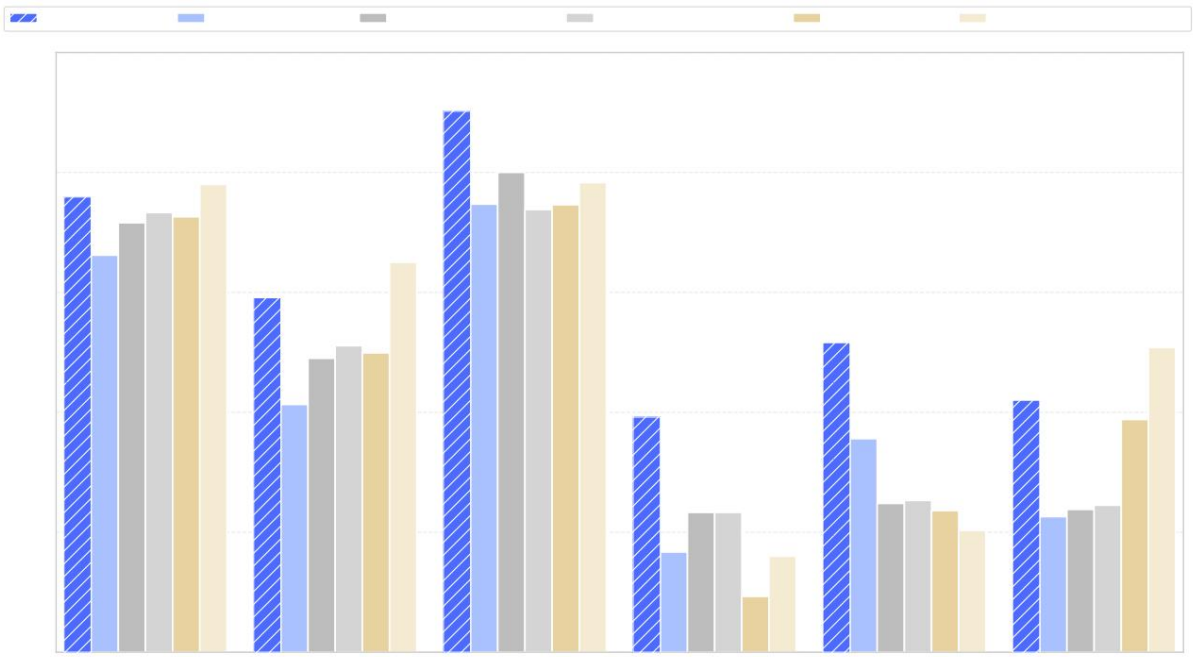


図 1 | DeepSeek-V3 とその同等製品のベンチマーク パフォーマンス。

コンテンツ

1. はじめに

近年、大規模言語モデル (LLM) は急速な反復と進化を遂げており (Anthropic、2024 年; Google、2024 年; OpenAI、2024a)、汎用人工知能 (AGI) との差は徐々に縮まっています。クローズドソース モデル以外にも、DeepSeek シリーズ (DeepSeek-AI、2024a、b、c、Guo ら、2024 年)、LLaMA シリーズ (AI@Meta、 2024a、b、Touvron ら、2023a、b)、Qwen シリーズ (Qwen、2023、2024a、b)、Mistral シリーズ (Jiangら、2023、Mistral、2024 年) などのオープンソース モデルも大きな進歩を遂げており、クローズドソース モデルとの差を縮めようと努めています。オープンソース モデルの機能の限界をさらに押し上げるために、モデルをスケールアップし、671B のパラメーターを持つ大規模な Mixture-of-Experts (MoE) モデルである DeepSeek-V3 を導入しました。このうち 37B はトークンごとにアクティブ化されます。

私たちは将来を見据えて、常に強力なモデル性能と経済的なコストを目指しています。そのため、アーキテクチャの面では、DeepSeek-V3 は効率的な推論のためにマルチヘッド潜在的注意 (MLA) (DeepSeek-AI、2024c) を、コスト効率の高いトレーニングのために DeepSeekMoE (Dai ら、2024) を採用しています。これら 2 つのアーキテクチャは DeepSeek- V2 (DeepSeek-AI、2024c) で検証されており、効率的なトレーニングと推論を実現しながら堅牢なモデル性能を維持する能力を実証しています。基本的なアーキテクチャに加えて、モデル機能をさらに強化するために 2 つの追加戦略を実装しています。まず、DeepSeek-V3 は、負荷分散を促進する取り組みから生じるモデル性能への悪影響を最小限に抑えることを目的として、負荷分散のための補助損失のない戦略 (Wang ら、2024a) を先駆的に採用しています。第二に、DeepSeek-V3 はマルチトークン予測トレーニング目標を採用しており、評価ベンチマークでの全体的なパフォーマンスが向上することが確認されています。

効率的なトレーニングを実現するために、FP8 混合精度トレーニングをサポートし、トレーニング フレームワークの包括的な最適化を実装します。低精度トレーニングは、効率的なトレーニングのための有望なソリューションとして登場しており (Dettmers ら、2022 年、Kalamkar ら、 2019 年、Narang ら、2017 年、Peng ら、2023b)、その進化はハードウェア機能の進歩と密接に関係しています (Luo ら、2024 年、Micikevicius ら、2022 年、Rouhani ら、2023a)。この研究では、FP8 混合精度トレーニング フレームワークを紹介し、非常に大規模なモデルでその有効性を初めて検証します。FP8 の計算とストレージをサポートすることで、トレーニングの高速化と GPU メモリ使用量の削減の両方を実現します。トレーニング フレームワークに関しては、パイプライン バブルが少なく、計算と通信のオーバーラップによってトレーニング中の通信の大部分を隠す、効率的なパイプライン並列処理を実現する DualPipe アルゴリズムを設計しています。このオーバーラップにより、モデルがさらにスケールアップしても、計算と通信の比率を一定に保つ限り、ノード間できめ細かなエキスパートを採用しながら、すべての通信オーバーヘッドをほぼゼロに抑えることができます。

さらに、InfiniBand (IB) と NVLink の帯域幅を最大限に活用するために、効率的なクロスノードの全対全通信カーネルも開発しています。さらに、メモリ フットプリントを細心の注意を払って最適化することで、コストのかかるテンソル並列処理を使用せずに DeepSeek-V3 をトレーニングできるようになりました。

これらの取り組みを組み合わせることで、高いトレーニング効率を実現します。

事前トレーニングでは、14.8T の高品質で多様なトークンで DeepSeek-V3 をトレーニングします。事前トレーニング プロセスは驚くほど安定しています。トレーニング プロセス全体を通じて、回復不可能な損失スパイクが発生したり、ロールバックする必要が生じたりすることはありませんでした。次に、DeepSeek-V3 の 2 段階のコンテキスト長拡張を実行します。第 1 段階では、最大コンテキスト長が 32K に拡張され、第 2 段階ではさらに 128K に拡張されます。その後、 DeepSeek-V3 のベース モデルに対して、教師ありファインチューニング (SFT) と強化学習 (RL) を含む事後トレーニングを実行し、人間の好みに合わせて、その可能性をさらに引き出します。事後トレーニング段階では、DeepSeek- R1 シリーズのモデルから推論機能を抽出し、同時にモデルの精度と学習効率のバランスを慎重に維持します。

研修費用	事前トレーニング	コンテキスト拡張	事後トレーニング	合計
H800 GPU 時間（米ドル）	2664K 532万8千ドル	119K 238万ドル	5K 0.01Mドル	2788K 557万6千ドル

表 1 | H800 のレンタル価格が GPU 時間あたり 2 ドルであると仮定した場合の DeepSeek-V3 のトレーニング コスト。

世代の長さ。

DeepSeek-V3を包括的なベンチマークで評価しました。
トレーニングコストの包括的な評価により、DeepSeek-V3-Baseが
特にコードと数学において、現在入手可能な最も強力なオープンソースベースモデルです。そのチャット
バージョンは他のオープンソースモデルよりも優れており、
GPT-4oやClaude-3.5-Sonnetなどの主要なクローズドソースモデルを一連の標準モデルに適用した。
およびオープンエンドのベンチマーク。

最後に、DeepSeek-V3の経済的なトレーニングコストを再度強調します。
表 1 は、アルゴリズム、フレームワーク、ハードウェアの最適化された共同設計によって実現されました。
事前トレーニング段階では、1兆トークンごとにDeepSeek-V3をトレーニングするのに必要なのはわずか180K
H800 GPU時間、つまり2048個のH800 GPUを搭載したクラスタでは3.7日です。その結果、事前トレーニング段
階は2か月未満で完了し、2664K GPU時間が費やされます。
コンテキスト長の拡張に119K GPU時間、トレーニング後に5K GPU時間、
DeepSeek-V3のフルトレーニングにかかるコストはわずか278万8千GPU時間です。
H800 GPUは1GPU時間あたり2ドルなので、トレーニングの総コストはわずか557万6000ドルです。
上記の費用には、DeepSeek-V3の公式トレーニングのみが含まれており、
アーキテクチャ、アルゴリズム、またはデータに関する以前の研究およびアブレーション実験に関連付けられています。

私たちの主な貢献は次のとおりです。

アーキテクチャ: 革新的な負荷分散戦略とトレーニング目標

- DeepSeek-V2の効率的なアーキテクチャをベースに、補助ロスのない
  負荷分散戦略により、発生するパフォーマンスの低下を最小限に抑えます
  負荷分散を促進することから。
- マルチトークン予測（MTP)の目的を調査し、モデル化に有益であることを証明しました。
  パフォーマンスが向上します。推論の高速化のための投機的デコードにも使用できます。

事前トレーニング :究極のトレーニング効率を目指して

- FP8混合精度トレーニングフレームワークを設計し、初めて検証しました。
  非常に大規模なモデルにおける FP8 トレーニングの実現可能性と有効性。
- アルゴリズム、フレームワーク、ハードウェアの共同設計を通じて、
  ノード間のMoEトレーニングにおける通信ボトルネックを解消し、ほぼ完全な計算と通信
  の重複を実現しました。これにより、トレーニングの効率が大幅に向上し、
  トレーニングコストを削減し、追加のオーバーヘッドなしでモデルサイズをさらに拡大できます。
- わずか266万H800 GPU時間という経済的なコストで、
  14.8T トークンの DeepSeek-V3 は、現在最強のオープンソース ベース モデルを生み出します。
  事前トレーニング後の後続のトレーニング ステージでは、わずか 0.1M GPU 時間しか必要ありません。

トレーニング後: DeepSeek-R1 からの知識抽出

- 我々は、DeepSeek R1シリーズモデルの1つである長い思考連鎖（CoT)モデルから推論能力を抽出する革新的な方法論を
  導入します。
  標準的なLLM、特にDeepSeek-V3に統合されています。当社のパイプラインは、

R1 の検証および反映パターンを DeepSeek-V3 に組み込み、推論パフォーマンスを大幅に向上させました。同時に、 DeepSeek-V3 の出力スタイルと長さも制御しています。

コア評価結果の概要

- 知識: (1) MMLU、MMLU-Pro、GPQAなどの教育ベンチマークでは、 DeepSeek-V3は他のすべてのオープンソースモデルを上回り、MMLUで88.5、 MMLU-Proで75.9、GPQAで59.1を達成しました。そのパフォーマンスは、 GPT-4oやClaude-Sonnet-3.5などの主要なクローズドソースモデルに匹敵し、この領域におけるオープンソースモデルとクローズドソースモデルのギャップを縮めています。 (2) 事実性ベンチマークでは、DeepSeek-V3は、 SimpleQAと中国語SimpleQAの両方でオープンソースモデルの中で優れたパフォーマンスを示しています。英語の事実知識(SimpleQA)ではGPT-4oとClaude-Sonnet-3.5に遅れをとっていますが、中国語の事実知識(Chinese SimpleQA)ではこれらのモデルを上回っており、中国語の事実知識における強さを際立たせています。 •コード、数学、推論： (1)DeepSeek-V3は、すべての非long-CoTオープンソースおよびクローズドソースモデルの中で、数学関連のベンチマークで最先端のパフォーマンスを達成しています。

特に、MATH-500などの特定のベンチマークではo1-previewよりも優れたパフォーマンスを発揮し、堅牢な数学的推論能力を実証しています。(2) コーディング関連のタスクでは、 DeepSeek-V3はLiveCodeBenchなどのコーディング競技ベンチマークで最高のパフォーマンスを発揮し、この分野における主要モデルとしての地位を固めています。エンジニアリング関連のタスクでは、DeepSeek-V3のパフォーマンスはClaude-Sonnet-3.5をわずかに下回りますが、それでも他のすべてのモデルを大幅に上回っており、さまざまな技術ベンチマークで競争力を発揮しています。

この論文の残りの部分では、まず DeepSeek-V3 モデル アーキテクチャの詳細を説明します(セクション 2)。次に、コンピューティング クラスター、トレーニング フレームワーク、FP8 トレーニングのサポート、推論展開戦略、将来のハードウェア設計に関する提案など、インフラストラクチャを紹介します。次に、トレーニング データの構築、ハイパーパラメータ設定、ロング コンテキスト拡張手法、関連する評価、およびいくつかの議論を含む事前トレーニング プロセスについて説明します (セクション 4)。

その後、教師ありファインチューニング (SFT)、強化学習 (RL)、対応する評価、および議論を含む、トレーニング後の取り組みについて説明します (セクション 5)。最後に、この研究を締めくくり、DeepSeek-V3 の既存の制限について説明し、将来の研究の潜在的な方向性を提案します (セクション 6)。

2. 建築

まず、効率的な推論のためのマルチヘッド潜在的注意(MLA) (DeepSeek-AI、2024c) と経済的なトレーニングのための DeepSeekMoE (Dai et al.、2024)を特徴とする DeepSeek-V3 の基本アーキテクチャを紹介します。次に、評価ベンチマークで全体的なパフォーマンスを向上させることが確認されているマルチトークン予測 (MTP) トレーニング目標を紹介します。明示的に言及されていないその他の細かい点については、DeepSeek-V3 は DeepSeek-V2 (DeepSeek-AI、2024c) の設定に準拠しています。

2.1. 基本アーキテクチャ

DeepSeek-V3の基本アーキテクチャは、依然としてTransformer（Vaswani et al., 2017）フレームワーク内にあります。効率的な推論と経済的なトレーニングのために、DeepSeek-V3は、 DeepSeek-V2で徹底的に検証されたMLAとDeepSeekMoEも採用しています。DeepSeek -V2と比較した例外は、補助損失のない負荷分散を追加導入していることです。

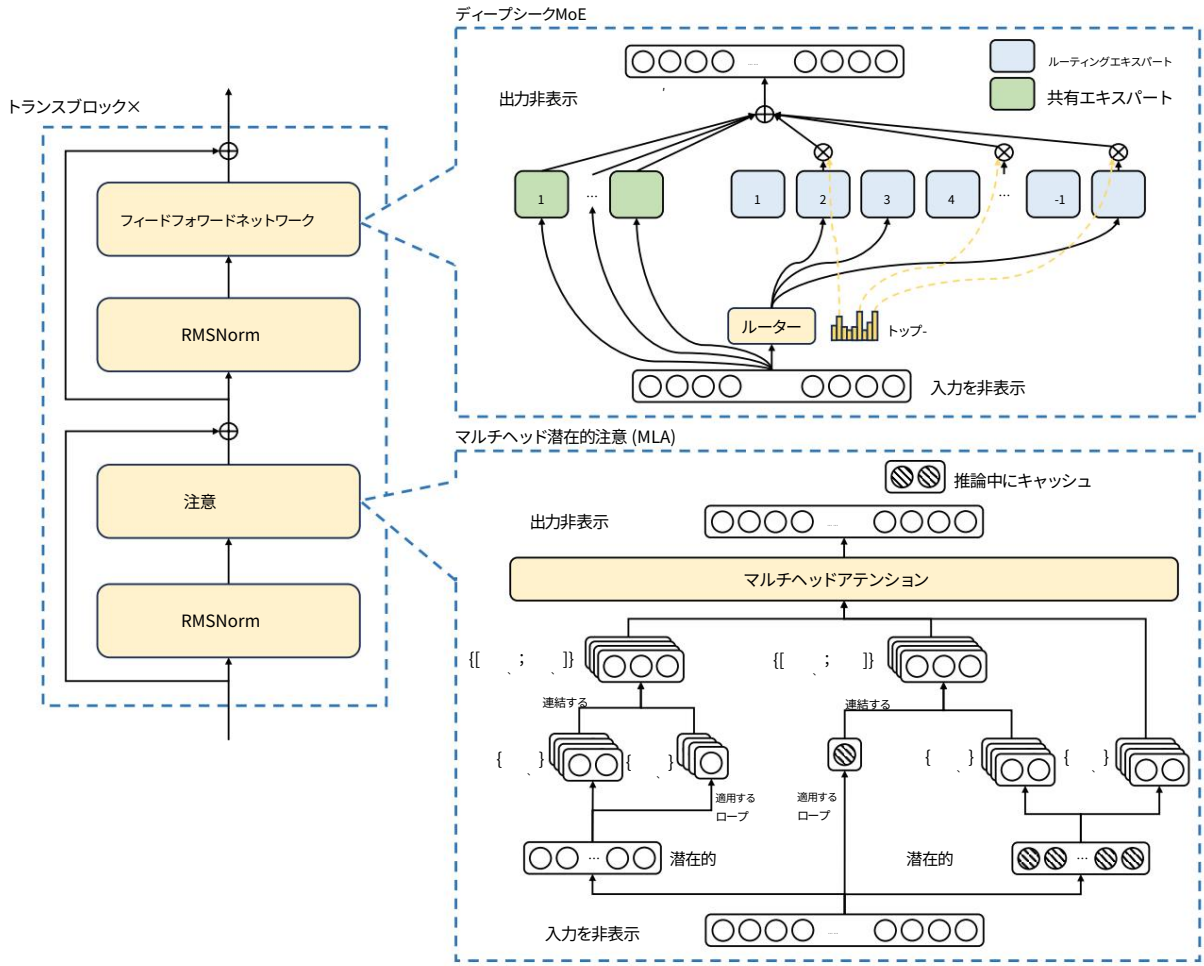


図2 | DeepSeek-V3の基本アーキテクチャの図解。DeepSeek-V2に続いて、
効率的な推論と経済的なトレーニングのために MLA と DeepSeekMoE を採用します。

DeepSeekMoEの戦略（Wang et al., 2024a)は、
負荷バランスを確保するための努力によって。図2はDeepSeek-V3の基本アーキテクチャを示しています。
このセクションでは、MLA と DeepSeekMoE の詳細について簡単に説明します。

2.1.1. マルチヘッド潜在的注意

DeepSeek-V3は、注目度に関してMLAアーキテクチャを採用しています。注目度を埋め込み次元、注目度ヘッドの数、ヘッドあたりの
次元、 h ∈ R sion、注目度レイヤーの-番目のトークンに対する注目度入力を表します。MLAの核となるのは、

注目キーと値の低ランクジョイント圧縮により、キー値（KV)キャッシュを削減
推論：

$$c = \quad h \text{、} \tag{1}$$

$$[k,1; k,2; \ldots; k \text{、}] = \text{く}= \quad c \text{、} \tag{2}$$

$$k = \mathrm{RoPE}(h)\text{、} \tag{3}$$

$$\text{け}\text{、} = [k \text{、}; k], \tag{4}$$

$$[v,1; v,2; \ldots; v \text{、}] = \text{ヴ}= \quad c \text{、} \tag{5}$$

ここで、 c ∈ Rはキーと値の圧縮潜在ベクトルであり、 (≪)は KV を示す。
圧縮次元; はダウン投影行列を表します。はそれぞれキーと値のアップ投影行列です。 は使用される行列です。 、 ∈ R ×
∈R ×
∈ R ×

回転位置埋め込み（RoPE)を運ぶ分離キーを生成する（Su et al., 2024）
RoPE(·)はRoPE行列を適用する演算を表し、 [·; ·]は連結を表します。注
MLAの場合、生成中に青いボックスで囲まれたベクトル（つまり、 cとk)のみがキャッシュされる必要がある。
これにより、KVキャッシュが大幅に削減され、同等のパフォーマンスが維持されます。
標準的なマルチヘッドアテンション（MHA）（Vaswani et al.、2017）。

注目クエリについては、低ランク圧縮も実行し、
トレーニング中の活性化メモリ:

$$c = \quad h \text{、} \tag{6}$$

$$[q,1; q,2; ...; q \text{、}] = q = \quad c \text{、} \tag{7}$$

$$[q,1; q,2; ...; q \text{、}] = q = \mathrm{RoPE}(c),], \tag{8}$$

$$q \text{、} = [q \text{、}; ク \text{、} \tag{9}$$

ここでc∈R ′ はクエリの圧縮潜在ベクトルである。 ′ (≪) はクエリを表す。
圧縮次元; はダウン投影とアップ投影である∈R ′× 、 ∈ R ×′
∈R ′
それぞれクエリの行列であり、分離された行列を生成する行列である。
RoPE を伝送するクエリ。

最終的に、注目クエリ（q最終注目出力u ： 、）、キー（k 、）、および値（v 、)を組み合わせて

$$o \text{、} = \sum_{=1} \text{ソフトマックス}\left(\frac{q \text{、} け \text{、}}{\sqrt{\quad +}}\right) v \text{、、} \tag{10}$$

$$u = [o,1; o,2; ...; o \quad \text{、}], \tag{11}$$

ここで、∈ R × は出力投影行列を表します。

2.1.2. 補助ロスフリー負荷分散を備えた DeepSeekMoE

DeepSeekMoEの基本アーキテクチャ。フィードフォワードネットワーク（FFN)の場合、DeepSeek-V3
DeepSeekMoEアーキテクチャを採用しています（Dai et al., 2024）。従来のMoEと比較して
GShard（Lepikhin et al., 2021)のようなアーキテクチャとは異なり、DeepSeekMoEはより細分化された専門家と
は、いくつかのエキスパートを共有エキスパートとして分離します。uを-番目のトークンのFFN入力とすると、次のように計算します。
FFN出力h ′ 次のように：

$$h' = u + \sum_{=1} \mathrm{FFN}(\ )\ (u) + \sum_{=1} \text{、} \mathrm{FFN}(\ )\ (u) \text{、} \tag{12}$$

$$\text{、} = \frac{\text{、}}{\sum_{=1} \text{、}} \text{、} \tag{13}$$

$$\text{、} = \text{、、} \in \mathrm{Topk}(\{, |1 \quad \},), \tag{14}$$

0、それ以外の場合は

$$\text{、} = \text{シグモイド} ue \quad \text{、} \tag{15}$$

ここで、およびはそれぞれ共有エキスパートとルーティングエキスパートの数を表します。FFN ( ) （·）
およびFFN( ) (·) はそれぞれ - 番目の共有エキスパートと - 番目のルーティングエキスパートを表します。
アクティブ化されたルーティングされたエキスパートの数。 は - 番目のエキスパートのゲート値です。
トークンとエキスパートの親和性、 eはルーティングされたエキスパートの重心ベクトル、 Topk(·,)は
-番目のトークンに対して計算された親和性スコアのうち最も高いスコアを含むセットと
すべてのルーティングされたエキスパート。DeepSeek-V2とは少し異なり、DeepSeek-V3はシグモイド関数を使用する。
親和性スコアを計算し、選択されたすべての親和性スコア間の正規化を適用して
ゲート値を生成します。

補助損失のない負荷分散。MoEモデルでは、エキスパート負荷の不均衡により、
ルーティングの崩壊（Shazeer et al., 2017）と、
専門家の並列処理。従来のソリューションは通常、補助損失に依存しています（Fedus et al.、2021;
不均衡な負荷を回避するために、補助損失が大きすぎると、
モデルのパフォーマンスを向上させる（Wang et al., 2024a）。負荷バランスと負荷分散のトレードオフを改善するために
モデルのパフォーマンスを向上させるために、補助損失のない負荷分散戦略を開発しました（Wang et al.、
2024a) は負荷バランスを確保するために導入されました。具体的には、各専門家にバイアス項を導入し、
対応する親和性スコアに追加する 、トップKルーティングを決定するには:

$$' = \quad 、、 \quad 、 + \in \mathrm{Topk}(\{ , + |1 \quad \}, 0、それ以外の場 \quad)、 \quad 、 \quad 合。 \tag{16}$$

バイアス項はルーティングにのみ使用されることに注意してください。ゲート値は、
FFN出力は、依然として元の親和性スコアから導出されます。トレーニング中は、、
各トレーニングステップのバッチ全体のエキスパート負荷を監視します。各ステップの最後に、
対応する専門家が過負荷状態の場合はバイアス項を減少させ、増加させる。
対応するエキスパートが過負荷の場合、バイアス更新と呼ばれるハイパーパラメータは
速度。ダイナミックな調整により、DeepSeek-V3は、
トレーニングを通じて負荷分散を促すモデルよりも優れたパフォーマンスを実現します。
純粋な補助損失。

補完的シーケンスワイズ補助損失。DeepSeek -V3は主に
補助損失のない負荷バランス戦略により、単一の負荷内での極端な不均衡を防止します。
シーケンスでは、補完的なシーケンスごとのバランス損失も使用します。

$$\mathrm{LBal} = \Sigma \quad 、 \quad =1 \tag{17}$$

$$= \Sigma \quad \frac{1}{} \quad =1 \quad 、 \in \mathrm{Topk}(\{ , |1 \quad \}, \quad) 、 \tag{18}$$

$$' = \frac{、}{} \quad 、 \quad =1、 \tag{19}$$

$$= \frac{1}{} \Sigma \quad ' \quad =1 \quad 、、 \tag{20}$$

ここでバランス係数はハイパーパラメータであり、非常に小さな値が割り当てられる。
DeepSeek-V3の値。1 (·)はインジケータ関数を示し、トークンの数を示す。
シーケンスごとのバランス損失は、各シーケンスの専門家の負荷を
バランスが取れている。

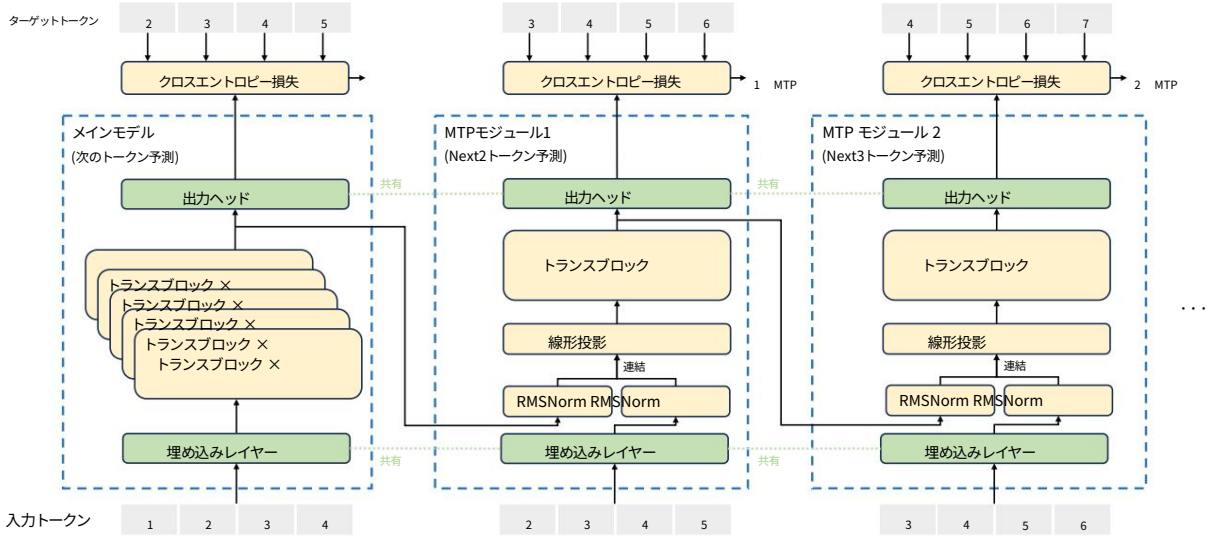


図3 |マルチトークン予測（MTP)の実装の図解。
各深度における各トークンの予測の完全な因果連鎖。

ノード制限ルーティング。DeepSeek -V2で使用されるデバイス制限ルーティングと同様に、DeepSeek-V3
また、トレーニング中の通信コストを制限するために制限されたルーティングメカニズムも使用します。つまり、
各トークンは最大でノードに送信され、そのノードは以下に従って選択される。
各ノードに分散しているエキスパートの最も高い親和性スコアの合計。この
制約条件の下で、私たちのMoEトレーニングフレームワークは、ほぼ完全な計算通信を実現できます。
重複します。

トークンドロップなし。効果的な負荷分散戦略により、DeepSeek-V3は良好な状態を保ちます。
トレーニング全体を通して負荷分散を行います。そのため、DeepSeek-V3はトレーニング中にトークンをドロップしません。
トレーニング。さらに、推論負荷を確実にするための特定の展開戦略も実装しています。
バランスをとるため、DeepSeek-V3 も推論中にトークンをドロップしません。

2.2. マルチトークン予測

Gloeckle et al. (2024)に触発されて、我々はマルチトークン予測（MTP)を調査し設定しました。
DeepSeek-V3の目標は、予測範囲を各時点の複数の将来のトークンに拡張することです。
一方で、MTP目標はトレーニング信号を高密度化し、
データ効率。一方で、MTPはモデルが表現を事前に計画することを可能にする。
将来のトークンをより正確に予測するために、図3にMTPの実装を示します。
Gloeckle et al. (2024) のモデルは、独立した
出力ヘッドでは、追加のトークンを順番に予測し、完全な因果連鎖を維持します。
各予測深度。このセクションでは、MTP 実装の詳細を紹介します。

MTPモジュール。具体的には、MTP実装では、追加のトークンを予測するためにシーケンシャルモジュールを使用します。 -番目のMTPモジュールは、共有埋め込み層Emb(·) ...
×2
出力ヘッドOutHead(·)、 TransformerブロックTRM(·)、および射影行列∈R 。
-番目の予測深度における-番目の入力トークンの表現を、まず-番目の
(− 1)番目の深さhのトークン $^{-1}$ ∈ Rと(+) 番目のトークンの埋め込み (+) ∈ R

線形投影の場合:

$$h' = [\mathrm{RMSNorm}(h^{-1}); \mathrm{RMSNorm}(\mathrm{Emb}(+))], \tag{21}$$

ここで、 [·; ·]は連結を表します。特に、 = 1の場合、 hはメインモデルによって与えられた表現を参照します。各MTPモジュールについて、その埋め込み層はメインモデルと共有されることに注意してください。結合されたh 'は、 -番目の深さのTransformerブロックの入力として機能し、現在の深さhでの出力表現を生成します。

$$^{時間1:-} = \mathrm{TRM}(h'_{1:-})、 \tag{22}$$

ここで、は入力シーケンスの長さを表し、はスライス操作（左境界と右境界の両方を含む)を表します。最後に、 hを入力として、共有出力ヘッドは、語彙サイズである番目の追加予測トークンの確率分布を計算します。

$$_{+1+} \in R 、$$

$$_{++1} = \mathrm{OutHead}(h) です。 \tag{23}$$

出力ヘッドOutHead(·)は表現をロジットに線形マッピングし、その後Softmax(·)関数を適用して、 -番目の追加トークンの予測確率を計算します。
また、各 MTP モジュールの出力ヘッドはメインモデルと共有されます。予測の因果関係の連鎖を維持するという私たちの原則は EAGLE (Li et al., 2024b) の原則と似ていますが、その主な目的は投機的デコード (Leviathan et al., 2023; Xia et al., 2023) であるのに対し、私たちはMTP を利用してトレーニングを改善しています。

MTP トレーニング目標。各予測深度について、クロスエントロピー損失 L MTP を計算します。

$$ら_{MTP} = クロスエントロピー(2+:+1, 2+:+1) = -\frac{1}{}\sum_{=2+}^{+1} \log[\ ], \tag{24}$$

ここで、は入力シーケンスの長さ、は - 番目の位置のグラウンドトゥルーストークン、および[ ] は- 番目の MTP モジュールによって与えられる の対応する予測確率を表します。
最後に、すべての深度にわたる MTP 損失の平均を計算し、重み付け係数を掛けて全体的な MTP 損失LMTP を取得します。これは、DeepSeek-V3 の追加のトレーニング目標として機能します。

$$\mathrm{LMTP} = \Sigma^{-}_{=1} L_{\mathrm{MTP}}。 \tag{25}$$

推論における MTP。当社の MTP 戦略は、主にメイン モデルのパフォーマンスを向上させることを目的としているため、推論中に MTP モジュールを直接破棄し、メイン モデルを独立して正常に機能させることができます。さらに、これらの MTP モジュールを投機的デコードに再利用して、生成のレイテンシをさらに改善することもできます。

3. インフラ

3.1. コンピューティング クラスター

DeepSeek-V3 は、2048 個の NVIDIA H800 GPU を搭載したクラスターでトレーニングされます。H800クラスターの各ノードには、ノード内の NVLink と NVSwitch によって接続された 8 個の GPU が含まれています。異なるノード間では、InfiniBand (IB) 相互接続が通信を容易にするために利用されます。

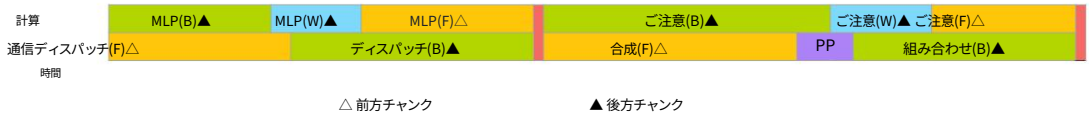


図4 |前方および後方のチャンクのペアに対するオーバーラップ戦略（
トランスブロックの境界は揃っていません）。オレンジは前進、緑は
「入力の逆」、青は「重みの逆」、紫はPP通信を表します。
赤は障壁を表します。全員対全員通信と PP 通信は両方とも完全に非表示にすることができます。

3.2. トレーニングフレームワーク

DeepSeek-V3のトレーニングは、効率的で
当社のエンジニアがゼロから作り上げた軽量トレーニングフレームワークです。
DeepSeek-V3 は、16 ウェイのパイプライン並列処理 (PP) (Qi et al.、2023a)、8 ノードにわたる 64 ウェイの Expert Paral-Lelism (EP) (Lepikhin et al.、2021)、および ZeRO-1 データ並列処理 (DP) (Rajbhandari et al.、2020) を適用します。

DeepSeek-V3の効率的なトレーニングを促進するために、細心の注意を払ったエンジニアリングを実施しています。
最適化。まず、効率的なパイプライン並列処理を実現する DualPipe アルゴリズムを設計します。
既存のPP法と比較して、DualPipeはパイプラインバブルが少なくなります。さらに重要なのは、
計算と通信の段階を前方プロセスと後方プロセスに重ね合わせ、
これにより、クロスノードによってもたらされる大きな通信オーバーヘッドの課題に対処します。
エキスパート並列処理。次に、効率的なクロスノード全対全通信カーネルを開発する。
IBとNVLinkの帯域幅を最大限に活用し、ストリーミングマルチプロセッサ（SM)を節約する
通信専用です。最後に、メモリ使用量を慎重に最適化します。
トレーニングにより、コストのかかる Tensor Parallelism (TP) を使用せずに DeepSeek-V3 をトレーニングできるようになります。

3.2.1. DualPipeと計算と通信の重複

DeepSeek-V3では、クロスノードエキスパート並列処理によって発生する通信オーバーヘッドは
その結果、計算と通信の比率が約1:1という非効率的なものになります。これに対処するために
この課題を解決するために、私たちはDualPipeと呼ばれる革新的なパイプライン並列アルゴリズムを設計しました。
前方および後方の計算通信フェーズを効果的にオーバーラップさせることでモデルのトレーニングを加速するだけでなく、パイプライン バブルも削減します。

DualPipeの重要なアイデアは、計算と通信を2つのパイプ内でオーバーラップさせることです。
個々の前方および後方チャンク。具体的には、各チャンクをアテンション、全体から全体へのディスパッチ、 MLP、および全体から全体への結合の4つのコンポーネントに分割します。特に、
後方チャンクでは、注意とMLPの両方がさらに2つの部分に分割され、後方では
ゼロバブル（Qi et al., 2023b)のように、重みの入力と逆方向の計算を行う。さらに、
PP通信コンポーネントがあります。図4に示すように、前方および後方のペアに対して、
後方チャンクでは、これらのコンポーネントを再配置し、GPU SMの比率を手動で調整します。
通信と計算に特化しています。この重複戦略により、
実行中に全員対全員通信とPP通信の両方を完全に隠蔽することができる。
効率的なオーバーラップ戦略として、完全なデュアルパイプスケジューリングを図5に示します。
双方向パイプラインスケジューリング。パイプラインの両端からマイクロバッチを供給する。
同時に、通信の大部分が完全に重複する可能性があります。
オーバーラップは、モデルがさらに拡大しても、一定値を維持する限り、
計算と通信の比率を下げても、ノード間できめ細かな専門家を雇用することができる。
ほぼゼロの全対全通信オーバーヘッドを実現します。

図 5 |双方向の 8 つの PP ランクと 20 個のマイクロバッチの DualPipe スケジューリングの例。
逆方向のマイクロバッチは順方向のものと対称なので、
図を簡略化するため、バッチIDは省略しています。共通の黒い枠線で囲まれた2つのセル
計算と通信が相互に重なり合っています。

方法	バブル	パラメータ	アクティベーション
1F1B	(− 1) (+) (− 1)	1×	
ZB1P	(+ − 2) (	1×	
DualPipe（弊社）	$\frac{}{2}$ − 1) (& + − 3)	2×	+ 1

表2 |異なるパイプライン並列処理におけるパイプラインバブルとメモリ使用量の比較
メソッド。フォワードチャンクの実行時間を示し、
完全な後方チャンクは、「重みのための後方」チャンクの実行時間を示し、& は相互に重なり合った 2 つの前方チャンクと後方チャンクの実行時間を示します。

さらに、通信負荷が重くない一般的なシナリオでも、デュアルパイプは依然として効率性の利点を発揮します。表2では、パイプラインバブルと
異なるPP方式でのメモリ使用量。表に示すように、ZB1P（Qi
1F1B（Harlap et al., 2018)と1F1B（Harlap et al., 2018)では、DualPipeはパイプラインの気泡を大幅に削減します。
ピーク活性化メモリを倍増させるだけです。DualPipeには $\frac{1}{}$
モデルパラメータのコピーを2つ保持しても、メモリが大幅に増加することはない。
トレーニング中に大きなEPサイズを使用するため、消費量が増加します。キメラ（Liと
Hoefler、2021）、DualPipeでは、パイプラインステージとマイクロバッチが
2、マイクロバッチをパイプラインステージで分割する必要がありません。さらに、DualPipeでは、
マイクロバッチの数が増えても、バブルもアクティベーションメモリも増加しません。

3.2.2. ノード間の全対全通信の効率的な実装

DualPipeの十分な計算性能を確保するために、効率的な
ノード間の全対全通信カーネル（ディスパッチと結合を含む)を節約する
通信専用のSMの数。カーネルの実装は、 MoEゲーティングアルゴリズムとクラスタのネットワークトポロジーと連携して設計されています。具体的には、
私たちのクラスターでは、クロスノードGPUはIBとノード内通信で完全に相互接続されています。
NVLinkを介して処理されます。NVLinkは160 GB/秒の帯域幅を提供し、IBの約3.2倍です。
（50 GB/秒）。IBとNVLinkの異なる帯域幅を効果的に活用するために、それぞれを制限しています。
トークンは最大4つのノードにディスパッチされ、IBトラフィックが削減されます。各トークンについて、
ルーティングの決定が行われると、まずIB経由で同じインノードのGPUに送信されます。
ターゲットノードにインデックスを張る。ターゲットノードに到達したら、
NVLink経由で、ターゲットエキスパートをホストする特定のGPUに瞬時に転送され、
その後到着するトークンによってブロックされる。このようにして、IBとNVLinkを介した通信は
完全に重複しており、各トークンはノードあたり平均3.2人の専門家を効率的に選択できる。
NVLinkによる追加オーバーヘッドを被ることなく、DeepSeek-V3

実際には 8 つのルーティングされたエキスパートのみを選択しますが、同じ通信コストを維持しながら、この数を最大 13 のエキスパート(4 ノード × 3.2 エキスパート/ノード)まで拡張できます。全体として、このような通信戦略では、IB と NVLink の帯域幅を完全に活用するには 20 の SM だけで十分です。

具体的には、ワープ特化技術（Bauer et al., 2014)を採用し、20個のSMを10個の通信チャネルに分割する。ディスパッチ処理中、（1)IB送信、（2）IBからNVLinkへの転送、および(3)NVLinkの受信は、それぞれのワープによって処理されます。各通信タスクに割り当てられるワープの数は、すべてのSMの実際のワークロードに応じて動的に調整されます。同様に、結合プロセスでは、(1)NVLink送信、 (2)NVLinkからIBへの転送と蓄積、および(3)IBの受信と蓄積も動的に調整されたワープによって処理されます。さらに、ディスパッチカーネルと結合カーネルはどちらも計算ストリームと重複するため、他のSM計算カーネルへの影響も考慮します。具体的には、カスタマイズされたPTX（並列スレッド実行)命令を採用し、通信チャンクサイズを自動調整することで、L2キャッシュの使用と他のSMへの干渉を大幅に削減します。

3.2.3. 最小限のオーバーヘッドでメモリを大幅に節約

トレーニング中のメモリ使用量を削減するために、次の手法を採用しています。

RMSNorm と MLA アッププロジェクションの再計算。バックプロパゲーション中にすべての RMSNorm 操作と MLA アッププロジェクションを再計算することで、出力アクティベーションを永続的に保存する必要がなくなります。わずかなオーバーヘッドで、この戦略によりアクティベーションを保存するためのメモリ要件が大幅に削減されます。

CPU での指数移動平均。トレーニング中、学習率の低下後のモデル パフォーマンスを早期に推定するために、モデル パラメータの指数移動平均 (EMA) を保存します。EMA パラメータは CPU メモリに保存され、各トレーニング ステップの後に非同期的に更新されます。この方法により、追加のメモリや時間のオーバーヘッドを発生させることなく、EMA パラメータを維持できます。

マルチトークン予測のための共有埋め込みと出力ヘッド。DualPipe戦略では、モデルの最も浅い層 (埋め込み層を含む) と最も深い層 (出力ヘッドを含む) を同じ PP ランクに展開します。この配置により、共有埋め込みと出力ヘッドのパラメーターと勾配をMTP モジュールとメイン モデル間で物理的に共有できます。この物理的な共有メカニズムにより、メモリ効率がさらに向上します。

3.3. FP8トレーニング

低精度トレーニングの最近の進歩（Dettmers et al., 2022; Noune et al., 2022; Peng et al., 2023b)に触発されて、FP8データ形式を利用してDeepSeek-V3をトレーニングする細粒度の混合精度フレームワークを提案します。低精度トレーニングは大きな可能性を秘めていますが、アクティベーション、重み、勾配に外れ値が存在することで制限されることがよくあります（Fishman et al., 2024; He et al.; Sun et al., 2024）。推論量子化では大きな進歩が遂げられていますが（Frantar et al., 2022; Xiao et al., 2023）、大規模言語モデルで低精度技術をうまく適用したことを示す研究は比較的少ないです。

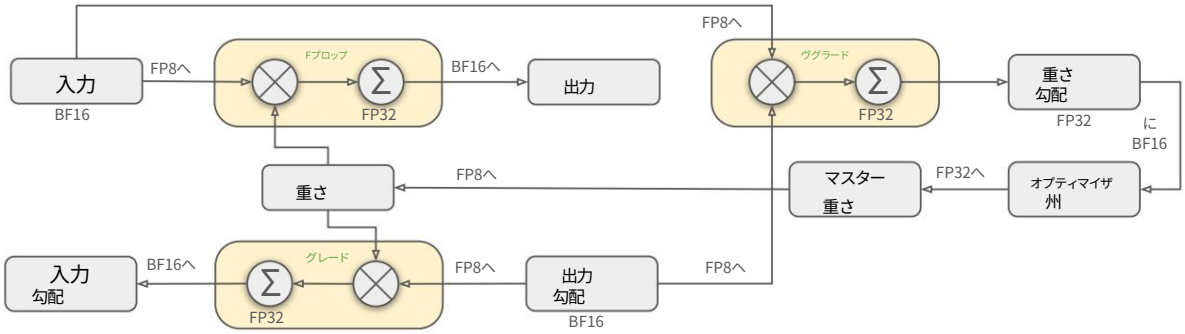


図 6 | FP8 データ形式を使用した全体的な混合精度フレームワーク。説明のために、線形演算子のみが示されています。

FP8 形式の動的範囲を効果的に拡張するために、きめ細かい量子化戦略を導入します。つまり、Output-Activation_{L+1} を 1× 要素でタイルごとにグループ化するか、また × 要素でブロックごとにグループ化します。関連する逆量子化のオーバーヘッドは、FP8 一般行列乗算 (GEMM) を正確に累積プロセスによって大幅に軽減されます。さらに、MoE トレーニングでの FP8 乗算の多くが 1 バイトのみを消費するために、FP8 でアクティベーションをキャッシュしてディスパッチし、低精度のオプティマイザー状態を BF16 に格納します。DeepSeek -V2-Lite と DeepSeek-V2 に類似した 2 つのモデル スケールで、約 1 兆トークンのトレーニングを行い、提案された FP8 混合精度フレームワークを検証します (詳細は付録 B.1 を参照)。特に、BF16 ベースラインと比較すると、FP8 トレーニング モデルの相対損失誤差は一貫して 0.25% 未満であり、トレーニングのランダム性の許容範囲内のレベルに十分収まっています。

達成するための重要な側面である、精度を高めます。

3.3.1. 混合精度フレームワーク

低精度トレーニングで広く採用されている手法 (Kalamkar et al., 2019; Narang et al., 2017) を基に、FP8 トレーニング用の混合精度フレームワークを提案します。このフレームワークでは、ほとんどの計算密度操作は FP8 で実行されますが、いくつかの重要な操作は、トレーニング効率と数値安定性のバランスをとるために、戦略的に元のデータ形式で維持されます。全体的なフレームワークを図 6 に示します。

まず、モデルのトレーニングを高速化するために、コア計算カーネルの大部分、つまりGEMM 演算は FP8 精度で実装されています。これらの GEMM 演算は、FP8テンソルを入力として受け入れ、BF16 または FP32 で出力を生成します。図 6 に示すように、線形演算子に関連付けられた 3 つの GEMM 、つまりFprop (フォワード パス)、Dgrad (アクティベーション バックワード パス)、およびWgrad (重みバックワード パス) はすべて FP8 で実行されます。この設計により、理論上は元の BF16 方式と比較して計算速度が 2 倍になります。さらに、FP8 Wgrad GEMM では、アクティベーションを FP8 に保存してバックワード パスで使用できます。これにより、メモリ消費が大幅に削減されます。

FP8 形式の効率性の利点にもかかわらず、一部の演算子は低精度の計算に敏感であるため、依然としてより高い精度が必要です。また、一部の低コストの演算子は、全体的なトレーニング コストへのオーバーヘッドを無視できるほどに高い精度を利用することもできます。このため、慎重な調査の結果、埋め込みモジュール、出力ヘッド、MoE ゲーティング モジュール、正規化演算子、およびアテンション演算子のコンポーネントについては、元の精度 (BF16 または FP32 など) を維持しています。これらの高精度の維持により、 DeepSeek-V3 の安定したトレーニング ダイナミクスが保証されます。数値安定性をさらに保証するために、マスター重み、重み勾配、およびオプティマイザーの状態をより高い精度で保存します。

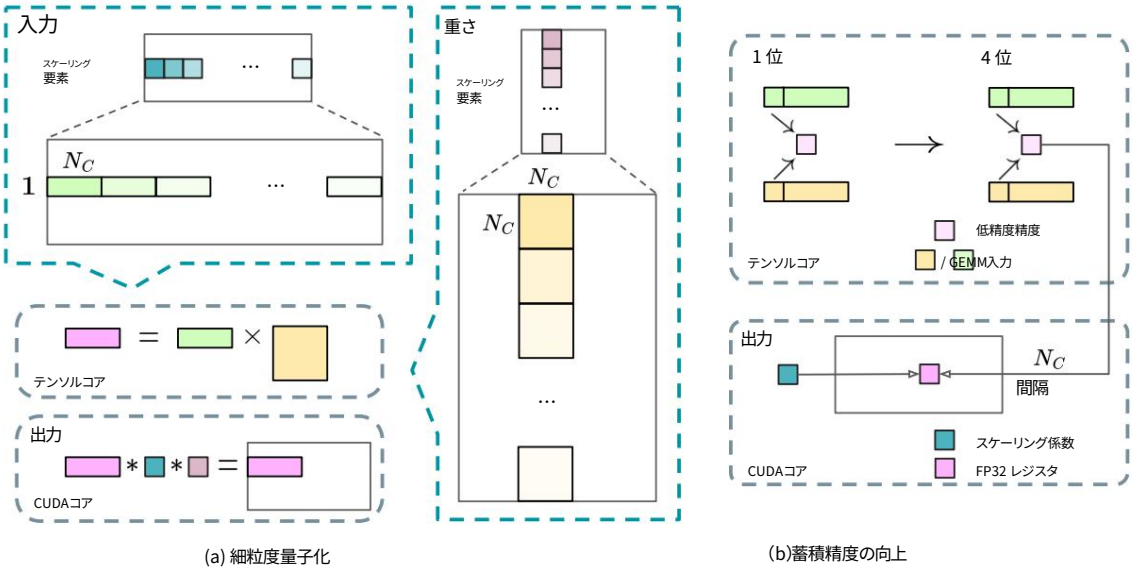


図 7 | (a)特徴の外れ値によって引き起こされる量子化誤差を軽減するための細粒度量子化方法を提案します。説明を簡略化するために、 Fpropのみを示しています。 (b)量子化戦略と組み合わせて、高精度の累積のために = 128 要素 MMA の間隔で CUDA コアに昇格することにより、FP8 GEMM の精度を向上させます。

これらの高精度コンポーネントはメモリのオーバーヘッドを発生させますが、分散トレーニング システム内の複数の DP ランクにわたる効率的なシャーディングによってその影響を最小限に抑えることができます。

3.3.2. 量子化と乗算による精度の向上

混合精度 FP8 フレームワークに基づいて、量子化方法と乗算プロセスの両方に焦点を当て、低精度のトレーニング精度を向上させるいくつかの戦略を紹介します。

細粒度量子化。低精度のトレーニング フレームワークでは、指数ビットの縮小によって制約される FP8 形式のダイナミック レンジが限られているため、オーバーフローとアンダーフローが一般的な課題となります。標準的な方法として、入力分布は、入力テンソルの最大絶対値を FP8 の最大表現可能値にスケーリングすることにより、FP8 形式の表現可能範囲に揃えられます (Narang ら、2017)。この方法では、低精度のトレーニングがアクティベーション外れ値に対して非常に敏感になり、量子化の精度が大幅に低下する可能性があります。この問題を解決するために、より細かいレベルでスケーリングを適用する細粒度量子化方法を提案します。図 7 (a) に示すように、(1) アクティベーションでは、 1x128タイル ベース (つまり、トークンあたり 128 チャネル)で要素をグループ化してスケーリングします。 (2)重みについては、 128x128ブロックベース（つまり、128入力チャネルごとに128出力チャネル）で要素をグループ化してスケーリングします。このアプローチにより、要素のより小さなグループに応じてスケールを適応させることで、量子化プロセスが外れ値に適切に対応できるようになります。付録B.2では、重みの量子化と同じ方法でブロックベースでアクティベーションをグループ化してスケーリングする場合のトレーニングの不安定性についてさらに説明します。

私たちの方法の重要な変更点の1つは、GEMM演算の内部次元に沿ったグループごとのスケーリング係数の導入です。この機能は、標準のFP8 GEMMでは直接サポートされていません。しかし、私たちの正確なFP32累積戦略と組み合わせることで、

効率的に実施される。

特に、私たちのきめ細かい量子化戦略はマイクロスケーリングフォーマットの考え方と非常に一致しています (Rouhani et al.、2023b)。一方、NVIDIA の次世代 GPU (Blackwell シリーズ) の Tensor コアは、より細かい量子化粒度のマイクロスケーリングフォーマットのサポートを発表しました (NVIDIA、2024a)。私たちの設計が、最新の GPU アーキテクチャに追いつくための将来の作業の参考になることを願っています。

累積精度の向上。低精度の GEMM 演算は、アンダーフローの問題に悩まされることが多く、その精度は主に高精度累積に左右されます。高精度累積は、一般的に FP32 精度で実行されます (Kalamkar ら、2019 年、Narang ら、2017 年)。ただし、 NVIDIA H800 GPU 上の FP8 GEMM の累積精度は、FP32 累積精度よりも大幅に低い約 14 ビットの保持に制限されていることがわかります。この問題は、内部次元Kが大きい場合 (Wortsman ら、 2023 年)、より顕著になります。これは、バッチ サイズとモデル幅が増加する大規模モデル トレーニングの一般的なシナリオです。たとえば、 K = 4096の 2 つのランダム マトリックスの GEMM 演算を予備テストで取り上げると、Tensor Cores の累積精度が制限されているため、最大相対誤差は約 2% になります。これらの問題にもかかわらず、限られた累積精度は依然としていくつかの FP8 フレームワーク (NVIDIA、2024b) のデフォルト オプションであり、トレーニングの精度が著しく制限されています。

この問題に対処するために、我々はより高い精度を得るためにCUDAコアへの昇格戦略を採用しています（Thakkar et al.、2023）。そのプロセスを図7（b)に示します。具体的には、 TensorコアでのMMA（行列乗算累積)実行中に、限られたビット幅を使用して中間結果が累積されます。の間隔に達すると、これらの部分的な結果はCUDAコアのFP32レジスタにコピーされ、そこでフル精度のFP32累積が実行されます。前述のように、私たちの細粒度量子化は、内部次元Kに沿ってグループごとのスケーリング係数を適用します。これらのスケーリング係数は、最小限の追加計算コストで、逆量子化プロセスとしてCUDAコア上で効率的に乗算できます。

この変更により、単一のワープグループに対する WGMMA (ワープグループ レベルの行列乗算累積) 命令の発行率が下がることは注目に値します。ただし、H800 アーキテクチャでは、2 つの WGMMA が同時に存続するのが一般的です。つまり、1 つのワープグループが昇格操作を実行している間に、もう 1 つのワープグループが MMA 操作を実行できます。この設計により、 2 つの操作のオーバーラップが可能になり、Tensor コアの高い使用率を維持できます。当社の実験に基づくと、 = 128 要素 (4 つの WGMMA に相当) の設定は、大幅なオーバーヘッドを発生させずに精度を大幅に向上できる最小の累積間隔を表します。

指数に対する仮数。以前の研究 (NVIDIA、2024b、Peng ら、2023b、Sun ら、2019b) で採用されたハイブリッド FP8 形式とは対照的に、 Fprop ではE4M3 (4 ビット指数と 3 ビット仮数) 、 DgradとWgradではE5M2 (5 ビット指数と 2 ビット仮数)を使用しますが、私たちはより高い精度を得るためにすべてのテンソルでE4M3形式を採用しています。このアプローチの実現可能性は、タイルとブロック単位のスケーリングというきめ細かい量子化戦略によるものです。より小さな要素グループを操作することで、私たちの方法論はこれらのグループ化された要素間で指数ビットを効果的に共有し、限られたダイナミック レンジの影響を軽減します。

オンライン量子化。遅延量子化はテンソル単位の量子化フレームワーク（NVIDIA、2024b; Peng et al.、2023b）で採用されており、最大絶対値の履歴を保持します。

以前の反復全体の値を使用して現在の値を推測します。正確なスケールを確保し、フレームワークを簡素化するために、 1x128 のアクティベーションタイルまたは128x128 の重みブロックごとに最大絶対値をオンラインで計算します。それに基づいてスケーリング係数を導出し、アクティベーションまたは重みをオンラインで FP8 形式に量子化します。

3.3.3. 低精度のストレージと通信

FP8 トレーニング フレームワークと組み合わせて、キャッシュされたアクティベーションとオプティマイザーの状態を低精度の形式に圧縮することで、メモリ消費と通信オーバーヘッドをさらに削減します。

低精度のオプティマイザー状態。FP32ではなく BF16 データ形式を採用して、 AdamW (Loshchilov および Hutter、2017) オプティマイザーの 1 次モーメントと 2 次モーメントを追跡し、目に見えるパフォーマンスの低下を招かないようにしています。ただし、マスター ウェイト (オプティマイザーによって保存) と勾配 (バッチ サイズの累積に使用) は、トレーニング全体を通じて数値の安定性を確保するために、FP32 に保持されています。

低精度アクティベーション。図 6 に示すように、 Wgrad演算は FP8 で実行されます。
メモリ消費量を削減するには、線形演算子の逆方向パスのアクティベーションを FP8 形式でキャッシュするのが自然な選択です。ただし、低コストで高精度のトレーニングを行うには、いくつかの演算子について特別な考慮が必要です。

(1)注意演算子後の線形入力。これらの活性化は注意演算子の逆方向パスでも使用されるため、精度に敏感になります。これらの活性化専用のカスタマイズされたE5M6データ形式を採用しています。

さらに、これらのアクティベーションは、逆方向パスで1x128 の量子化タイルから128x1タイルに変換されます。余分な量子化エラーが発生しないように、すべてのスケーリング係数は丸められ、つまり 2 の整数乗になります。

(2) MoEにおけるSwiGLU演算子の入力。メモリコストをさらに削減するために、 SwiGLU演算子の入力をキャッシュし、その出力を逆方向パスで再計算します。これらのアクティベーションは、メモリ効率と計算精度のバランスをとるために、細粒度量子化法を使用してFP8にも保存されます。

低精度通信。通信帯域幅は、MoE モデルのトレーニングにおける重大なボトルネックです。この課題を軽減するために、MoE アッププロジェクションの前にアクティベーションを FP8 に量子化し、 MoE アッププロジェクションの FP8 Fpropと互換性のあるディスパッチコンポーネントを適用します。アテンション演算子の後のLinearの入力と同様に、このアクティベーションのスケーリング係数は 2 の整数乗です。同様の戦略が、 MoE ダウンプロジェクションの前のアクティベーション勾配に適用されます。フォワードおよびバックワードの両方の結合コンポーネントについて、トレーニングパイプラインの重要な部分でトレーニング精度を維持するために、BF16 で保持します。

3.4. 推論と展開

DeepSeek-V3 は H800 クラスターに導入され、各ノード内の GPU はNVLink を使用して相互接続され、クラスター全体のすべての GPU は IB を介して完全に相互接続されます。オンライン サービスのサービス レベル目標 (SLO) と高スループットの両方を同時に確保するために、事前入力ステージとデコード ステージを分離する次の導入戦略を採用しています。

3.4.1. 事前入力

事前入力ステージの最小展開単位は、32 個の GPU を備えた 4 つのノードで構成されます。注意部分では、シーケンス並列処理 (SP) を備えた 4 方向テンソル並列処理 (TP4) と、 8 方向データ並列処理 (DP8) を組み合わせて使用します。TP サイズが 4 と小さいため、TP通信のオーバーヘッドが制限されます。MoE部分では、32 方向エキスパート並列処理 (EP32) を使用します。これにより、各エキスパートが十分に大きなバッチ サイズを処理し、計算効率が向上します。

MoEの全対全通信では、トレーニングと同じ方法を使用します。まずIB を介してノード間でトークンを転送し、次に NVLink を介してノード内 GPU 間で転送します。特に、浅いレイヤーの密な MLP には 1 方向のテンソル並列処理を使用して、TP通信を節約します。

MoE部分の異なるエキスパート間で負荷分散を実現するには、各 GPU がほぼ同じ数のトークンを処理するようにする必要があります。このために、冗長エキスパートの展開戦略を導入し、高負荷のエキスパートを複製して冗長展開します。高負荷のエキスパートは、オンライン展開中に収集された統計に基づいて検出され、定期的に (たとえば、10 分ごとに) 調整されます。冗長エキスパートのセットを決定した後、観測された負荷に基づいてノード内の GPU 間でエキスパートを慎重に再配置し、ノード間の全対全通信のオーバーヘッドを増やすことなく、GPU 間で負荷をできるだけ分散するように努めます。DeepSeek-V3 の展開では、事前入力ステージ用に 32 個の冗長エキスパートを設定しました。各 GPU では、ホストする元の 8 個のエキスパートに加えて、追加の冗長エキスパートを 1 個ホストします。

さらに、事前入力段階では、スループットを向上させ、全対全通信および TP 通信のオーバーヘッドを隠すために、同様の計算ワークロードを持つ 2 つのマイクロバッチを同時に処理し、 1 つのマイクロバッチの注意とMoEを別のマイクロバッチのディスパッチと結合と重ね合わせます。

最後に、エキスパートの動的冗長性戦略を検討しています。この戦略では、各 GPU がより多くのエキスパート (たとえば、16 人のエキスパート) をホストしますが、各推論ステップでアクティブになるのは 9 人だけです。各レイヤーでの全対全操作が開始される前に、グローバルに最適なルーティング スキームをオンザフライで計算します。事前入力段階でかなりの計算が行われることを考えると、このルーティング スキームを計算するオーバーヘッドはほぼ無視できます。

3.4.2. デコード

デコード中、共有エキスパートをルーティングされたエキスパートとして扱います。この観点から、各トークンはルーティング中に 9 つのエキスパートを選択します。共有エキスパートは、常に選択される高負荷のエキスパートと見なされます。デコード ステージの最小展開単位は、 320 GPU を備えた 40 ノードで構成されます。アテンション部分では、DP80 と組み合わせた SP を備えた TP4 を使用し、 MoE部分ではEP320 を使用します。MoE 部分では、各 GPU は 1 つのエキスパートのみをホストし、64 個の GPU が冗長エキスパートと共有エキスパートをホストする役割を担います。ディスパッチ部分と結合部分の全対全通信は、低レイテンシを実現するために、IB 経由の直接ポイントツーポイント転送を介して実行されます。さらに、IBGDA (NVIDIA、2022) テクノロジーを活用して、レイテンシをさらに最小限に抑え、通信効率を高めています。

事前入力と同様に、オンライン サービスからの統計的なエキスパート負荷に基づいて、一定の間隔で冗長エキスパートのセットを定期的に決定します。ただし、各 GPU は 1 つのエキスパートのみをホストするため、エキスパートを再配置する必要はありません。また、デコード用の動的冗長戦略も検討しています。ただし、これには、オーバーヘッドを削減するために、グローバルに最適なルーティング スキームとディスパッチカーネルとの融合を計算するアルゴリズムのより慎重な最適化が必要です。

さらに、スループットを向上させ、全対全通信のオーバーヘッドを隠すために、デコード段階で同様の計算ワークロードを持つ 2 つのマイクロバッチを同時に処理することも検討しています。事前入力とは異なり、デコード段階では注意が多くの時間を消費します。そのため、 1 つのマイクロバッチの注意を別のマイクロバッチのディスパッチ + MoE + 結合と重ねます。デコード段階では、エキスパートあたりのバッチ サイズは比較的小さく (通常 256 トークン以内)、ボトルネックとなるのは計算ではなくメモリ アクセスです。MoE 部分では 1 つのエキスパートのパラメーターをロードするだけでよいため、メモリアクセスのオーバーヘッドは最小限に抑えられ、SM を少なくしても全体的なパフォーマンスに大きな影響を与えることはありません。したがって、注意部分の計算速度に影響を与えないように、 SM のごく一部だけをディスパッチ + MoE + 結合に割り当てることができます。

3.5. ハードウェア設計に関する提案

全対全通信と FP8 トレーニング スキームの実装に基づいて、 AI ハードウェア ベンダーにチップ設計に関する次の提案をします。

3.5.1. 通信ハードウェア

DeepSeek-V3 では、計算と通信のオーバーラップを実装して、計算中の通信遅延を隠します。これにより、シリアル計算と通信に比べて、通信帯域幅への依存が大幅に軽減されます。ただし、現在の通信実装は高価な SM に依存しており (たとえば、H800 GPU で使用可能な 132 個の SM のうち 20 個をこの目的に割り当てています)、計算スループットが制限されます。さらに、SM を通信に使用すると、テンソル コアが完全に十分に活用されないため、大幅な非効率が生じます。

現在、SM は主に、全員対全員の通信に対して次のタスクを実行します。

- 単一の GPU から同じノード内の複数の GPU 宛てのIB トラフィックを集約しながら、IB (InfiniBand) と NVLink ドメイン間でデータを転送します。
- RDMA バッファー (登録された GPU メモリ領域) と入出力バッファー間でデータを転送します。
- すべて対すべての結合のための削減操作を実行します。
- IB および NVLink ドメイン全体で複数のエキスパートにチャンク データを転送する際のきめ細かいメモリ レイアウトを管理します。

将来、ベンダーがこれらの通信タスクを貴重な計算ユニット SM からオフロードし、NVIDIA SHARP Graham ら (2016) のような GPU コプロセッサまたはネットワーク コプロセッサとして機能するハードウェアを開発することを期待しています。さらに、アプリケーション プログラミングの複雑さを軽減するために、このハードウェアは計算ユニットの観点から IB (スケールアウト) および NVLink (スケールアップ) ネットワークを統合することを目指しています。この統合インターフェイスにより、計算ユニットは、単純なプリミティブに基づいて通信要求を送信することにより、IB-NVLink 統合ドメイン全体で読み取り、書き込み、マルチキャスト、リデュースなどの操作を簡単に実行できます。

3.5.2. コンピューティングハードウェア

Tensor コアにおける FP8 GEMM の累積精度の向上。NVIDIA Hopper アーキテクチャの現在の Tensor コア実装では、FP8 GEMM の累積精度が限られています。最大指数に基づいて右シフトして 32 個の仮数積を整列させた後、Tensor コアは各仮数積の上位 14 ビットのみを加算に使用します。

この範囲を超えるビットは切り捨てられます。加算結果をレジスタに蓄積する場合も、 14 ビットの精度が使用されます。私たちの実装では、128 回のFP8×FP8乗算の加算結果をCUDA コアの FP32 精度のレジスタに蓄積することで、この制限を部分的に緩和しています。これは FP8 トレーニングを成功させるのに役立ちますが、FP8 GEMM 蓄積精度における Hopper アーキテクチャのハードウェアの欠陥による妥協にすぎません。

将来のチップはより高い精度を採用する必要があります。

タイル単位およびブロック単位の量子化のサポート。現在の GPU はテンソル単位の量子化のみをサポートしており、タイル単位やブロック単位の量子化のようなきめの細かい量子化のネイティブ サポートがありません。現在の実装では、間隔に達すると、部分的な結果が Tensor Core から CUDA Core にコピーされ、スケーリング係数で乗算されて、 CUDA Core の FP32 レジスタに追加されます。精密な FP32 累積戦略と組み合わせることで、逆量子化のオーバーヘッドは大幅に軽減されますが、 Tensor Core と CUDA Core 間の頻繁なデータ移動によって、計算効率が制限されます。したがって、将来のチップでは、Tensor Core がスケーリング係数を受け取れるようにし、グループ スケーリングで MMA を実装することで、きめの細かい量子化をサポートすることを推奨します。このようにして、部分和の累積と逆量子化全体を Tensor Core 内で直接完了し、最終結果が生成されるため、頻繁なデータ移動を回避できます。

オンライン量子化のサポート。現在の実装では、オンライン量子化の有効性が私たちの研究で実証されているにもかかわらず、オンライン量子化を効果的にサポートするのに苦労しています。既存のプロセスでは、量子化のために HBM (高帯域幅メモリ) から 128 個の BF16 アクティベーション値 (前回の計算の出力) を読み取る必要があり、量子化された FP8 値は HBM に書き戻され、MMA のために再度読み取られます。この非効率性に対処するために、将来のチップでは FP8 キャストと TMA (テンソル メモリ アクセラレータ) アクセスを 1 つの融合操作に統合することをお勧めします。これにより、グローバル メモリから共有メモリへのアクティベーションの転送中に量子化が完了し、頻繁なメモリの読み取りと書き込みを回避できます。また、高速化のためにワープ レベルのキャスト命令をサポートすることもお勧めします。これにより、レイヤー正規化と FP8 キャストのより優れた融合がさらに促進されます。あるいは、コンピューティング ロジックを HBM の近くに配置したニア メモリ コンピューティング アプローチを採用することもできます。この場合、BF16要素は HBM から GPU に読み込まれると FP8 に直接キャストできるため、オフチップメモリ アクセスが約 50% 削減されます。

転置 GEMM 操作のサポート。現在のアーキテクチャでは、行列転置と GEMM 操作を融合するのが面倒です。私たちのワークフローでは、フォワード パス中のアクティベーションは1x128 FP8 タイルに量子化されて保存されます。バックワード パス中は、行列を読み出し、逆量子化、転置、 128x1タイルに再量子化してHBM に保存する必要があります。メモリ操作を減らすために、トレーニングと推論の両方で必要な精度については、MMA 操作の前に共有メモリから行列を直接転置して読み取ることができるように将来のチップを推奨します。FP8 形式変換と TMA アクセスの融合と組み合わせることで、この機能強化により量子化ワークフローが大幅に効率化されます。

4. 事前トレーニング

4.1. データ構築

DeepSeek-V2と比較して、数学的サンプルとプログラミングサンプルの比率を高め、多言語カバレッジを拡大することで事前トレーニングコーパスを最適化しました。

英語と中国語。また、データ処理パイプラインは、コーパスの多様性を維持しながら冗長性を最小限に抑えるように改良されています。Ding et al. (2024) に触発され、データの整合性のためにドキュメントパッキング方法を実装していますが、トレーニング中にクロスサンプルアテンションマスキングは組み込んでいません。最後に、DeepSeek-V3 のトレーニングコーパスは、トークナイザー内の 14.8T の高品質で多様なトークンで構成されています。

DeepSeekCoder-V2 (DeepSeek-AI、2024a) のトレーニング プロセスでは、 Fill-in-Middle (FIM) 戦略によって次のトークンの予測機能が損なわれることなく、コンテキスト キューに基づいてモデルが中間のテキストを正確に予測できることが確認されています。DeepSeekCoder-V2 に合わせて、DeepSeek-V3 の事前トレーニングにも FIM 戦略を組み込んでいます。具体的には、Prefix-Suffix-Middle (PSM) フレームワークを使用して、次のようにデータを構造化します。

<|fim_begin|>前<|fim_hole|> suf<|fim_end|>中間<|eos_token|>。

この構造は、事前梱包プロセスの一部としてドキュメント レベルに適用されます。 FIM戦略は、PSM フレームワークと一致して 0.1 のレートで適用されます。

DeepSeek-V3 のトークナイザーは、128K トークンの拡張語彙を備えたバイトレベルの BPE (Shibata et al., 1999) を採用しています。当社のトークナイザーのプレトークナイザーとトレーニング データは、多言語圧縮効率を最適化するように変更されています。さらに、DeepSeek-V2 と比較して、新しいプレトークナイザーは句読点と改行を組み合わせたトークンを導入しています。ただし、このトリックにより、モデルが終端の改行のない複数行のプロンプトを処理する場合、特に少数ショットの評価プロンプトの場合、トークン境界バイアス (Lundberg、2023) が発生する可能性があります。
この問題に対処するために、トレーニング中にこのような結合トークンの一定の割合をランダムに分割します。これにより、モデルはより広範囲の特殊なケースにさらされ、このバイアスが軽減されます。

4.2. ハイパーパラメータ

モデルのハイパーパラメータ。Transformerレイヤーの数を 61 に、隠し次元を 7168 に設定します。学習可能なすべてのパラメータは、標準偏差0.006 でランダムに初期化されます。MLA では、アテンション ヘッドの数を 128 に設定します。KV 圧縮次元は 512 に設定され、クエリ圧縮次元'は 1536 に設定されています。分離されたクエリとキーの場合、ヘッドあたりの次元を 64 に設定します。最初の 128 および 1 人あたりの寸法
3 つのレイヤーを除くすべての FFN を MoE レイヤーに置き換えます。各 MoE レイヤーは、1 つの共有エキスパートと 256 のルーティングされたエキスパートで構成され、各エキスパートの中間隠し次元は2048 です。ルーティングされたエキスパートのうち、8 つのエキスパートがトークンごとにアクティブ化され、各トークンは最大 4 つのノードに送信されるようになります。マルチトークン予測の深さは 1 に設定されています。つまり、次のトークンの他に、各トークンは 1 つの追加トークンを予測します。 DeepSeek-V2 と同様に、DeepSeek-V3 も圧縮された潜在ベクトルの後に追加の RMSNorm レイヤーを採用し、幅のボトルネックで追加のスケーリング係数を乗算します。この構成では、 DeepSeek-V3 は合計 671B のパラメータで構成され、そのうち 37B がトークンごとにアクティブ化されます。

ハイパーパラメータのトレーニング。ハイパーパラメータを1 = 0.95、 weight_decay = 0.1 に設定して、 AdamW オプティマイザー (Loshchilov および Hutter、2017) を使用しますで事前トレーニング中に、最大値を0.9、シーケンス長を2に 4K に設定し、14.8T トークDeepSeek-V3 を事前トレーニングします。学習率のスケジュールについては、最初の 2K ステップで最初に 0 から 2.2 × 10－4まで線形に増加します。次に、モデルが 10T トレーニング トークンを消費するまで、学習率を 2.2 × 10－4に一定に保ちます。その後、コサイン減衰曲線に従って、4.3T トークンで学習率を 2.2 × 10－5まで徐々に減衰させます。最後の500Bトークンのトレーニング中、最初の333Bトークンでは2.2 × 10－5の一定学習率を維持し、別の一定学習率に切り替えます。

残りの 167B トークンでは 7.3 × 10－6です。勾配クリッピング ノルムは 1.0 に設定されています。バッチサイズ スケジューリング戦略を採用し、最初の 469B トークンのトレーニングではバッチ サイズを 3072 から 15360 に徐々に増やし、残りのトレーニングでは 15360 を維持します。パイプラインの並列処理を利用して、モデルの異なるレイヤーを異なる GPU に展開し、各レイヤーでは、ルーティングされたエキスパートが 8 つのノードに属する 64 個の GPU に均一に展開されます。

ノード制限ルーティングの場合、各トークンは最大 4 つのノード (つまり、 = 4) に送信されます。補助損失のない負荷分散の場合、最初の 14.3Tトークンのバイアス更新速度を 0.001 に設定し、残りの 500B トークンを 0.0 に設定します。バランス損失の場合、単一のシーケンス内での極端な不均衡を回避するために、0.0001 に設定します。MTP損失重みは、最初の 10T トークンを 0.3 に設定し、残りの 4.8T トークンを 0.1 に設定します。

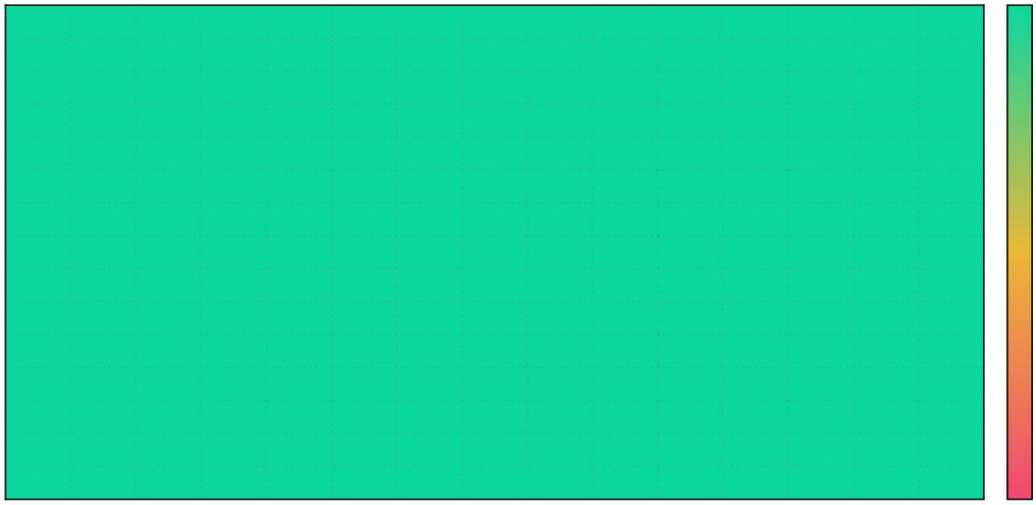


図 8 | 「Needle In A Haystack」(NIAH) テストの評価結果。DeepSeek-V3 は、最大 128K までのすべてのコンテキスト ウィンドウ長で良好なパフォーマンスを発揮します。

4.3. ロングコンテキスト拡張

DeepSeek-V3 でロングコンテキスト機能を有効にするために、DeepSeek-V2 (DeepSeek-AI、2024c) と同様のアプローチを採用しています。事前トレーニング段階の後、コンテキスト拡張に YaRN (Peng ら、2023a) を適用し、それぞれ 1000 ステップで構成される 2 つの追加トレーニング フェーズを実行して、コンテキスト ウィンドウを 4K から 32K、さらに 128K へと段階的に拡張します。YaRN 構成は DeepSeek-V2 で使用されているものと一致しており、分離された共有キーkにのみ適用されます。ハイパーパラメータは両方のフェーズで同一のままで、スケール= 40、 = 1、 = 32、スケーリング係数√ = 0.1 ln + 1 です。最初のフェーズでは、シーケンス長は 32K に設定され、バッチサイズは 1920 です。2 番目のフェーズでは、シーケンス長は128K に増加され、バッチサイズは 480 に減少します。両方のフェーズの学習率は7.3 × 10－6に設定され、事前トレーニング段階の最終学習率と一致します。

この2段階の拡張トレーニングにより、DeepSeek-V3は強力なパフォーマンスを維持しながら、最大128Kの長さの入力を処理できます。図8は、監督された微調整後のDeepSeek-V3が「Needle In A Haystack」で顕著なパフォーマンスを達成していることを示しています。

(NIAH) テストでは、最大 128K のコンテキスト ウィンドウ長にわたって一貫した堅牢性が実証されています。

4.4. 評価

4.4.1. 評価ベンチマーク

DeepSeek-V3 の基本モデルは、英語と中国語が大部分を占める多言語コーパスで事前トレーニングされているため、主に英語と中国語の一連のベンチマークと多言語ベンチマークでそのパフォーマンスを評価します。評価は、HAI-LLM フレームワークに統合された社内評価フレームワークに基づいています。検討対象のベンチマークは次のように分類され、リストされています。下線付きのベンチマークは中国語のベンチマークで、二重下線付きのベンチマークは多言語のベンチマークです。

複数被験者の多肢選択データセットには、MMLU (Hendrycks et al., 2020)、MMLU- Redux (Gema et al., 2024)、MMLU-Pro (Wang et al., 2024b)、 MMMLU (OpenAI, 2024b)、 C-Eval (Huang et al., 2023)、CMMLU (Li et al., 2023) が含まれます。

言語理解および推論データセットには、HellaSwag (Zellers et al., 2019)、 PIQA (Bisk et al., 2020)、ARC (Clark et al., 2018)、BigBench Hard (BBH) (Suzgun et al., 2022) などがあります。

クローズドブック質問応答データセットには、TriviaQA（Joshi et al., 2017）やNaturalQuestions（Kwiatkowski et al.、2019）。

読解データセットには、RACE Lai et al. が含まれます。 (2017)、DROP (Dua et al.、2019)、 C3 (Sun et al.、2019a)、CMRC (Cui et al.、2019)。

参照曖昧性解消データセットには、 CLUEWSC （Xu et al.、2020）とWinoGrandeが含まれる。坂口ら（2019）。

言語モデリングデータセットにはPile（Gao et al.、2020）が含まれます。

中国語の理解と文化に関するデータセットには、CCPM（Li et al.、2021）が含まれます。

数学データセットには、GSM8K（Cobbe et al., 2021）、MATH（Hendrycks et al., 2021）、MGSMが含まれます。(Shi et al., 2023)、CMath (Wei et al., 2023)。

コードデータセットには、HumanEval (Chen et al., 2021)、LiveCodeBench-Base (0801-1101) (Jain et al., 2024)、MBPP (Austin et al., 2021)、および CRUXEval (Gu et al., 2024) が含まれます。

標準化された試験にはAGIEval （Zhong et al., 2023）が含まれます。AGIEvalには、英語と中国語のサブセット。

前回の研究 (DeepSeek-AI、2024b、c) に続き、HellaSwag、PIQA、WinoGrande、RACE-Middle、RACE-High、 MMLU、MMLU-Redux、MMLU-Pro、MMMLU、ARC-Easy、ARC-Challenge、C-Eval、CMMLU、 C3、CCPM などのデータセットに対してはパープレキシティ ベースの評価を採用し、TriviaQA、NaturalQuestions、DROP、 MATH、GSM8K、MGSM、HumanEval、MBPP、LiveCodeBench-Base、CRUXEval、BBH、AGIEval、CLUEWSC、CMRC、CMath に対しては生成ベースの評価を採用します。さらに、Pile-test に対して言語モデリング ベースの評価を実行し、異なるトークナイザーを使用するモデル間の公平な比較を保証するために、Bits-Per-Byte (BPB) をメトリックとして使用します。

4.4.2. 評価結果

表 3 では、DeepSeek-V3 のベース モデルを、 DeepSeek-V2-Base (DeepSeek-AI、2024c) (以前のリリース)、Qwen2.5 72B Base (Qwen、2024b)、LLaMA-3.1 405B Base (AI@Meta、2024b) などの最先端のオープンソース ベース モデルと比較しています。これらのモデルはすべて社内の評価フレームワークで評価し、同じ評価設定が共有されるようにしています。
過去数か月にわたる評価フレームワークの変更により、パフォーマンスは

	ベンチマーク（メトリック）	# ショット	DeepSeek-V2 ベース	Qwen2.5 72Bベース	LLaMA-3.1 405Bベース	DeepSeek-V3 ベース
	建築	-	文部科学省	密集	密集	文部科学省
	# アクティブ化されたパラメータ	-	21B	72B	405B	37B
	# 合計パラメータ	-	236B	72B	405B	671B
英語	パイルテスト（BPB）	-	0.606	0.638	0.542	0.548
	BBH （EM）	3ショット 5	78.8	79.8	82.9	87.5
	MMLU （エムエル）	ショット 5	78.4	85.0	84.4	87.1
	MMLU-Redux (EM)	ショット 5	75.6	83.2	81.3	86.2
	MMLU-プロ（EM）	ショット 3	51.4	58.3	52.8	64.4
	ドロップ（F1）	ショット 25	80.4	80.6	86.0	89.0
	ARC-Easy (EM)	ショット 25シ	97.6	98.4	98.4	98.9
	ARCチャレンジ（EM）	ヨット 10ショ	92.2	94.5	95.3	95.3
	ヘラスワグ（EM）	ット 0ショッ	87.1	84.8	89.2	88.9
	ピカ（EM）	ト 5ショッ	83.9	82.6	85.9	84.7
	ウィノグランデ（EM）	ト 5ショッ	86.3	82.3	85.2	84.9
	RACE-ミドル（EM）	ト 5ショッ	73.1	68.1	74.2	67.1
	RACE-高(EM)	ト 5ショッ	52.6	50.3	56.8	51.3
	トリビアQA （EM）	ト 5ショッ	80.0	71.9	82.7	82.9
	ナチュラルクエスチョン（EM）	ト 0ショット	38.6	33.2	41.5	40.0
	AGIEval （EM）		57.5	75.8	60.6	79.6
コード	HumanEval (Pass@1) 0ショット 3ショット		43.3	53.0	54.9	65.2
	MBPP (パス@1)		65.0	72.6	68.4	75.4
	LiveCodeBench-Base (Pass@1) 3ショット		11.6	12.9	15.5	19.4
	CRUXEval-I (EM) 2ショット		52.5	59.1	58.5	67.3
	CRUXEval-O (EM) 2ショット		49.8	59.9	59.9	69.8
数学	GSM8K （EM）	8ショット	81.6	88.3	83.5	89.3
	数学（EM）	4ショット 8	43.4	54.4	49.0	61.6
	MGSM （EM）	ショット 3	63.6	76.2	69.9	79.8
	数学（EM）	ショット	78.7	84.5	77.3	90.7
中国語	CLUEWSC (EM)	5ショット 5	82.0	82.5	83.0	82.7
	C評価（EM）	ショット 5	81.4	89.2	72.5	90.1
	CMMLU （エムエルエム）	ショット	84.0	89.5	73.7	88.8
	CMRC （エムアールシー）	1ショット	77.4	75.8	76.0	76.3
	C3 （EM）	0ショット	77.4	76.7	79.7	78.6
	CCPM （EM）	0ショット	93.0	88.5	78.6	92.0
多言語 MMMLU 非英語(EM)		5発	64.0	74.8	73.8	79.4

表3 | DeepSeek-V3-Baseと他の代表的なオープンソースベースの比較
すべてのモデルは当社の内部フレームワークで評価され、同じ評価を共有します

設定。差が 0.3 を超えないスコアは、同じレベルであると見なされます。DeepSeek- V3-Base は、ほとんどのベンチマーク、特に数学およびコード タスクで最高のパフォーマンスを実現します。

DeepSeek-V2-Baseは、以前に報告した結果と若干異なる結果を示しています。全体的に、
DeepSeek-V3-Baseは、DeepSeek-V2-BaseおよびQwen2.5 72B Baseを総合的に上回ります。
LLaMA-3.1 405B Baseをほとんどのベンチマークで上回り、実質的に
最も強力なオープンソースモデル。

より詳細な観点から、DeepSeek-V3-Baseを他のオープンソースと比較します。
（１）DeepSeek-V2-Baseと比較して、
モデルアーキテクチャ、モデルサイズとトレーニング トークンのスケールアップ、そして強化
データ品質に関しては、DeepSeek-V3-Baseは予想通り大幅に優れたパフォーマンスを達成しました。(2)
最先端の中国オープンソースモデルであるQwen2.5 72Bベースと比較すると、
有効化されたパラメータの半分では、DeepSeek-V3-Baseも顕著な利点を示します。

特に英語、多言語、コード、数学のベンチマークでは、中国語のベンチマークに関しては、中国語の多肢選択問題であるCMMLUを除いて、DeepSeek-V3-Baseは Qwen2.5 72Bよりも優れた性能を発揮します。(3)LLaMA-3.1 405Bベースと比較して、最大の 11倍のアクティブパラメータを持つオープンソースモデルであるDeepSeek-V3-Baseも 多言語、コード、数学のベンチマークでパフォーマンスが大幅に向上しました。英語と 中国語のベンチマークでは、DeepSeek-V3-Baseが競争力以上のパフォーマンスを発揮し、 特に、BBH、MMLU シリーズ、DROP、C-Eval、CMMLU、CCPM に適しています。

効率的なアーキテクチャと包括的なエンジニアリングの最適化により、DeepSeek- V3は非常に高いトレーニング効率を実現しています。当社のトレーニングフレームワークとインフラストラクチャでは、1兆トークンごとにDeepSeek-V3をトレーニングするのに必要なH800 GPU時間はわずか180K時間です。 72B または 405B の高密度モデルをトレーニングするよりもはるかに安価です。

ベンチマーク（メトリック）	# ショット	小さな萌え ベースライン	小さな萌え MTP付き	大きな萌え ベースライン	大きな萌え MTP付き
# アクティブ化されたパラメータ(推論)	-	2.4億	2.4億	209億	209億
# 合計パラメータ（推論）	-	157億	157億	228.7億	228.7億
# トレーニングトークン	-	1.33T	1.33T	540B	540B
パイルテスト（BPB）	-	0.729	0.729	0.658	0.657
BBH （EM）	3ショット	39.0	41.4	70.0	70.7
MMLU （エムエル）	5発	50.0	53.3	67.5	66.6
ドロップ（F1）	1ショット	39.2	41.3	68.5	70.6
トリビアQA （EM）	5発	56.9	57.7	67.0	67.3
ナチュラルクエスチョン（EM）	5ショット 0	22.7	22.3	27.2	28.5
HumanEval (合格@1)	ショット 3	20.7	26.8	44.5	53.7
MBPP (パス@1)	ショット 8	35.8	36.8	61.6	62.2
GSM8K （EM）	ショット 4	25.4	31.4	72.3	74.0
数学（EM）	ショット	10.7	12.6	38.6	39.8

表4 | MTP戦略によるアブレーション結果。MTP戦略は一貫して ほとんどの評価ベンチマークにおけるモデルのパフォーマンス。

4.5. 議論

4.5.1. マルチトークン予測のためのアブレーション研究

表4にMTP戦略のアブレーション結果を示します。具体的には、 異なるスケールの2つのベースラインモデルに基づくMTP戦略。小規模では、 1.33Tトークンの合計157億のパラメータで構成されるベースラインMoEモデル。大規模では、 我々は5400億トークンで合計2287億のパラメータからなるベースラインMoEモデルを訓練した。 それらのトレーニングデータと他のアーキテクチャを同じに保ちながら、1深度を追加します MTPモジュールをそれらに導入し、比較のためにMTP戦略で2つのモデルをトレーニングします。 推論中はMTPモジュールを直接破棄するため、比較したモジュールの推論コストは モデルは全く同じです。表から、MTP戦略は一貫して ほとんどの評価ベンチマークでモデルのパフォーマンスが向上します。

4.5.2. 補助損失のないバランス調整戦略のためのアブレーション研究

表5に補助損失のないバランス戦略のアブレーション結果を示します。 この戦略を異なるスケールの2つのベースラインモデルで検証します。小規模では、 1.33Tトークンで合計157億のパラメータを含むベースラインMoEモデルをトレーニングします。 大規模では、5780億のトークンで合計2287億のパラメータを含むベースラインMoEモデルをトレーニングします。

ベンチマーク（メトリック）ショット数		小さな萌え 補助ロスベース	小さな萌え 補助ロスフリー	大きなMoE 補助ロスベース	大きなMoE 補助ロスフリー
# アクティブ化されたパラメータ	-	24億	24億	209億	209億
# 合計パラメータ	-	157億	157億	2287億	228.7億
# トレーニングトークン	-	1.33兆	1.33兆	5780億	578B
パイルテスト（BPB）	-	0.727	0.724	0.656	0.652
BBH (EM) 5	3ショット	37.3	39.3	66.7	67.9
MMLU (EM)	5ショット	51.0	51.8	68.3	67.2
ドロップ（F1）	ショット	38.1	39.0	67.1	67.1
トリビアQA （EM）	5ショット	58.3	58.5	66.7	67.7
NaturalQuestions (EM)	5ショット	23.2	23.4	27.1	28.1
HumanEval (Pass@1)	0ショット	22.0	22.6	40.2	46.3
MBPP (パス@1)	3ショット	36.6	35.8	59.2	61.2
GSM8K (EM)	8ショット	27.1	29.6	70.7	74.5
数学（EM）	4ショット	10.9	11.1	37.2	39.6

表5 |補助損失のないバランス戦略のアブレーション結果。
純粋に補助損失ベースの方法である補助損失のない戦略は、一貫してより良い結果を達成します。
ほとんどの評価ベンチマークにおけるモデルのパフォーマンス。

どちらのベースラインモデルも、負荷バランスを促進するために補助損失のみを使用し、
トップKアフィニティ正規化によるシグモイドゲーティング関数。制御するためのハイパーパラメータ
補助損失の強度はそれぞれDeepSeek-V2-LiteおよびDeepSeek-V2と同じです。
これら2つのベースラインモデルの上に、トレーニングデータと他のアーキテクチャを維持しながら、
同様に、補助損失をすべて除去し、補助損失のないバランス戦略を導入します。
比較すると、補助損失のない戦略は一貫して
ほとんどの評価ベンチマークでより優れたモデルパフォーマンスを実現します。

4.5.3. バッチ方式の負荷分散とシーケンス方式の負荷分散

補助損失のないバランス調整とシーケンスごとの補助損失の主な違いは、
バランス調整の範囲はバッチ方式とシーケンス方式の2つです。シーケンス方式と比べると
補助損失に対して、バッチワイズバランシングはより柔軟な制約を課す。
各シーケンスのドメイン内バランス。この柔軟性により、専門家はより専門的に
異なるドメイン。これを検証するために、 Pile テスト セットの異なるドメインで、16B 補助損失ベースの
ベースラインと 16B 補助損失フリー モデルのエキスパート負荷を記録して分析します。
図9に示すように、補助損失のないモデルでは、より大きな
予想通りの専門家の特化パターン。

この柔軟性とモデルの利点との相関関係をさらに調査するために
パフォーマンスを向上させるために、バッチごとの補助損失を設計し検証し、
各シーケンスではなく、各トレーニングバッチごとに負荷分散を行う。実験結果から、
同様のレベルのバッチ負荷バランスを達成すると、バッチ補助損失は
補助損失のない方法と同様のモデル性能を達成することもできる。具体的には、
1B MoEモデルを用いた実験では、検証損失は2.258（シーケンスワイズ補助損失を使用）、2.253（補助損失フリー法を使用）、2.253（バッチワイズ補助損失を使用）でした。
補助損失）。3B MoEモデルでも同様の結果が得られました。シーケンスワイズ補助損失を使用するモデルでは検証損失が2.085に達し、補助損失のないモデルでは検証損失が2.085に達しました。
方法またはバッチごとの補助損失は、同じ検証損失 2.080 を達成します。

さらに、バッチ方式の負荷分散方法は一貫したパフォーマンスを示しているが、
利点がある一方で、効率性に関して2つの潜在的な課題も抱えている。(1)内部の負荷不均衡

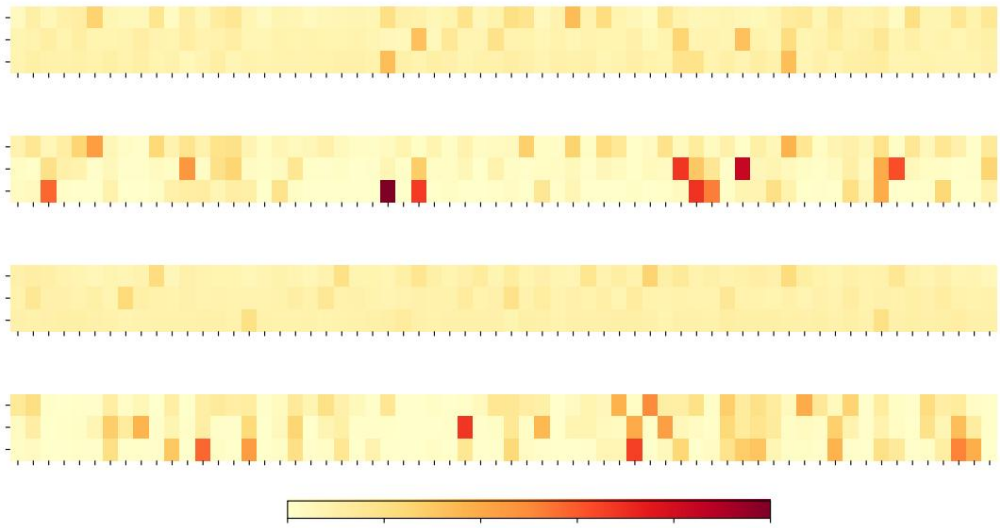


図 9 | Pile テスト セットの 3 つのドメインにおける補助損失なしモデルと補助損失ありモデルのエキスパート負荷。補助損失なしモデルは、補助損失ありモデルよりもエキスパート特化パターンが大きいことが示されています。相対エキスパート負荷は、実際のエキスパート負荷と理論的にバランスのとれたエキスパート負荷の比率を示します。スペースの制約により、例として 2 つのレイヤーの結果のみを示します。すべてのレイヤーの結果は付録 C に記載されています。

（1)特定のシーケンスまたは小さなバッチ、および（2)推論中のドメインシフトによる負荷の不均衡。最初の課題は、大規模なエキスパート並列処理とデータ並列処理を使用するトレーニングフレームワークによって自然に解決され、各マイクロバッチのサイズが大きく保証されます。2番目の課題についても、セクション3.4で説明するように、冗長なエキスパート展開を備えた効率的な推論フレームワークを設計および実装して、これを克服します。

5. 研修後

5.1. 教師あり微調整

私たちは、複数のドメインにまたがる 150 万のインスタンスを含むように命令チューニング データセットをキュレートし、各ドメインでは特定の要件に合わせて調整された異なるデータ作成方法を採用しています。

推論データ。数学、コード競争問題、論理パズルに重点を置いたものなど、推論関連のデータセットについては、社内のDeepSeek-R1 モデルを活用してデータを生成します。具体的には、R1 で生成されたデータは高い精度を示していますが、考えすぎ、不適切なフォーマット、長すぎるなどの問題があります。私たちの目標は、R1 で生成された推論データの高い精度と、通常のフォーマットの推論データの明快さと簡潔さのバランスを取ることです。

方法論を確立するために、まず、コード、数学、一般的な推論などの特定のドメインに合わせたエキスパートモデルを開発します。これには、教師ありファインチューニング (SFT) と強化学習 (RL) のトレーニングパイプラインを組み合わせて使用します。このエキスパートモデルは、最終モデルのデータジェネレーターとして機能します。トレーニングプロセスでは、各インスタンスに対して 2 つの異なるタイプの SFT サンプルを生成します。1 つ目は、問題と元の応答を <問題、元の応答> の形式で組み合わせたもので、2 つ目はシステムプロンプトを組み込んだものです。

問題と R1 応答を、<システム プロンプト、問題、R1 応答> の形式で入力します。

システム プロンプトは、モデルが反映と検証のメカニズムが充実した応答を生成するように導く指示を含むように細心の注意を払って設計されています。RL フェーズでは、明示的なシステム プロンプトがない場合でも、モデルは高温サンプリングを利用して、R1 で生成されたデータと元のデータの両方からのパターンを統合する応答を生成します。数百の RL ステップの後、中間 RL モデルはR1 パターンを組み込むことを学習し、全体的なパフォーマンスを戦略的に向上させます。

RL トレーニング フェーズが完了すると、拒否サンプリングを実装して、最終モデル用の高品質の SFT データをキュレートします。このとき、エキスパート モデルがデータ生成ソースとして使用されます。この方法により、最終的なトレーニング データは、簡潔で効果的な応答を生成しながら、DeepSeek-R1 の長所を維持できます。

非推論データ。創作文、ロールプレイ、簡単な質問への回答などの非推論データについては、DeepSeek-V2.5 を使用して応答を生成し、人間の注釈者を雇ってデータの正確性と正しさを検証します。

SFT 設定。SFTデータセットを使用して、5 × 10－6から始まり徐々に 1 × 10－6まで減少するコサイン減衰学習率スケジューリングを使用して、DeepSeek-V3-Base を 2 エポックにわたって微調整します。
トレーニング中、各シーケンスは複数のサンプルからパックされます。ただし、サンプル マスキング戦略を採用して、これらのサンプルが分離され、相互に見えない状態を維持します。

5.2. 強化学習

5.2.1. 報酬モデル

私たちは、RL プロセスでルールベースの報酬モデル (RM) とモデルベースの RM を採用しています。

ルールベースの RM。特定のルールを使用して検証できる質問については、ルールベースの報酬システムを採用してフィードバックを決定します。たとえば、特定の数学の問題には決定論的な結果があり、モデルが指定された形式 (ボックス内など) で最終回答を提供することを要求し、ルールを適用して正確性を検証できるようにします。同様に、LeetCode の問題では、コンパイラーを使用してテスト ケースに基づいてフィードバックを生成できます。可能な限りルールベースの検証を活用することで、このアプローチは操作や悪用に対して耐性があるため、より高いレベルの信頼性を確保できます。

モデルベースの RM。自由形式の正解回答がある質問の場合、報酬モデルを使用して、応答が予想される正解と一致するかどうかを判断します。逆に、創造的な執筆など、明確な正解がない質問の場合、報酬モデルは、質問とそれに対応する回答を入力としてフィードバックを提供する役割を担います。報酬モデルは、DeepSeek-V3 SFT チェックポイントからトレーニングされます。信頼性を高めるために、最終的な報酬を提供するだけでなく、報酬につながる思考の連鎖も含む選好データを構築します。このアプローチは、特定のタスクにおける報酬ハッキングのリスクを軽減するのに役立ちます。

5.2.2. グループ相対ポリシー最適化

DeepSeek-V2（DeepSeek-AI、2024c)と同様に、我々はグループ相対ポリシー最適化（GRPO）（Shao et al.、2024)を採用し、通常は同じ
政策モデルとしてサイズを考慮せず、代わりにグループスコアからベースラインを推定します。具体的には、
各質問において、GRPOは古い政策モデルから出力のグループ{ 1, 2, ・・・ }をサンプリングする。
そして、次の目的を最大化することでポリシー モデルを最適化します。

J ( ) = E[( ), { } =1 ~ (|)]

$$\frac{1}{}\sum_{=1} \text{分} \quad \frac{(\,|\,)}{(\,|\,)} \quad \text{、クリップ} \quad \frac{(\,|\,)}{(\,|\,)}, 1 - \ , 1 + \quad \text{ーダ} \quad , \tag{26}$$

$$\text{デ} \quad = \frac{(\,|\,)}{(\,|\,)} - \text{ログ} \frac{(\,|\,)}{1, (\,|\,)} - \tag{27}$$

ここで、およびはハイパーパラメータ、は参照モデル、および導出された利点である。
各グループ内の出力に対応する報酬{ 1, 2, . . . , }から:

$$= \frac{- \text{平均}(\{ 1, 2, \cdots, \})}{\text{std}(\{ 1, 2, \cdots, \})} 。 \tag{28}$$

コーディング、数学、ライティング、ロールプレイングなど、さまざまな分野からのプロンプトを取り入れています。
RLプロセス中に、モデルをより正確に調整するだけでなく、質問への回答も行います。
人間の好みに非常に近いだけでなく、特にベンチマークでのパフォーマンスも向上します。
利用可能な SFT データが限られているシナリオ。

5.3. 評価

5.3.1. 評価設定

評価ベンチマーク。ベースモデルのテストに使用したベンチマークとは別に、
さらに、IFEval（Zhou et al., 2023）、FRAMES（Krishna et al., 2024)、LongBench v2 (Bai et al.、2024)、GPQA (Rein et al.、2023)、SimpleQA (OpenAI、2024c)、C- SimpleQA (He et al.、2024)、SWE-Bench Verified (OpenAI、2024d)、Aider LiveCodeBench (Jain[1]、2024年8月から2024年11月までの出題）、コードフォース高校数学オリンピック（CNMO 2024)3 、アメ[2]、中国国民リカ招待数学
2024 年試験 (AIME 2024) (MAA、2024)。

比較ベースライン。DeepSeek -V2-0506、DeepSeek-V2.5-0905、Qwen2.5 72B Instructなど、いくつかの強力なベースラインに対してチャットモデルの包括的な評価を実施しました。
LLaMA-3.1 405B Instruct、Claude-Sonnet-3.5-1022、GPT-4o-0513。DeepSeek-V2の場合
モデルシリーズでは、比較のために最も代表的なバリアントを選択します。クローズドソースの場合
モデルごとに、それぞれの API を通じて評価が実行されます。

詳細な評価構成。MMLU 、DROPなどの標準ベンチマークの場合、
GPQA および SimpleQA では、 simple-evals フレームワーク4 から評価プロンプトを採用しています。

1https://aider.chat
2https://codeforces.com
3https://www.cms.org.cn/Home/comp/comp/cid/12.html
4https://github.com/openai/simple-evals

ゼロショット設定の MMLU-Redux には、Zero-Eval プロンプト形式 (Lin、2024) を使用します。
その他のデータセットについては、データセット作成者が提供するデフォルトのプロンプトを使用して、元の評価プロトコルに従います。コードと数学のベンチマークについては、HumanEval-Mulデータセット
8つの主流プログラミング言語（Python、Java、Cpp、C#、JavaScript、TypeScript、
合計でPHP、Bash）です。CoTと非CoTの手法を使用してモデルのパフォーマンスを評価します
LiveCodeBenchでは、2024年8月から2024年11月までのデータが収集されます。
Codeforcesデータセットは競合他社の割合を使用して測定されます。SWE-Benchで検証されています
エージェントレスフレームワーク（Xia et al., 2024）を使用して評価しました。評価には「diff」形式を使用します。
Aider関連のベンチマーク。数学の評価については、AIMEとCNMO 2024が
温度0.7で評価され、結果は16回の実行で平均化されますが、MATH-500
貪欲なデコードを採用しています。すべてのモデルで、各トークンの最大8192トークンを出力できます。
ベンチマーク。

	ベンチマーク（メトリック）	ディープシーク V2-0506	ディープシーク V2.5-0905	Qwen2.5 72B-Inst.	LLaMA-3.1 405B-Inst.	クロード-3.5- ソネット-1022	GPT-4o 0513	ディープシーク V3
	建築	文部科学省	MoE 濃密	密集		-	-	文部科学省
	# アクティブ化されたパラメータ	21B	21B	72B	405B	-	-	37B
	# 合計パラメータ	236B	236B	72B	405B	-	-	671B
英語	MMLU （エムエル）	78.2	80.6	85.3	88.6	88.3	87.2	88.5
	MMLU-Redux (EM)	77.9	80.3	85.6	86.2	88.9	88.0	89.1
	MMLU-プロ（EM）	58.5	66.2	71.6	73.3	78.0	72.6	75.9
	DROP （3ショットF1）	83.0	87.8	76.7	88.7	88.3	83.7	91.6
	IF-Eval (プロンプト厳密)	57.7	80.6	84.1	86.0	86.5	84.3	86.1
	GPQA-ダイヤモンド（合格@1）	35.3	41.3	49.0	51.1	65.0	49.9	59.1
	SimpleQA (正解)	9.0	10.2	9.1	17.1	28.4	38.2	24.9
	フレーム（付属品）	66.9	65.4	69.8	70.0	72.5	80.5	73.3
	LongBench v2 (Acc.)	31.6	35.4	39.4	36.1	41.0	48.1	48.7
コード	人間評価-Mul (Pass@1)	69.3	77.4	77.3	77.2	81.7	80.5	82.6
	LiveCodeBench (Pass@1-COT)	18.8	29.2	31.1	28.4	36.3	33.4	40.5
	LiveCodeBench (合格@1)	20.3	28.4	28.7	30.1	32.8	34.2	37.6
	コードフォース（パーセンタイル）	17.5	35.6	24.8	25.3	20.3	23.6	51.6
	SWE 検証済み(解決済み)	-	22.6	23.8	24.5	50.8	38.8	42.0
	補助編集(Acc.)	60.3	71.6	65.4	63.9	84.2	72.9	79.7
	Aider-Polyglot (Acc.)	-	18.2	7.6	5.8	45.3	16.0	49.6
数学	AIME 2024 (合格@1)	4.6	16.7	23.3	23.3	16.0	9.3	39.2
	数学500 （EM）	56.3	74.7	80.0	73.8	78.3	74.6	90.2
	CNMO 2024 (合格@1)	2.8	10.8	15.9	6.8	13.1	10.8	43.2
中国語	CLUEWSC (EM)	89.9	90.4	91.4	84.7	85.4	87.9	90.9
	C評価（EM）	78.6	79.5	86.1	61.5	76.7	76.0	86.5
	C-SimpleQA (正解)	48.5	54.1	48.4	50.4	51.3	59.3	64.8

表6 | DeepSeek-V3と他の代表的なチャットモデルの比較。すべてのモデル
出力長を8Kに制限する設定で評価されます。
1000個未満のサンプルを、温度設定を変えながら複数回テストして、
堅牢な最終結果。DeepSeek-V3は最高のパフォーマンスを誇るオープンソースモデルであり、
最先端のクローズドソースモデルに対して競争力のあるパフォーマンスを発揮します。

5.3.2. 標準評価

表6は評価結果を示しており、DeepSeek-V3が最も優れたオープンソースモデルであることを示しています。さらに、最先端のクローズドソースモデルに対しても競争力があります。
GPT-4o や Claude-3.5-Sonnet などのモデル。

英語ベンチマーク。MMLUは、さまざまな知識ドメインとタスクにわたる大規模言語モデルのパフォーマンスを評価するために設計された、広く認知されたベンチマークです。DeepSeek-V3 は、LLaMA-3.1-405B、GPT-4o、Claude-Sonnet 3.5 などのトップクラスのモデルと同等の競争力のあるパフォーマンスを示し、Qwen2.5 72B を大幅に上回ります。

さらに、DeepSeek-V3 は、より難しい教育知識ベンチマークである MMLU-Pro でも優れており、Claude-Sonnet 3.5 に僅差で追随しています。ラベルを修正した MMLU の改良版である MMLU-Redux では、 DeepSeek-V3 は他の製品よりも優れています。さらに、博士レベルの評価テストベッドである GPQA-Diamond でも、 DeepSeek-V3 は優れた結果を達成し、Claude 3.5 Sonnet に次ぐランクとなり、他のすべての競合製品を大幅に上回っています。

DROP、LongBench v2、FRAMES などのロングコンテキスト理解ベンチマークでは、 DeepSeek-V3 は引き続きトップクラスのモデルとしての地位を確立しています。DROPの 3 ショット設定で 91.6 という素晴らしい F1 スコアを達成し、このカテゴリの他のすべてのモデルを上回っています。

10 万トークンのコンテキストを超える質問応答を必要とするベンチマークである FRAMES では、DeepSeek - V3 は GPT-4o にわずかに遅れをとりながらも、他のすべてのモデルを大幅に上回っています。これは、非常に長いコンテキストのタスクを処理する DeepSeek-V3 の強力な能力を示しています。
DeepSeek-V3 のロングコンテキスト機能は、 DeepSeek V3 の発売のわずか数週間前にリリースされたデータセットである LongBench v2 でのクラス最高のパフォーマンスによってさらに実証されています。事実知識ベンチマークである SimpleQA では、DeepSeek-V3 は主に設計の焦点とリソース割り当てにより、GPT-4o および Claude-Sonnet に遅れをとっています。DeepSeek-V3 は、中国語の知識を学習するためにより多くのトレーニング トークンを割り当てるため、 C-SimpleQA で優れたパフォーマンスを発揮します。命令に従うベンチマークでは、DeepSeek-V3 は前身の DeepSeek-V2 シリーズを大幅に上回り、ユーザー定義のフォーマット制約を理解して遵守する能力が向上していることが強調されています。

コードと数学のベンチマーク。コーディングは、LLM にとって困難で実用的なタスクであり、 SWE-Bench-Verified や Aider などのエンジニアリングに重点を置いたタスク、およびHumanEval や LiveCodeBench などのアルゴリズム タスクが含まれます。エンジニアリング タスクでは、DeepSeek-V3 はClaude-Sonnet-3.5-1022 に遅れをとっていますが、オープン ソース モデルを大幅に上回っています。オープン ソースのDeepSeek-V3 は、コーディング関連のエンジニアリング タスクの進歩を促進することが期待されています。DeepSeek -V3 は、その強力な機能へのアクセスを提供することで、ソフトウェア エンジニアリングやアルゴリズム開発などの分野で革新と改善を促進し、開発者や研究者がコーディング タスクでオープン ソース モデルが達成できる限界を押し広げることを可能にします。アルゴリズム タスクでは、DeepSeek-V3 は優れたパフォーマンスを示し、 HumanEval-Mul や LiveCodeBench などのベンチマークですべてのベースラインを上回っています。この成功は、アルゴリズムに重点を置いたタスクにおけるコード生成と問題解決の能力を効果的に強化する、高度な知識蒸留技術によるものです。

数学ベンチマークでは、DeepSeek-V3 は優れたパフォーマンスを発揮し、ベースラインを大幅に上回り、非 o1 ライクなモデルに新たな最先端技術をもたらしました。具体的には、 AIME、MATH-500、CNMO 2024 において、DeepSeek-V3 は 2 番目に優れたモデルである Qwen2.5 72B を絶対スコアで約 10% 上回り、このような難しいベンチマークでは大きな差となります。この驚くべき機能は、非 o1 ライクなモデルに非常に有益であることが証明されている DeepSeek-R1 の蒸留技術の有効性を強調しています。

中国語ベンチマーク。QwenとDeepSeekは、中国語と英語の両方を強力にサポートする2つの代表的なモデルシリーズです。事実に基づくベンチマークである中国語SimpleQAでは、DeepSeek -V3はQwen2.5-72Bを16.4ポイント上回っていますが、Qwen2.5は18Tトークンを含むより大きなコーパスでトレーニングされており、これはDeepSeek-V3の14.8Tトークンより20%多いものです。

モデル	アリーナハード	AlpacaEval 2.0
DeepSeek-V2.5-0905	76.2	50.5
Qwen2.5-72B-Instruct	81.2	49.1
LLaMA-3.1 405B	69.3	40.5
GPT-4o-0513 クロ	80.4	51.1
ード-ソネット-3.5-1022 DeepSeek-	85.2	52.0
V3	85.5	70.0

表 7 |英語の自由形式の会話の評価。AlpacaEval 2.0 では、長さを制御した勝率を指標として使用します。

事前にトレーニング済み。

中国の教育知識評価の代表的なベンチマークであるC-Evalについて、
CLUEWSC（中国ウィノグラードスキーマチャレンジ）、DeepSeek-V3、Qwen2.5-72B展示
同様のパフォーマンスレベルは、両方のモデルが困難な状況に十分に最適化されていることを示しています。
中国語の推論と教育タスク。

5.3.3. 自由記述式評価

標準的なベンチマークに加えて、オープンエンド生成モデルも評価します。
タスクはLLMを審査員として用いて実施され、結果は表7に示されている。具体的には、
AlpacaEval 2.0（Dubois et al., 2024)とArena-Hard（Li et al.,
2024a)では、GPT-4-Turbo-1106をペアワイズ比較の判定に利用しています。Arena-Hardでは、
DeepSeek-V3はベースラインGPT-4-0314に対して86%を超える優れた勝率を達成しました。
クロード•ソネット3.5-1022のようなトップクラスのモデルと同等の性能を発揮します。
DeepSeek-V3の強力な機能、特に複雑なプロンプトの処理において、
コーディングとデバッグのタスク。さらに、DeepSeek-V3は画期的なマイルストーンを達成しました。
アリーナハードベンチマークで85%を超えた最初のオープンソースモデルとして。この成果は
オープンソースモデルとクローズドソースモデル間のパフォーマンスギャップを大幅に埋めます。
困難な領域でオープンソース モデルが達成できることの新たな基準を設定します。

同様に、DeepSeek-V3はAlpacaEval 2.0で優れたパフォーマンスを発揮し、クローズドソースとオープンソースの両方のモデルを上回っています。これは、
文章作成タスクと簡単な質問回答シナリオの処理。特に、
DeepSeek-V2.5-0905は20%という大幅な差で、
単純なタスクに取り組み、その進歩の有効性を示します。

5.3.4. 生成報酬モデルとしての DeepSeek-V3

DeepSeek-V3の判断能力を最先端のモデルであるGPT-4oと比較します。
表8は、RewardBench（Lambert
ら、2024年）。DeepSeek-V3はGPT-4o-0806の最高バージョンと同等のパフォーマンスを達成した。
クロード-3.5-ソネット-1022は他のバージョンを上回っています。さらに、判断能力
DeepSeek-V3の性能は投票技術によっても向上します。そのため、DeepSeek- V3と投票を組み合わせて自由回答形式の質問に対する自己フィードバックを提供し、
アライメント プロセスの有効性と堅牢性。

モデル	チャット	チャットハード	安全性推論	平均	
GPT-4o-0513	96.6	70.4	86.7	84.9	84.7
GPT-4o-0806	96.1	76.1	88.1	86.6	86.7
GPT-4o-1120	95.8	71.3	86.2	85.2	84.6
クロード-3.5-ソネット-0620 96.4	クロー	74.0	81.6	84.7	84.2
ド-3.5-ソネット-1022 96.4		79.7	91.1	87.6	88.7
DeepSeek-V3 96.9 DeepSeek-V3		79.8	87.0	84.3	87.0
(maj@6) 96.9		82.6	89.5	89.2	89.6

表 8 | RewardBench における GPT-4o、Claude-3.5-sonnet、DeepSeek-V3 のパフォーマンス。

モデル	ライブコードベンチ-CoT MATH-500			
	パス@1	長さ	パス@1	長さ
DeepSeek-V2.5 ベースライン	31.1	718	74.6	769
DeepSeek-V2.5 +R1 蒸留	37.4	783	83.2	1510

表 9 | DeepSeek-R1 からの蒸留の貢献。Live- CodeBench と MATH-500 の評価設定は表 6 と同じです。

5.4. 議論

5.4.1. DeepSeek-R1からの抽出

DeepSeek-V2.5をベースにDeepSeek-R1からの蒸留の寄与をアブレーションします。
ベースラインは短いCoTデータでトレーニングされているが、競合他社は専門家によって生成されたデータを使用している。
上記のチェックポイント。

表9は、蒸留データの有効性を示しており、 LiveCodeBenchとMATH-500ベンチマークの両方で大幅な改善が見られます。私たちの実験では、興味深いトレードオフが明らかになりました。蒸留によりパフォーマンスが向上しますが、
平均応答長。モデルの精度と計算量のバランスを保つために
効率性を高めるために、DeepSeek-V3 の蒸留に最適な設定を慎重に選択しました。

私たちの研究は、推論モデルからの知識の蒸留が、トレーニング後の最適化の有望な方向性を示していることを示唆しています。私たちの現在の研究は、データの蒸留に焦点を当てていますが、
数学やコーディングの分野から、このアプローチはより幅広い応用の可能性を示している
さまざまなタスク領域にわたって、これらの特定の領域で実証された有効性は、
長いCoT蒸留は、複雑な推論を必要とする他の認知タスクにおけるモデルのパフォーマンスを向上させるのに役立つ可能性がある。このアプローチをさまざまな分野でさらに調査する。
ドメインは、将来の研究にとって重要な方向性であり続けます。

5.4.2. 自己報酬

報酬はRLにおいて、最適化プロセスを導く重要な役割を果たします。コーディングや数学のシナリオなど、外部ツールによる検証が簡単な領域では、RLは
非常に効果的です。しかし、より一般的なシナリオでは、フィードバックを構築することは
ハードコーディングによるメカニズムは現実的ではない。DeepSeek-V3の開発中、
これらのより広い文脈では、私たちは憲法AIアプローチ（Bai et al., 2022)を採用し、
DeepSeek-V3自体の投票評価結果をフィードバックソースとして利用します。この方法は

顕著なアライメント効果を生み出し、主観評価におけるDeepSeek-V3 のパフォーマンスを大幅に向上させました。追加の構成入力を統合することで、DeepSeek-V3 は構成方向に向けて最適化できます。補足情報と LLM をフィードバック ソースとして組み合わせるこのパラダイムは、非常に重要であると私たちは考えています。LLMは、さまざまなシナリオからの非構造化情報を報酬に変換できる多目的プロセッサとして機能し、最終的に LLM の自己改善を促進します。自己報酬を超えて、一般的なシナリオでモデル機能を一貫して向上させるための他の一般的でスケーラブルな報酬方法を発見することにも取り組んでいます。

5.4.3. マルチトークン予測評価

DeepSeek-V3 は、次の単一のトークンを予測する代わりに、 MTP 技術を使用して次の 2 つのトークンを予測します。投機的デコードのフレームワーク (Leviathan ら、 2023 年、Xia ら、2023 年) と組み合わせることで、モデルのデコード速度を大幅に加速できます。当然、追加で予測されたトークンの受け入れ率に関する疑問が生じます。当社の評価によると、2 番目のトークン予測の受け入れ率は、さまざまな世代トピックにわたって 85% から 90% の範囲であり、一貫した信頼性を示しています。この高い受け入れ率により、DeepSeek-V3 はデコード速度を大幅に向上させ、1.8 倍のTPS (トークン/秒) を実現できます。

6. 結論、限界、今後の方向性

本稿では、14.8T トークンでトレーニングされた、合計 671B のパラメーターと 37B のアクティブ化パラメーターを備えた大規模な MoE 言語モデルである DeepSeek-V3 を紹介します。MLA およびDeepSeekMoE アーキテクチャに加えて、負荷分散のための補助損失のない戦略も開拓し、より強力なパフォーマンスのためにマルチトークン予測トレーニング目標を設定します。FP8 トレーニングと綿密なエンジニアリング最適化のサポートにより、DeepSeek-V3 のトレーニングは費用対効果に優れています。トレーニング後も、 DeepSeek-R1 シリーズのモデルから推論機能を抽出することに成功しています。包括的な評価により、DeepSeek-V3 は現在入手可能な最も強力なオープンソース モデルとして浮上し、 GPT-4o や Claude-3.5-Sonnet などの主要なクローズド ソース モデルに匹敵するパフォーマンスを実現していることが実証されています。強力なパフォーマンスにもかかわらず、経済的なトレーニング コストも維持されています。事前トレーニング、コンテキスト長の拡張、事後トレーニングを含む完全なトレーニングには、わずか 278.8 万時間の H800 GPU 時間しか必要ありません。

DeepSeek-V3 は優れたパフォーマンスとコスト効率に優れていると認識していますが、特に導入に関していくつかの制限があることも認識しています。まず、効率的な推論を保証するために、DeepSeek-V3 の推奨導入単位は比較的大きく、小規模チームにとっては負担になる可能性があります。次に、DeepSeek- V3 の導入戦略により、エンドツーエンドの生成速度は DeepSeek-V2 の 2 倍以上になりましたが、さらに強化できる可能性はまだ残っています。幸いなことに、これらの制限は、より高度なハードウェアの開発によって自然に解決されると予想されます。

DeepSeekは、オープンソースモデルを長期的視点で追求する路線を一貫して貫き、 AGI（汎用人工知能）という究極の目標に着実に近づくことを目指しています。今後は、以下の方向性で戦略的に研究投資を行っていく予定です。

- 私たちは、モデルアーキテクチャを継続的に研究し、改良し、トレーニングと推論の効率をさらに向上させ、無限のコンテキスト長を効率的にサポートできるように努めます。さらに、Transformer のアーキテクチャ上の制限を打ち破り、モデリング機能の限界を押し広げていきます。

- トレーニングデータの量と質を継続的に改善し、追加のトレーニング信号ソースの組み込みを検討して、より包括的な範囲の次元でデータのスケーリングを推進します。 •モデルの深い思考能力を継続的に調査して繰り返し、推論の長さと深さを拡大することで、モデルの知性と問題
解決能力を強化することを目指します。 •研究中に固定されたベンチマークセットを最適化する傾向を防ぐため、より包括的で多次元的なモデル評価方法を検討します。これにより、モデル機能について誤解を招く印象を与え、基礎的な評価に影響を与える可能性が
あります。

評価。

参考文献

AI@Meta。Llama 3 モデル カード、2024a。URL https://github.com/meta-llama/llama3/bl ob/main/MODEL_CARD.md。

AI@Meta。Llama 3.1 モデル カード、2024b。URL https://github.com/meta-llama/llama-m odels/blob/main/models/llama3_1/MODEL_CARD.md。

アントロピック。クロード 3.5 ソネット、2024 年。URL https://www.anthropic.com/news/claude-3-5-ソネット。

J. オースティン、A. オデナ、M. ナイ、M. ボスマ、H. ミカレフスキー、D. ドーハン、E. ジャン、C. カイ、M. テリー、 Q. ル、他。大規模な言語モデルを使用したプログラム合成。 arXiv プレプリント arXiv:2108.07732、2021 。

Y. Bai、S. Kadavath、S. Kundu、A. Askell、J. Kernion、A. Jones、A. Chen、A. Goldie、A. Mirhoseini、 C. McKinnon、他。 AI の憲法: AI フィードバックによる無害性。 arXiv プレプリント arXiv:2212.08073、2022。

Y. Bai、S. Tu、J. Zhang、H. Peng、X. Wang、X. Lv、S. Cao、J. Xu、L. Hou、Y. Dong、J. Tang、およびJ. Li。LongBench v2: 現実的なロングコンテキストマルチタスクに関するより深い理解と推論に向けて。arXiv プレプリント arXiv:2412.15204、2024。

M. Bauer、S. Treichler、および A. Aiken。Singe: GPU で高パフォーマンスを実現するためのワープ特殊化の活用。第 19 回 ACM SIGPLAN シンポジウム「並列プログラミングの原理と実践」の議事録、 PPoPP '14、119 ～ 130 ページ、ニューヨーク、ニューヨーク州、米国、2014 年。Association for Computing Machinery。ISBN 9781450326568。doi: 10.1145/2555243.2555258。URL https://doi.org/10.1145/2555243.2555258。

Y. Bisk、R. Zellers、RL Bras、J. Gao、Y. Choi。PIQA:自然言語における物理的常識の推論。第 34 回 AAAI 人工知能会議、AAAI 2020、第 32 回革新的人工知能応用会議、IAAI 2020、第 10 回 AAAI 人工知能の教育的進歩に関するシンポジウム、EAAI 2020、ニューヨーク、ニューヨーク、米国、2020 年 2 月 7 ～ 12 日、 7432 ～ 7439 ページ。AAAI Press、2020 年。doi: 10.1609/aaai.v34i05.6239。URL https://doi.org/10.1609/aaai.v34i05.6239。

M. チェン、J. トゥーレク、H. ジュン、Q. ユアン、HP デ オリベイラ ピント、J. カプラン、H. エドワーズ、Y. バーダ、 N. ジョセフ、G. ブロックマン、A. レイ、R. プリ、G. クルーガー、M. ペトロフ、H. クラーフ、G. サストリー、P. ミシュキン、 B. チャン、S. グレイ、N. ライダー、M. パブロフ、A. パワー、L. カイザー、M. ババリアン、C. ウィンター、P. ティレット、FP サッチ、D. カミングス、M. プラパート、F. チャンツィス、E. バーンズ、A. ハーバート フォス、WH ガス、 A. ニコル、A. ペインオ、N. テザック、J. タン、I. バブシュキン、S. バラジ、S. ジェイン、W. サンダース、C. ヘッセ、

AN Carr、J. Leike、J. Achiam、V. Misra、E. Morikawa、A. Radford、M. Knight、M. Brundage、 M. Murati、K. Mayer、P. Welinder、B. McGrew、D. Amodei、S. McCandlish、I. Sutskever、 W. Zaremba。コードでトレーニングされた大規模言語モデルの評価。CoRR 、 abs/2107.03374、2021。URL https://arxiv.org/abs/2107.03374.

P. Clark、I. Cowhey、O. Etzioni、T. Khot、A. Sabharwal、C. Schoenick、および O. Tafjord。質問応答を解決したと思いますか? arc、AI2 推論チャレンジに挑戦してください。CoRR 、 abs/1803.05457、2018。URL http://arxiv.org/abs/1803.05457 。

K. Cobbe、V. Kosaraju、M. Bavarian、M. Chen、H. Jun、L. Kaiser、M. Plappert、J. Tworek、J. Hilton、R. Nakano、他「数学の文章問題を解くための検証者のトレーニング」 arXiv プレプリント arXiv:2110.14168、2021 年。

Y. Cui、T. Liu、W. Che、L. Xiao、Z. Chen、W. Ma、S. Wang、および G. Hu。中国語機械読解のためのスパン抽出データセット。K. Inui、J. Jiang、V. Ng、および X. Wan 編、 Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing および第 9 回国際自然言語処理合同会議 (EMNLP-IJCNLP)、ページ 5883–5889、香港、中国、2019 年 11 月。計算言語学協会。doi: 10.18653/v1/D19-1600。URL https://aclanthology.org/D19-1600 。

D. Dai、C. Deng、C. Zhao、RX Xu、H. Gao、D. Chen、J. Li、W. Zeng、X. Yu、Y. Wu、Z. Xie、YK Li、P. Huang、F. Luo、C. Ruan、Z. Sui、W. Liang。Deepseekmoe：専門家混合言語モデルにおける究極の専門家特化に向けて。CoRR 、 abs/2401.06066、2024。URL https://doi.org/10.48550/arXiv.2401.06066。

DeepSeek-AI. Deepseek-coder-v2: コードインテリジェンスにおけるクローズドソースモデルの障壁を打ち破る。CoRR 、 abs/2406.11931、2024a。URL https://doi.org/10.48550/arXiv.2406.11 931。

DeepSeek-AI。Deepseek LLM: 長期主義によるオープンソース言語モデルのスケーリング。CoRR 、 abs/2401.02954、2024b。URL https://doi.org/10.48550/arXiv.2401.02954。

DeepSeek-AI。Deepseek-v2: 強力で経済的、効率的な専門家混合言語モデル。CoRR 、 abs/2405.04434、2024c。URL https://doi.org/10.48550/arXiv.2405.04434。

T. Dettmers、M. Lewis、Y. Belkada、および L. Zettlemoyer。Gpt3. int8 (): 大規模なトランスフォーマー用の 8 ビット行列乗算。ニューラル情報処理システムの進歩、 35:30318– 30332、2022 年。

H. Ding、Z. Wang、G. Paolini、V. Kumar、A. Deoras、D. Roth、およびS. Soatto。切り捨てが少ないほど言語モデルが改善されます。arXivプレプリントarXiv：2404.10830、2024。

D. Dua、Y. Wang、P. Dasigi、G. Stanovsky、S. Singh、および M. Gardner。「DROP:段落全体にわたる離散推論を必要とする読解ベンチマーク」。J. Burstein、C. Doran、および T. Solorio 編著、 Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies、NAACL-HLT 2019、ミネアポリス、ミネソタ州、米国、2019 年 6 月 2 ～ 7 日、第 1 巻 (長文および短文論文)、 2368 ～ 2378 ページ。Association for Computational Linguistics、2019 年。doi: 10.18653/V1/N19-1246。 URL https://doi.org/10.18653/v1/n19-1246.

Y. デュボア、B. ガランボシ、P. リャン、TB 橋本。長さ制御されたアルパカエバル: シンプルな自動評価者のバイアスを解消する方法。arXiv プレプリント arXiv:2404.04475, 2024。

W. Fedus、B. Zoph、N. Shazeer。「スイッチトランスフォーマー :シンプルで効率的なスパース性を備えた兆パラメータモデルへのスケーリング」 CoRR、 abs/2101.03961、2021年。URL https://arxiv.org/abs/2101.03961 。

M. Fishman、B. Chmiel、R. Banner、および D. Soudry。FP8 トレーニングを 1 兆トークンの LLM にスケーリングします。arXiv プレプリント arXiv:2409.12517、2024。

E. Frantar、S. Ashkboos、T. Hoefler、D. Alistarh。Gptq: トレーニング後の正確な量子化生成的事前学習済みトランスフォーマー向け。arXiv プレプリント arXiv:2210.17323, 2022。

L. Gao、S. Biderman、S. Black、L. Golding、T. Hoppe、C. Foster、J. Phang、H. He、A. Thite、 N. Nabeshima、他「The Pile: 言語モデリングのための多様なテキストの 800GB データセット」 arXiv プレプリント arXiv:2101.00027、2020 年。

AP Gema、JOJ Leang、G. Hon、A. Devoto、ACM Mancino、R. Saxena、X. He、Y. Zhao、X. Du、MRG Madani、C. Barale、R. McHardy、J. Harris、J. Kaddour、E. van Krieken、およびP. Minervini。 mmluはもう終わりですか？ CoRR、 abs/2406.04127、2024。URL https://doi.or g/10.48550/arXiv.2406.04127。

F. Gloeckle、BY Idrissi、B. Rozière、D. Lopez-Paz、および G. Synnaeve。「マルチトークン予測による大規模言語モデルの高速化と改善」。第41 回国際機械学習会議、ICML 2024、オーストリア、ウィーン、2024 年 7 月 21 ～ 27 日。OpenReview.net 、2024 年。URL https://openreview.net/forum?id=pEWAcejiU2。

Google。次世代モデル: Gemini 1.5、2024年。URL https://blog.google/technology/ai/google-gemini-次世代モデル-2024 年 2 月。

RL Graham、D. Bureddy、P. Lui、H. Rosenstock、G. Shainer、G. Bloch、D. Goldenerg、M. Dubman、 S. Kotchubievsky、V. Koushnir、他「スケーラブルな階層型集約プロトコル (SHArP):効率的なデータ削減のためのハードウェア アーキテクチャ」。2016年第 1 回 HPC における通信最適化に関する国際ワークショップ (COMHPC)、1 ～ 10 ページ。IEEE、2016 年。

A. Gu、B. Rozière、H. Leather、A. Solar-Lezama、G. Synnaeve、SI Wang。Cruxeval:コードの推論、理解、実行のベンチマーク、2024 年。

D. Guo、Q. Zhu、D. Yang、Z. Xie、K. Dong、W. Zhang、G. Chen、X. Bi、Y. Wu、YK Li、F. Luo、 Y. Xiong、W. Liang。Deepseek-coder: 大規模言語モデルとプログラミングが出会うとき - コードインテリジェンスの台頭。CoRR 、 abs/2401.14196、2024。URL https://doi.org/10.485 50/arXiv.2401.14196。

A. ハーラップ、D. ナラヤナン、A. パニシャイー、V. セシャドリ、N. デバヌール、G. ガンガー、および P. ギボンズ。Pipedream: 高速かつ効率的なパイプライン並列 DNN トレーニング、2018 年。URL https://arxiv.org/abs/1806.03377 。

B. He、L. Noci、D. Paliotta、I. Schlag、および T. Hofmann。「Transformer トレーニングにおける外れ値の特徴の理解と最小化」。第 38 回神経情報処理システム年次会議。

Y. He、S. Li、J. Liu、Y. Tan、W. Wang、H. Huang、X. Bu、H. Guo、C. Hu、B. Zheng、他「Chi-nese simpleqa: 大規模言語モデルのための中国語事実性評価」 arXiv プレプリント arXiv:2411.07140、2024。

D. Hendrycks、C. Burns、S. Basart、A. Zou、M. Mazeika、D. Song、J. Steinhardt。大規模マルチタスク言語理解の測定。arXiv プレプリント arXiv:2009.03300、2020 年。

D. Hendrycks、C. Burns、S. Kadavath、A. Arora、S. Basart、E. Tang、D. Song、J. Steinhardt。数学データセットによる数学問題解決の測定。arXivプレプリント arXiv:2103.03874、2021年。

Y. Huang、Y. Bai、Z. Zhu、J. Zhang、J. Zhang、T. Su、J. Liu、C. Lv、Y. Zhang、J. Lei、他「C-Eval:基礎モデルのためのマルチレベル、マルチ分野の中国語評価スイート」 arXiv プレプリント arXiv:2305.08322、2023 年。

N. Jain、K. Han、A. Gu、W. Li、F. Yan、T. Zhang、S. Wang、A. Solar-Lezama、K. Sen、I. Stoica。Livecodebench: コードの大規模言語モデルの全体的かつ汚染のない評価。CoRR、abs/2403.07974、2024。URL https://doi.org/10.48550/arXiv.2403.07974。

AQ Jiang、A. Sablayrolles、A. Mensch、C. Bamford、DS Chaplot、D. dl Casas、F. Bressand、 G. Lengyel、G. Lample、L. Saulnier、他。ミストラル7b。 arXiv プレプリント arXiv:2310.06825、2023。

M. Joshi、E. Choi、D. Weld、および L. Zettlemoyer。TriviaQA: 読解力のための大規模な遠隔教師付きチャレンジデータセット。R. Barzilay および M.-Y. Kan 編、 Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)、ページ 1601–1611、バンクーバー、カナダ、2017 年 7 月。Association for Computational Linguistics。doi: 10.18653/v1/P17-1147。URL https://aclanthology.org/P17-1147。

D. カラムカール、D. ムディゲレ、N. メレンプディ、D. ダス、K. バナジー、S. アヴァンチャ、DT ブートゥリ、 N. ジャマラマダカ、J. ファン、H. ユエン、他。深層学習トレーニングのための bfloat16 の研究。 arXiv プレプリント arXiv:1905.12322、2019。

S. クリシュナ、K. クリシュナ、A. モハナニー、S. シュワルツ、A. スタンブラー、S. ウパディヤイ、M. ファルキ。事実、フェッチ、推論 :検索強化型生成の統一評価。CoRR 、 abs/2409.12941、2024。doi :10.48550/ARXIV.2409.12941。URL https://doi.org/10.485 50/arXiv.2409.12941。

T. Kwiatkowski、J. Palomaki、O. Redfield、M. Collins、AP Parikh、C. Alberti、D. Epstein、I. Polosukhin、J. Devlin、K. Lee、K. Toutanova、L. Jones、M. Kelcey、M. Chang、AM Dai、J. Uszkoreit、Q. Le、および S. Petrov。自然な質問 :質問応答研究のベンチマーク。Trans . Assoc. Comput. Linguistics、 7:452–466、2019。doi :10.1162/tacl_a_00276。URL https://doi.org/10.1162/tacl_a_00276.

G. Lai、Q. Xie、H. Liu、Y. Yang、EH Hovy。RACE: 試験からの大規模な読解データセット。M. Palmer、R. Hwa、S. Riedel 編著、 Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing、EMNLP 2017、コペンハーゲン、デンマーク、2017 年 9 月 9 ～ 11 日、 785 ～ 794 ページ。Association for Computational Linguistics、2017 年。doi: 10.18653/V1/D17-1082。URL https://doi.org/10.18653/v1/d1 7-1082。

N. Lambert、V. Pyatkin、J. Morrison、L. Miranda、BY Lin、K. Chandu、N. Dziri、S. Kumar、 T. Zick、Y. Choi、他「Rewardbench: 言語モデリングのための報酬モデルの評価」 arXiv プレプリント arXiv:2403.13787、2024。

D. Lepikhin、H. Lee、Y. Xu、D. Chen、O. Firat、Y. Huang、M. Krikun、N. Shazeer、Z. Chen。Gshard: 条件付き計算と自動シャーディングによる巨大モデルのスケーリング。第 9 回国際学習表現会議、ICLR 2021。OpenReview.net 、2021 年。URL https://openreview.net/forum?id=qrwe7XHTmYb.

Y. Leviathan、M. Kalman、および Y. Matias。「投機的デコードによるトランスフォーマーからの高速推論」。国際機械学習会議、ICML 2023、2023 年 7 月 23 ～ 29 日、米国ハワイ州ホノルル、「機械学習研究の議事録」第 202 巻、19274 ～ 19286ページ。PMLR、2023 年。URL https://proceedings.mlr.press/v202/leviathan23a.html。

H. リー、Y. チャン、F. コト、Y. ヤン、H. チャオ、Y. ゴン、N. ドゥアン、および T. ボールドウィン。 CMMLU:中国語における大量のマルチタスク言語理解を測定します。 arXiv プレプリント arXiv:2306.09212、2023 。

S. Li および T. Hoefler。Chimera: 双方向パイプラインによる大規模ニューラル ネットワークの効率的なトレーニング。高性能コンピューティング、ネットワーキング、ストレージ、分析に関する国際会議SC '21 の議事録、1 ～ 14 ページ。ACM、2021 年 11 月。doi: 10.1145/345881 7.3476145。URL http://dx.doi.org/10.1145/3458817.3476145。

T. Li、W.-L. Chiang、E. Frick、L. Dunlap、T. Wu、B. Zhu、JE Gonzalez、I. Stoica。クラウドソーシングされたデータから高品質のベンチマークへ: Arena-hard およびベンチビルダー パイプライン。arXivプレプリント arXiv:2406.11939、2024a。

W. Li、F. Qi、M. Sun、X. Yi、J. Zhang。Ccpm: 中国古典詩マッチングデータセット、2021年。

Y. Li、F. Wei、C. Zhang、および H. Zhang。EAGLE: 投機的サンプリングには特徴の不確実性の再考が必要。第 41 回国際機械学習会議、ICML 2024、オーストリア、ウィーン、2024 年 7 月 21 ～ 27 日。OpenReview.net、2024b。URL https://openreview.net /forum?id=1NdN7eXyb4。

リン著。ZeroEval: 言語モデルを評価するための統一フレームワーク、2024 年 7 月。URL https://github.com/WildEval/ZeroEval。

I. Loshchilov と F. Hutter。分離重み減衰正規化。arXivプレプリント arXiv:1711.05101、2017年。

S. Lundberg. プロンプトデザインの芸術: プロンプトの境界とトークンの治癒、2023年。URL https://towardsdatascience.com/the-art-of-prompt-design-prompt-bound aries-and-token-healing-3b2448b0be38。

Y. Luo、Z. Zhang、R. Wu、H. Liu、Y. Jin、K. Zheng、M. Wang、Z. He、G. Hu、L. Chen、他 Ascend ディープラーニング用のHiFloat8形式。arXivプレプリントarXiv:2409.16626, 2024。

MAA。アメリカ招待数学試験 - aime。アメリカ招待数学試験 - AIME 2024、2024年 2 月。URL https://maa.org/math-competitions/american-invitational-mathematics-examination-aime 。

P. Micikevicius、D. Stosic、N. Burgess、M. Cornea、P. Dubey、R. Grisenthwaite、S. Ha、A. Heinecke、P. Judd、J. Kamilu、他。ディープラーニング用のFP8フォーマット。 arXiv プレプリント arXiv:2209.05433、2022 。

ミストラル。より安く、より良く、より速く、より強く。AIの限界を押し広げ、誰でもアクセス可能、2024年。URL https://mistral.ai/news/mixtral-8x22b。

S. Narang、G. Diamos、E. Elsen、P. Micikevicius、J. Alben、D. Garcia、B. Ginsburg、M. Hous-ton、O. Kuchaiev、G. Venkatesh、他「混合精度トレーニング」。国際会議「学習表現」、2017 年。

B. Noune、P. Jones、D. Justus、D. Masters、C. Luschi。深層ニューラルネットワークのための8ビット数値形式ネットワーク。arXivプレプリントarXiv:2206.02915、2022。

NVIDIA。NVIDIA Magnum IO NVSH- MEM と GPUDirect Async を使用して HPC システムのネットワーク パフォーマンスを向上します。https ://developer.nvidia.com/blog/improving-net work-performance-of-hpc-systems-using-nvidia-magnum-io-nvshmem-and-g pudirect-async、 2022年。

NVIDIA。Blackwell アーキテクチャ。https ://www.nvidia.com/en-us/data-center/technologies/ブラックウェルアーキテクチャ/、 2024年a.

NVIDIA. TransformerEngine、2024b。URL https://github.com/NVIDIA/TransformerEngine .アクセス日: 2024-11-19。

オープンAI。こんにちは、GPT-4o、2024a。 URL https://openai.com/index/hello-gpt-4o/。

OpenAI。多言語大規模マルチタスク言語理解 (mmmlu)、2024b。URL https://huggingface.co/datasets/openai/MMMLU。

OpenAI. SimpleQA の紹介、2024c。URL https://openai.com/index/introducing -simpleqa/。

OpenAI。SWE-bench verified のご紹介。2024 年以降に人間が検証した SWE-bench のサブセットをリリースします。URL https://openai.com/index/introducing-swe-bench -verified/。

B. Peng、J. Quesnelle、H. Fan、E. Shippole。Yarn: 大規模なコンテキストウィンドウの効率的な拡張言語モデル。arXivプレプリントarXiv:2309.00071, 2023a。

H. Peng、K. Wu、Y. Wei、G. Zhao、Y. Yang、Z. Liu、Y. Xiong、Z. Yang、B. Ni、J. Hu、他。 FP8-LM: FP8大規模言語モデルのトレーニング。arXivプレプリントarXiv:2310.18313, 2023b。

P. Qi、X. Wan、G. Huang、M. Lin。ゼロバブルパイプライン並列処理。arXivプレプリント arXiv:2401.10241, 2023a.

P. Qi、X. Wan、G. Huang、M. Lin. ゼロバブルパイプライン並列処理、2023b。URL https: //arxiv.org/abs/2401.10241.

Qwen。Qwen技術レポート。arXivプレプリントarXiv:2309.16609、2023。

Qwen。Qwen1.5、2024a の紹介。URL https://qwenlm.github.io/blog/qwen1.5。

Qwen。Qwen2.5: 基礎モデルのパーティ、2024b。URL https://qwenlm.github.io/blog/qwen2.5。

S. Rajbhandari、J. Rasley、O. Ruwase、および Y. He. Zero: 1 兆パラメータ モデルのトレーニングに向けたメモリ最適化。SC20 : 高性能コンピューティング、ネットワーキング、ストレージ、および分析に関する国際会議、1 ～ 16 ページ。IEEE、2020 年。

D. Rein、BL Hou、AC Stickland、J. Petty、RY Pang、J. Dirani、J. Michael、および SR Bowman。GPQA: 大学院レベルの Google 対応 Q&A ベンチマーク。arXivプレプリント arXiv:2311.12022、2023 。

BD Rouhani、R. Zhao、A. More、M. Hall、A. Khodamoradi、S. Deng、D. Choudhary、M. Cornea、 E. Dellinger、K. Denolf、他「ディープラーニングのためのマイクロスケーリングデータ形式」 arXiv プレプリント arXiv:2310.10537, 2023a。

BD Rouhani、R. Zhao、A. More、M. Hall、A. Khodamoradi、S. Deng、D. Choudhary、M. Cornea、 E. Dellinger、K. Denolf、他「ディープラーニングのためのマイクロスケーリングデータ形式」 arXiv プレプリント arXiv:2310.10537, 2023b。

K. 坂口、RL Bras、C. Bhagavatula、Y. Choi。 Winogrande: 敵対的なウィノグラード 2019 年の大規模なスキーマ チャレンジ。

Z. Shao、P. Wang、Q. Zhu、R. Xu、J. Song、M. Zhang、Y. Li、Y. Wu、および D. Guo。Deepseekmath: オープン言語モデルにおける数学的推論の限界を押し広げる。arXivプレプリント arXiv:2402.03300、2024。

N. Shazeer、A. Mirhoseini、K. Maziarz、A. Davis、QV Le、GE Hinton、J. Dean。とてつもなく大きなニューラル ネットワーク: スパース ゲートのエキスパート混合層。第 5 回国際学習表現会議、ICLR 2017。OpenReview.net 、2017 年。URL https://openreview.net/forum?id=B1ckMDqlg 。

F. Shi、M. Suzgun、M. Freitag、X. Wang、S. Srivats、S. Vosoughi、HW Chung、Y. Tay、S. Ruder、 D. Zhou、D. Das、および J. Wei。言語モデルは、多言語の思考連鎖推論システムです。第11回国際学習表現会議、ICLR 2023、ルワンダ、キガリ、2023年5月1日～5日。OpenReview.net 、2023年。URL https://openreview.net/forum?id=fR3wGCk-IXp 。

Y. Shibata, T. Kida, S. Fukamachi, M. Takeda, A. Shinohara, T. Shinohara, and S. Arikawa. バイトペアエンコーディング: パターンマッチングを高速化するテキスト圧縮方式。1999 年。

J. Su、M. Ahmed、Y. Lu、S. Pan、W. Bo、Y. Liu。Roformer :回転式変圧器を備えた強化変圧器位置埋め込み。ニューロコンピューティング、568:127063、2024。

K. Sun、D. Yu、D. Yu、C. Cardie。中国語機械読解に挑戦するための事前知識の調査、2019a。

M. Sun、X. Chen、JZ Kolter、Z. Liu。大規模言語モデルにおける大規模な活性化。arXiv プレプリントarXiv:2402.17762, 2024.

X. Sun、J. Choi、C.-Y. Chen、N. Wang、S. Venkataramani、VV Srinivasan、X. Cui、W. Zhang、およびK. Gopalakrishnan。ディープニューラルネットワークのハイブリッド 8 ビット浮動小数点 (HFP8) トレーニングと推論。ニューラル情報処理システムの進歩、32、2019b。

M. Suzgun、N. Scales、N. Schärli、S. Gehrmann、Y. Tay、HW Chung、A. Chowdhery、QV Le、 EH Chi、D. Zhou、他「挑戦的なビッグベンチタスクと、思考連鎖で解決できるかどうか」arXiv プレプリント arXiv:2210.09261、2022 年。

V. Thakkar、P. Ramani、C. Cecka、A. Shivam、H. Lu、E. Yan、J. Kosaian、M. Hoemmen、H. Wu、A. Kerr、M. Nicely、D. Merrill、D. Blasig、F. Qiao、P. Majcher、P. Springer、M. Hohnerbach、 J. Wang、M. Gupta。 CUTLASS、2023 年 1 月。URL https://github.com/NVIDIA/cutlass。

H. Touvron、T. Lavril、G. Izacard、X. Martinet、M.-A. Lachaux、T. Lacroix、B. Rozière、N. Goyal、 E. Hambro、F. Azhar、他「LLaMA: オープンで効率的な基礎言語モデル」 arXiv プレプリント arXiv:2302.13971, 2023a。

H. トゥヴロン、L. マーティン、K. ストーン、P. アルバート、A. アルマハイリ、Y. ババエイ、N. バシリコフ、S. バトラ、 P. バルガヴァ、S. ボサレ、D. ビケル、L. ブレッチャー、C. カントンフェレール、M. チェン、G. ククルル、D. エシオブ、 J. フェルナンデス、J. フー、W. フー、B. フラー、 C. ガオ、V. ゴスワミ、N. ゴヤル、A. ハーツホルン、S. ホセイニ、

R. ホウ、H. イナン、M. カルダス、V. ケルケズ、M. カブサ、I. クルーマン、A. コレネフ、PS コウラ、 M. ラショー、T. ラブリル、J. リー、D. リスコビッチ、Y. ルー、Y. マオ、X. マーティネット、T. ミハイロフ、P. ミシュラ、 I. モリボグ、Y. ニー、A. ポールトン、J. Reizenstein、R. Rungta、K. Saladi、A. Schelten、R. Silva、EM
スミス、R. スブラマニアン、XE タン、B. タン、R. テイラー、A. ウィリアムズ、JX クアン、P. シュー、Z. ヤン、 I. ザロフ、Y. チャン、A. ファン、M. カンバドゥル、S. ナラン、A. ロドリゲス、R. ストジニッチ、S. エドゥノフ、 T. シャロム。 Llama 2: オープンな基盤と微調整されたチャット モデル。 CoRR、 abs/2307.09288、2023b 。土井: 10.48550/arXiv.2307.09288。 URL https://doi.org/10.48550/arXiv.2307。 09288.

A. バスワニ、N. シャジーア、N. パルマー、J. ウシュコライト、L. ジョーンズ、A.N. ゴメス、 .カイザー、そしてI.ポロ•スクヒン。必要なのは注意力だけです。神経情報処理システムの進歩、 30、2017 。

L. Wang、H. Gao、C. Zhao、X. Sun、および D. Dai。専門家の混合のための補助損失のない負荷分散戦略。CoRR 、 abs/2408.15664、2024a。URL https://doi.org/10.48550/arX iv.2408.15664。

Y. Wang、X. Ma、G. Zhang、Y. Ni、A. Chandra、S. Guo、W. Ren、A. Arulraj、X. He、Z. Jiang、T. Li、 M. Ku、K. Wang、A. Zhuang、R. Fan、X. Yue、W. Chen。 Mmlu-pro: より堅牢で挑戦的なマルチタスク言語理解ベンチマーク。 CoRR、 abs/2406.01574、2024b。
URL https://doi.org/10.48550/arXiv.2406.01574.

T. Wei、J. Luan、W. Liu、S. Dong、B. Wang。Cmath: 言語モデルは中国語に合格できるか
小学校の算数のテスト？、2023年。

M. Wortsman、T. Dettmers、L. Zettlemoyer、A. Morcos、A. Farhadi、およびL. Schmidt。大規模視覚言語モデルのための安定した低精度トレーニング。ニューラル情報処理システムの進歩、36 :10271–10298、2023年。

H. Xi、C. Li、J. Chen、J. Zhu。4ビット整数によるトランスフォーマーのトレーニング。ニューラルネットワークの進歩
情報処理システム、36:49146–49168、2023年。

CS Xia、Y. Deng、S. Dunn、L. Zhang。エージェントレス :LLMベースのソフトウェアエンジニアリングエージェントの謎を解明。arXivプレプリント、2024年。

H. Xia、T. Ge、P. Wang、S. Chen、F. Wei、および Z. Sui。投機的デコード :投機的実行を利用して seq2seq 生成を高速化する。計算言語学協会の調査結果 :EMNLP 2023、シンガポール、2023 年 12 月 6 ～ 10 日、 3909 ～ 3925 ページ。
計算言語学協会、2023年。URL https://doi.org/10.18653/v1/ 2023.findings-emnlp.257。

G. Xiao、J. Lin、M. Seznec、H. Wu、J. Demouth、および S. Han。「Smoothquant:大規模言語モデルのための正確で効率的なトレーニング後の量子化」。国際機械学習会議、38087～38099 ページ。PMLR、2023 年。

L. Xu、H. Hu、X. Zhang、L. Li、C. Cao、Y. Li、Y. Xu、K. Sun、D. Yu、C. Yu、Y. Tian、Q. Dong、W. Liu、 B. Shi、Y. Cui、J. Li、J. Zeng、R. Wang、W. Xie、Y. Li、Y. Patterson、Z. Tian、Y. Zhang、H. Zhou、 S. Liu、 Z. Zhao、Q. Zhao、C. Yue、X. Zhang、Z. Yang、K. Richardson、Z. Lan。 CLUE: 中国語理解評価ベンチマーク。 D. Scott、N. Bel、C. Zong編著、 Proceedings of the 28th International Conference on Computational Linguistics、COLING 2020、バルセロナ、スペイン（オンライン）、2020年12月8日～13日、4762～4772ページ。国際計算言語学委員会、2020年。doi: 10.18653/V1/2020.COLING-MAIN.419。URL https://doi.org/10.18653/v1/2020.coling-main.419。

R. Zellers、A. Holtzman、Y. Bisk、A. Farhadi、Y. Choi。HellaSwag: 機械は本当にあなたの文章を完成させることができるのか? A. Korhonen、DR Traum、L. Màrquez 編著、 Proceedings of the 57th Conference of the Association for Computational Linguistics、ACL 2019、フィレンツェ、イタリア、2019 年 7 月 28 日〜8 月 2 日、第 1 巻: Long Papers、 4791〜4800 ページ。Association for Computational Linguistics、2019 年。doi: 10.18653/v1/p19-1472。URL https://doi.org/10.18653/v1/p19-1472。

W. Zhong、R. Cui、Y. Guo、Y. Liang、S. Lu、Y. Wang、A. Saied、W. Chen、および N. Duan。 AGIEval:基礎モデルを評価するための人間中心のベンチマーク。 CoRR、 abs/2304.06364、2023。土井: 10.48550/arXiv.2304.06364。URL https://doi.org/10.48550/arXiv.2304.06364。

J. Zhou、T. Lu、S. Mishra、S. Brahma、S. Basu、Y. Luan、D. Zhou、およびL. Hou。指示に従う大規模言語モデルの評価。arXiv プレプリント arXiv:2311.07911, 2023。

付録

A. 貢献と謝辞

研究とエンジニアリング
劉愛新
ビン•シュエ
王秉軒
ウー•ボチャオ
成達 呂
趙成剛
チェンチー•デン
チャン•チェンユ*
チョン•ルアン
ダマイ•ダイ
Daya Guo
Dejian Yang Deli
Chen Erhang
Li Fangyun
Lin Fucond Dai
Fuli Luo*
Guangbo Hao
Guanting Chen
Guowei Li H. Zhang
Han Bao*
Hanwei Xu
Haocheng
Wang* Haowei
Zhang Honhui Ding
Huajian Xin* Huazuo
Gao Hui Qu Jianzhong
Guo Jiashi Li
Jiawei Wang*
Jingchang
Chen Jingyang Yuan
ジュンジエ•
チウ•ジュンロン•リー•
ジュンシャオ•ソン•カイド
ン
カイフー*
カイゲ•ガオ
カン•グアン
黄克欣
クアイユ
リーン•ワン
張 楽相
梁趙
王立同
張麗月
ミンチュアン•チャン
張明華
明慧唐
パンパン•ファン
王 ペイイー
王 千成
朱 奇豪
陳琴宇
杜秋思
葛 瑞奇
張瑞松
潘瑞哲
王 潤志
徐潤馨
張若宇
シャンハオ•ルー
周尚燕
陳山煌
イエ•シェンフェン
馬志
王詩宇
ユ•シュイピン
周順鋒
シャッティングパン
タオ•ユン
ティエン•ペイ
曾王定
趙万佳*
ウェン•リュー
ウェンフェン•リャン
ウェンジュン•ガオ
ウェンチン•ユー
張文涛
シャオビ
劉暁東
王暁漢
陳暁康
張暁康
シャオタオ•ニエ
シン•チェン
シン•リウ

シン•シエ
劉星超
ユ•シンカイ
ヤン•シンユ
シンユアン•リー
蘇雪成
林旭恒
YK リー
YQ Wang YX
Wei Yang
Zhang Yanhong
Xu Yao Li Yao
Zhao
Yaofeng Sun
Yaohui Wi Yu
Yichao Zhang Yifan
Shi
Yiliang Xiong Ying
He Yishi Piao
Yisong W

タン•イーシュアン
馬 易陽*
劉 怡源
郭永強
ウー•ユー
袁欧
王玉端
岳公
ゾウ•ユヘン
何宇佳
ユンファン•ション
羅玉祥
ユー•ユシャン
劉玉軒
Yuyang Zhou ZF
Wu ZZ Ren
Zehui Ren
Zhangli Sha
Zhe Fu Zhean Xu
Zhenda
Xie Zhengyan
Zhang Zhewen
Hao Zhibin Gou Zhicheng
Ma

ヤン•ジガン
シャオ•ジーホン
呉志宇
李卓樹
グ•ジフイ
朱子佳
劉子俊*
リー•ジリン
謝紫薇
ソン•ジヤン
ガオ•ツィイー
パン•ジジェン

データ注釈Bei Feng Hui Li
JL Cai Jiaqi
Ni

徐雷
Meng Li
Ning Tian RJ
Chen RL Jin
Ruyi Chen
SS Li Shuang
Zhou
Tianyu Sun XQ Li
Xiangyue Jin
Xiaojin
Shen Xiaosha
Chen Xiaowen
Sun Xiaoxiang
Wang Xinnan Song
Shinyi Zhou YX Zhu
Yanhong Xu
Yanping Huang
Yaohui Li Yi
Zheng Yuchen
Zhu Yunxian Ma Zhen
Huang
Zhipeng Xu
Zhongyu Zhang

ビジネスとコンプライアンス
ジ•ドンジエ

ジャン•リャン
ジン•チェン
•レイ•シア•
ミャオジュン ワン•ミ
ンミン リー•ペン
チャン•シャオチ
ン ウー•シェンフェ
ン イェ•ティ•ワン
WLシャオ
ウェイ•アン
王仙祖
新夏山
イン•タン
ユクン•ツァ
ヤン•ユティン
張珍

各役割内で、著者はファーストネームのアルファベット順でリストされます。* マークの付いた名前は、チームを離れた個人を表します。

B. 低精度トレーニングのためのアブレーション研究

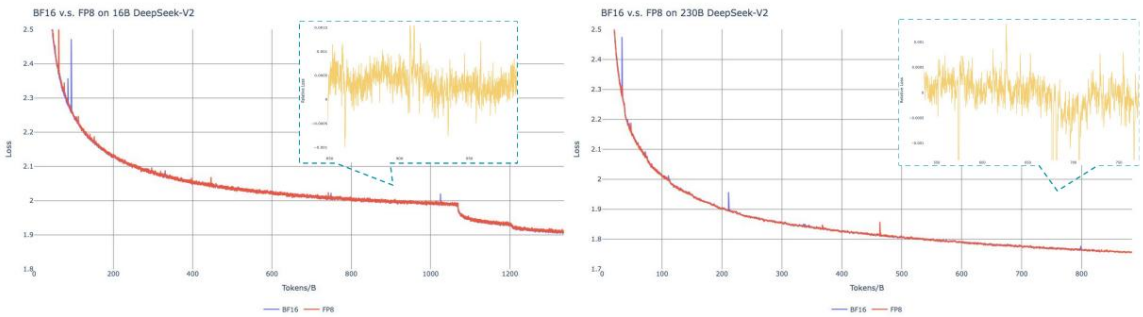


図 10 | BF16 と FP8 トレーニング間の損失曲線の比較。結果は、係数 0.9 の指数移動平均 (EMA) によって平滑化されています。

B.1. FP8 と BF16 のトレーニング

異なるスケールの 2 つのベースライン モデルに基づく BF16 トレーニングとの比較により、FP8 混合精度フレームワークを検証します。小規模では、1.33T トークンで合計約 160 億のパラメータを含むベースライン MoE モデルをトレーニングします。大規模では、約 0.9T トークンで合計約 2300 億のパラメータを含むベースライン MoE モデルをトレーニングします。
図 10 にトレーニング曲線を示し、高精度の蓄積と細粒度の量子化戦略により相対誤差が 0.25% 未満に留まることを示しています。

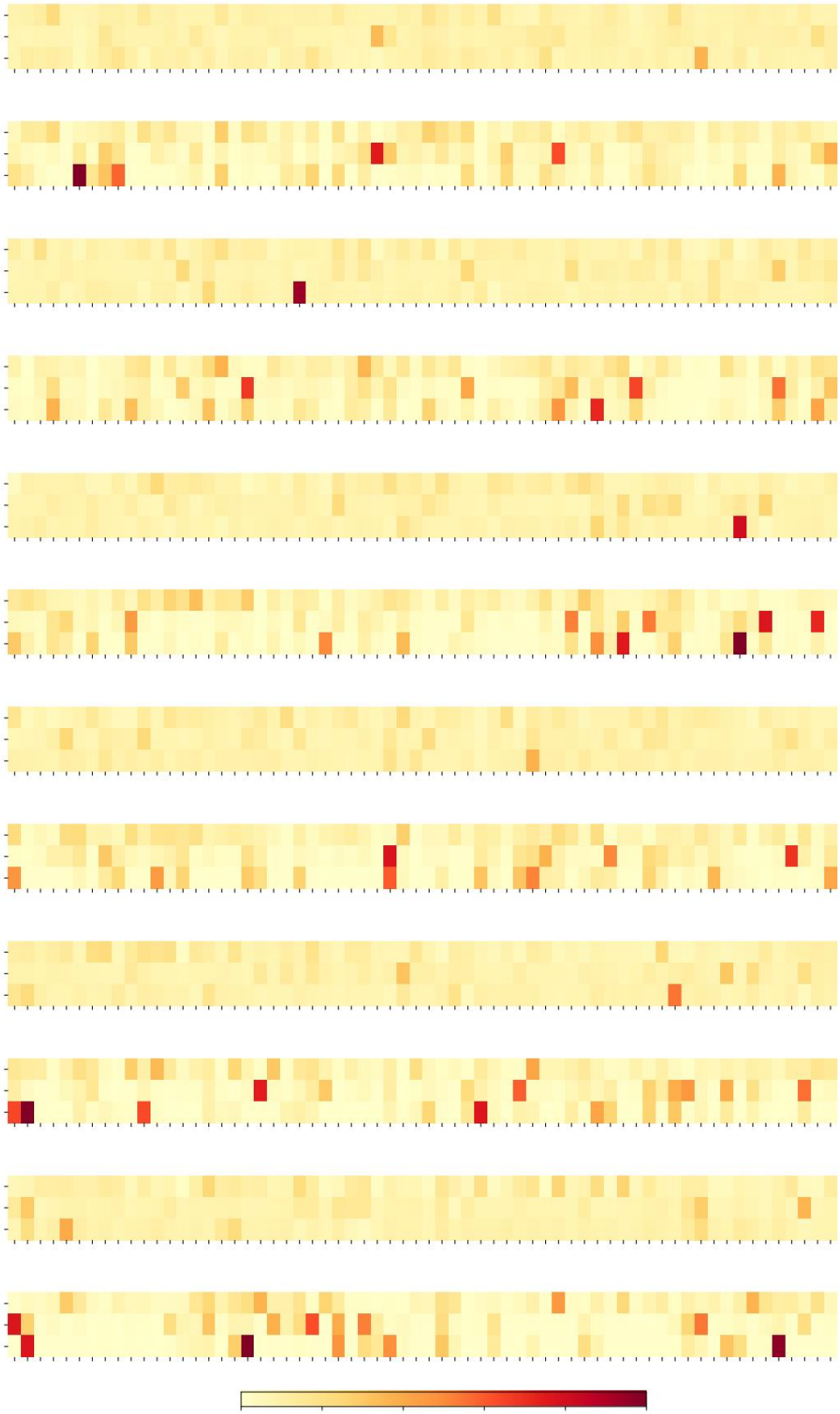
B.2. ブロック単位の量子化についての議論

タイル単位の細粒度量子化は、特徴の外れ値によって生じる誤差を効果的に軽減しますが、活性化量子化には異なるグループ化、つまりフォワードパスでは1x128 、バックワードパスでは128x1 が必要です。活性化勾配についても同様のプロセスが必要です。簡単な戦略は、モデルの重みを量子化するのと同じように、 128x128要素ごとにブロック単位の量子化を適用することです。この方法では、バックワードには転置のみが必要です。したがって、 Dgrad に関連付けられたすべてのテンソルをブロック単位で量子化する実験を行います。結果から、活性化勾配を計算し、チェーンのように浅い層に逆伝播する Dgrad 操作は、精度に非常に敏感であることがわかります。具体的には、活性化勾配のブロック単位の量子化により、

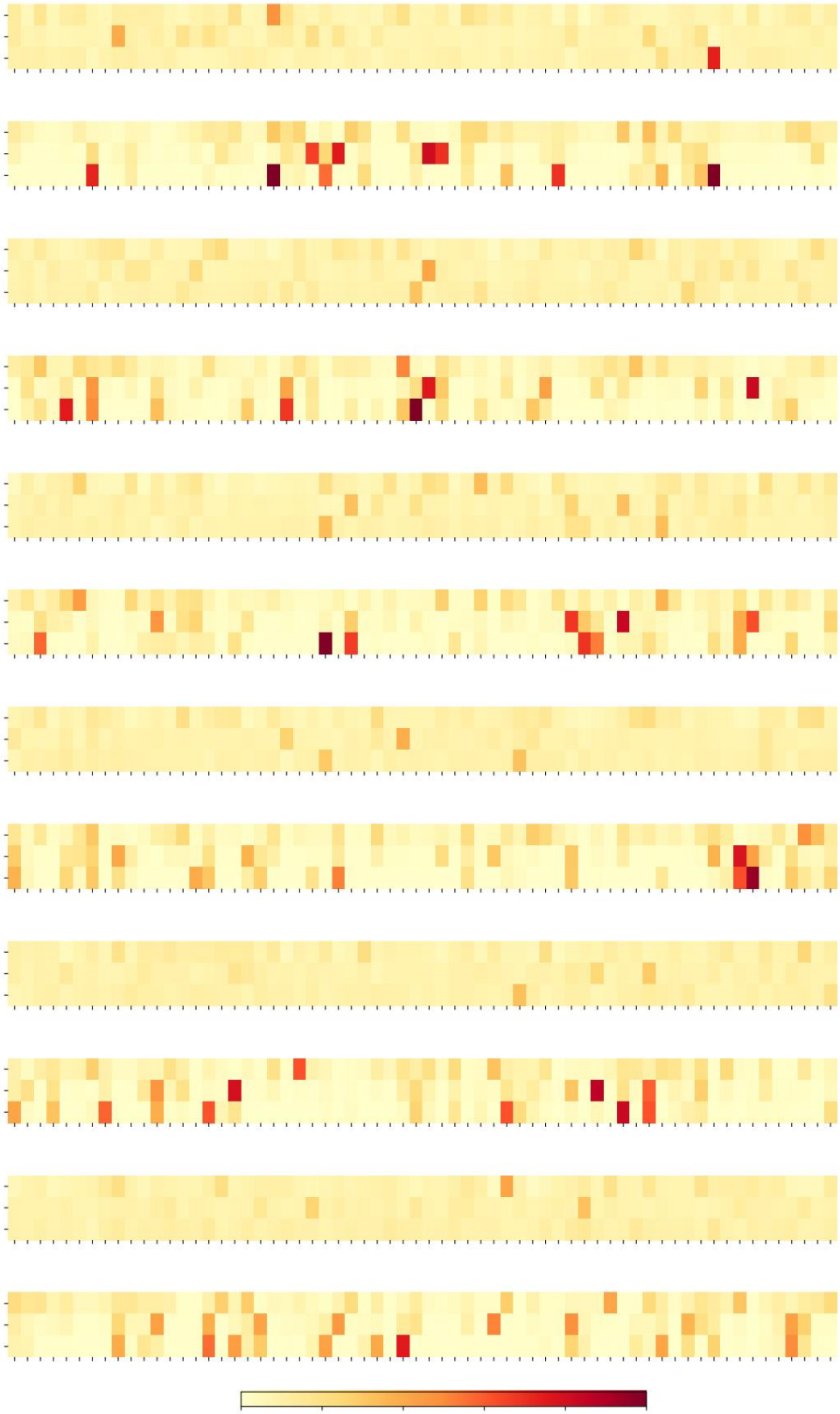
約 160 億個のパラメータで構成され、約 3000 億個のトークンでトレーニングされた MoE モデルにおけるモデルの発散。この感度は、トークン間の活性化勾配が非常に不均衡であるため、トークン相関の外れ値が発生すると仮定しています(Xi et al., 2023)。これらの外れ値は、ブロック単位の量子化アプローチでは効果的に管理できません。

C. 16B補助損失ベースおよび補助損失ベースの専門家の特化パターン無料モデル

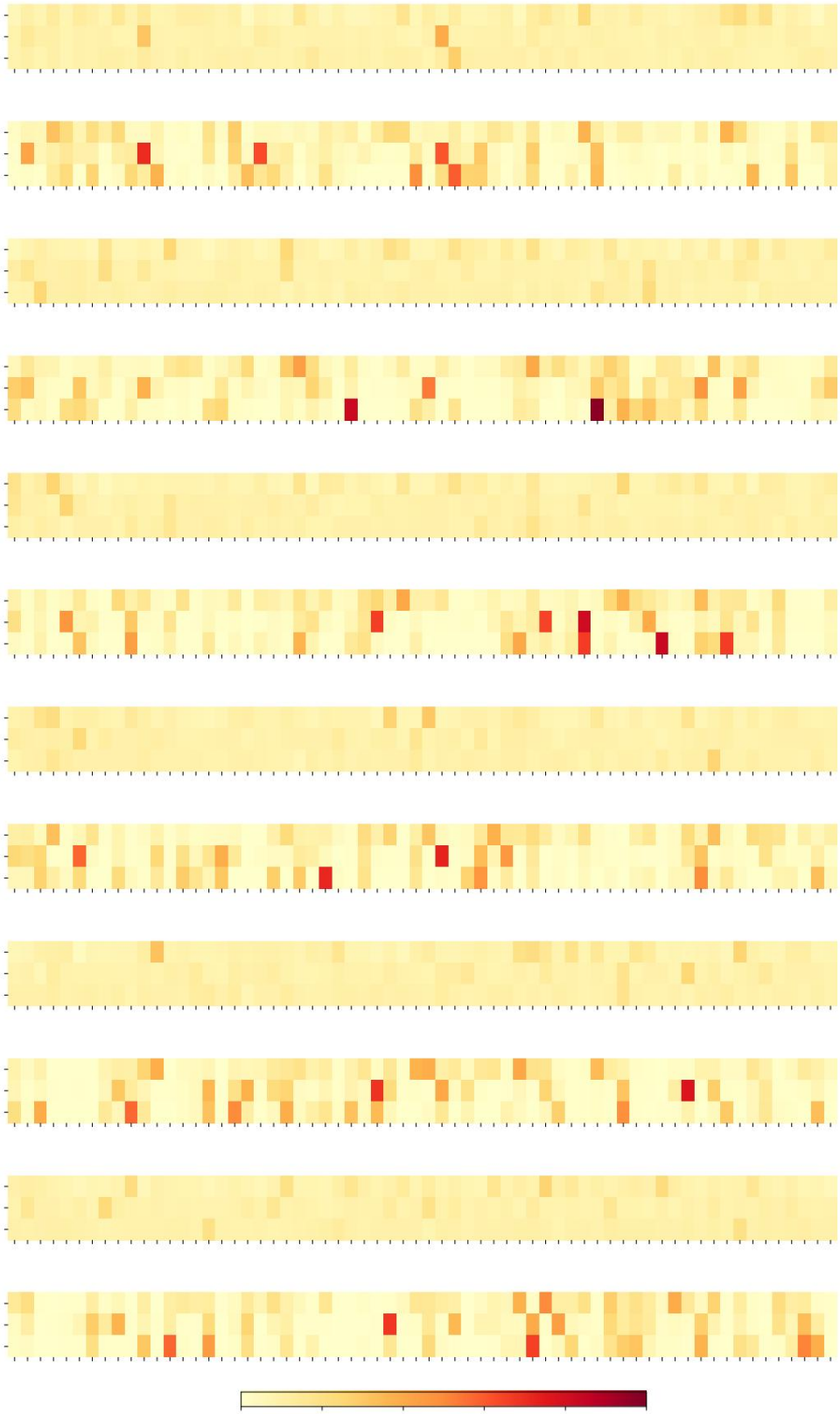
Pile テスト セットで、16B 補助損失ベースのベースラインと補助損失のないモデルの専門家負荷を記録します。図 10 に示すように、補助損失のないモデルは、すべてのレイヤーで専門家の特化度が高い傾向があります。



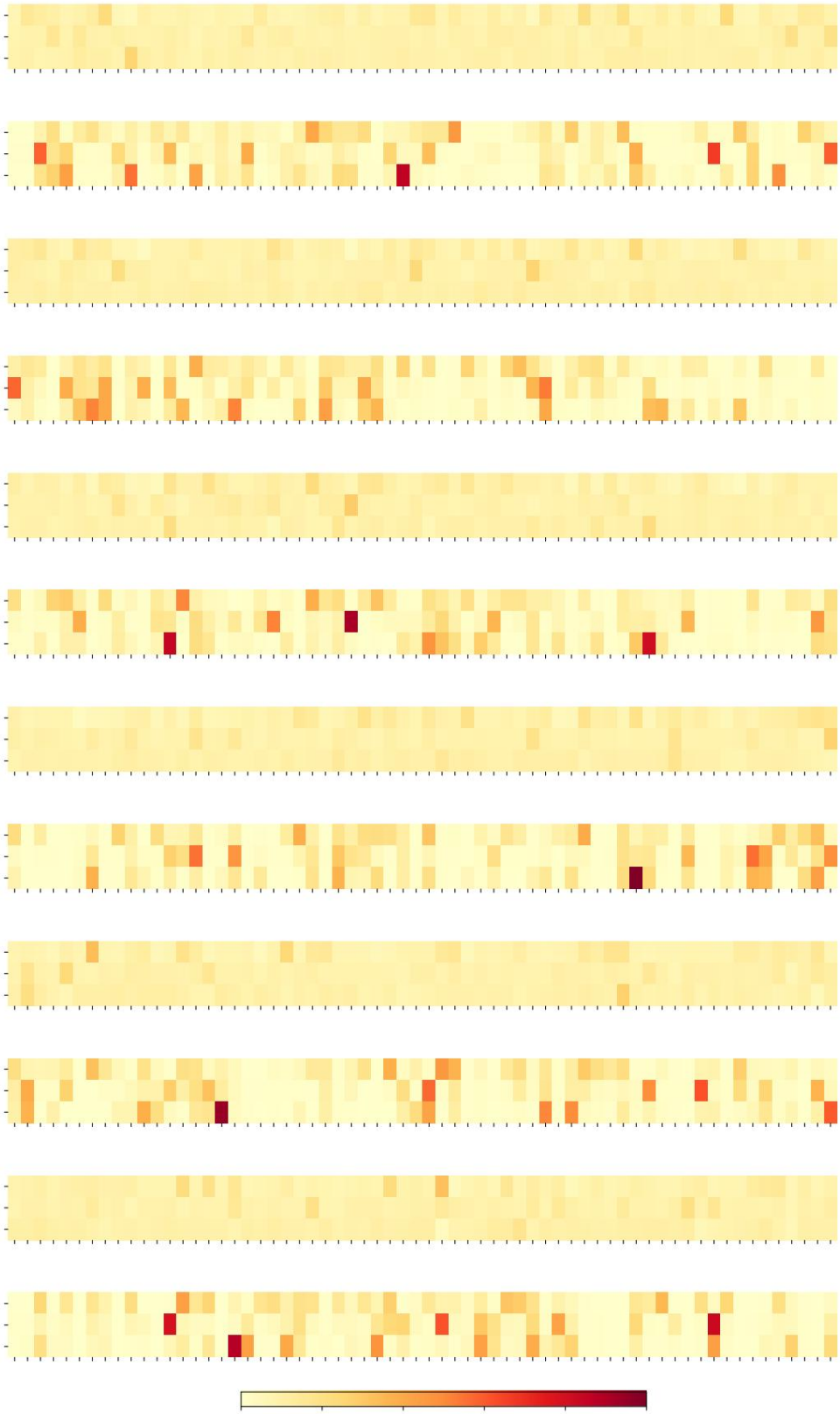
(a) レイヤー1〜7



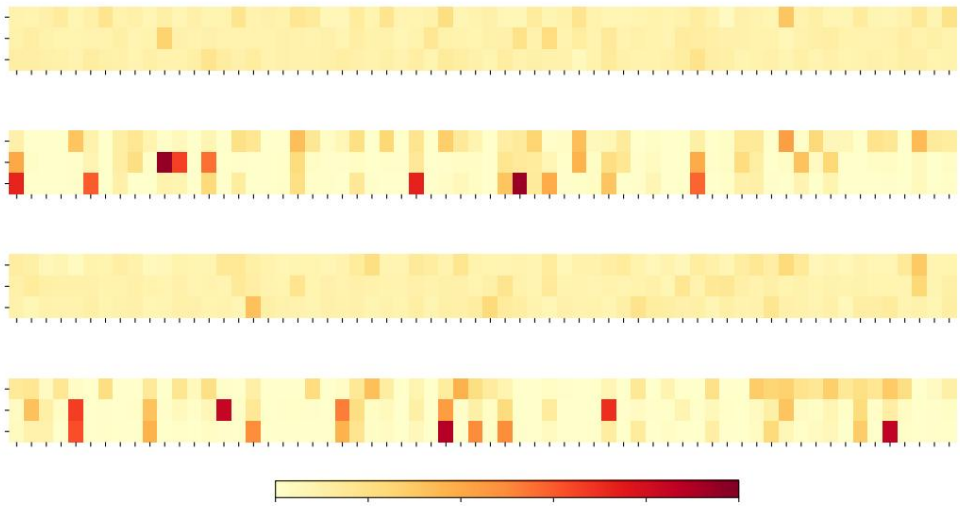
(b) レイヤー7〜13



(c) レイヤー13〜19



(d) レイヤー19-25



(e) レイヤー25-27

図 10 | Pile テスト セットの3 つのドメインにおける補助損失なしモデルと補助損失ありモデルのエキスパート負荷。補助損失なしモデルは、補助損失ありモデルよりもエキスパート特化パターンが大きいことが示されています。相対エキスパート負荷は、実際のエキスパート負荷と理論的にバランスのとれたエキスパート負荷の比率を示します。