ChatGPT

NII発表「LLM-jp-4 8B」「LLM-jp-4 32B-A3B」深掘りと国内外モデル比較

エグゼクティブサマリ

NII（大規模言語モデル研究開発センター／LLMC）は、約12兆トークン規模（総学習 11.7T tokens）の多段学習で構築した国産LLMとして「LLM-jp-4 8B」（密モデル）と「LLM-jp-4 32B-A3B」（MoE：総32B・活性3.8B規模）を、オープンソースライセンス（Apache 2.0）で公開した。最大の設計上の特徴は、(1) 65,536トークンの長文入出力（コンテキスト）を前提にした設定、(2) 32B側が 128-expert/Top-8 のMoEで「計算量はA3B級、重みは32B級」という“推論効率志向”、(3) 日本語と英語の両方での対話性能をLLM-as-a-Judgeで評価し、3回の推論・評価の平均としてスコアを提示している点にある。 [1]

学習データは、総プール 19.5T tokens（英語 17.8T／日本語 0.7T／その他言語 0.85T／コード 0.2T）を用意し、最終的な事前学習では“混合比を最適化”して 10.5T tokens を使用、さらに中間学習（mid-training）として合成データやInstruction Pre-Trainingデータを含む 1.2T tokens を追加した（合計 11.7T tokens）。 [2]

評価面では、llm-jp-judge が提示する日本語MT-Bench／MT-Bench（英語）で、LLM-jp-4 32B-A3B（thinking, reasoning_effort=medium）が 7.82（JA）/ 7.86（EN）、LLM-jp-4 8B（thinking, medium）が 7.54（JA）/ 7.79（EN）を報告している。さらに安全性（AnswerCarefully）と一般品質（llm-jp-instructions）も同一枠組みで測定している。 [3]

一方で、再現性に関する一次情報は「評価コードや一部データセットは公開されているが、(a) 評価の judge が商用モデル（gpt-5.4）であること、(b) 学習コーパスは“多くは公開”されつつも“ライセンス制約で未公開部分がある”こと、(c) トークナイザ語彙が“純SentencePiece学習では再現できない”構成であること」により、完全再現（end-to-end）の難度は高い。 [4]

比較対象としてユーザー指定の Qwen3 系（8B／32B／30B-A3B）と比べると、LLM-jp-4 は「長文65kを“素の設定”で狙い、日英混在の巨大学習と日本語向けデータ整備で対話性能を押し上げる」路線であるのに対し、Qwen3 は「32kネイティブ＋YaRNで131k拡張」「thinking/non-thinking切替」「エージェント（Qwen-Agent）まで含む総合スタック」志向が強い。 [5]

公開物とモデル群の全体像

公開物は大きく「モデル重み」「利用のための付帯コード（トークナイザ・テンプレート・パーサ）」「評価基盤」「（多くは）学習データ」から成る。

モデルは Hugging Face 上で base/thinking（+ instruct 系の存在も示唆）として提供され、ライセンスは Apache 2.0 が明示されている。Hugging Face [6] 上のモデルカードでは、base は“事前学習＋中間学習のみ”、thinking は “SFT＋DPOで整列（RLなし）”と整理されている。 [7]

周辺ツールとして、(1) トークナイザ／チャットテンプレート（llm-jp-tokenizer）、(2) 実装例（cookbook）、(3) LLM-as-a-Judge評価（llm-jp-judge）、(4) 多タスク自動評価（llm-jp-eval）が公式に誘導される。GitHub [8] 上でApache 2.0として提供されている。 [9]

国内プロジェクトとしての位置づけを時系列で把握するため、公開情報から最低限のタイムラインを作る（ここでは主要公開に限定。年内の追加計画など詳細は未指定）。 [10]

```
timeline
  title 日本語オープンLLM系の公開タイムライン（主要イベント）
  2023 : LLM-jp 13B v1.0 公開（日本語LLM系の初期公開群）
  2024 : LLM-jp-13B v2.0 公開（構築資源公開の流れ強化）
  2024 : Swallow（Mistral）公開（Mistral/Mixtralの日本語強化）
  2025 : LLM-jp-3.1 instruct4 公開（継続事前学習＋後段改善）
  2026-04-03 : LLM-jp-4 8B / 32B-A3B 公開（12兆級コーパスで学習）
```

モデル設計とアーキテクチャ

LLM-jp-4 は「8B密モデル」と「32B-A3B MoEモデル」を並行提供し、両者とも最大 65,536 トークンの長文処理を前提にしている。 [11]

主要スペック（一次情報で確認できる範囲）は以下。

LLM-jp-4 8B（dense）

アーキテクチャは Llama 系（config上は `LlamaForCausalLM`）で、32層・hidden 4096・Attention head 32（KV head 8のGQA）になっている。最大コンテキスト `max_position_embeddings=65536`、RoPE拡張は `rope_scaling=null` のまま `rope_theta=500000` を採用しており、RoPE設定による長文化を選んでいることが分かる。dtype は bfloat16。語彙数は 196,608。 [12]

LLM-jp-4 32B-A3B（MoE）

アーキテクチャは Qwen3 MoE 系（config上は `Qwen3MoeForCausalLM`）で、32層・hidden 2560・Q head 40（KV head 4）・expert 128・token当たりTop-8（`num_experts_per_tok=8`）が明示される。router補助損失係数は 0.01。こちらも 65,536 トークン、`rope_theta=500000`、dtype bfloat16、語彙数 196,608。総パラメータ 32.14B、活性パラメータ 3.83B（A3B）という表記が、モデルカードの表と発表の説明で一致する。 [13]

設計上の含意（推定を含む）

32B-A3B は「重み（保存・配布）は32B級」のためVRAM消費は大きい一方、「活性（計算）パラメータ」は3〜4B帯に落ちるため、同一世代の密32Bよりは計算量あたりのスループットが出やすい設計である（推定：一般にMoE Top-kはdenseよりFLOPsを抑えやすい）。ただし expert の分散（expert parallel）やゲーティングのオーバーヘッド、バッチサイズ依存など運用要因が支配的になり得る点は未指定。 [14]

トークナイザ／入出力フォーマット

LLM-jp-4 は Hugging Face tokenizers の Unigram byte-fallback をベースに、llm-jp-tokenizer ver4.0 の語彙を変換しているとし、さらに「純粋なSentencePiece学習では語彙を再現できない」と明記している。つまり、語彙生成の再現実験は“データ公開待ち”か“追加手順の公開待ち”が前提になる。 [15]

また、付属の chat_template は“Current date”“Reasoning: reasoning_effort”の埋め込み、tool calls（functions.*）や、analysis/final のチャネルタグを扱う構成になっている。これは、thinkingモデルが“reasoning_effort（low/medium/high）”を持つ設計と整合している。 16

さらに cookbook は、(a) trust_remote_code が必要、(b) OpenAI Harmony response format を既定構造として採用、(c) detokenize周辺の既知問題回避のため独自 tokenizer 実装（llmjp4_tokenizer.py）を同梱、といった運用上の注意を与える。供給網の観点では“trust_remote_code を避けるならコードを明示的に取り込め”という選択肢も提示される。 17

学習データと品質管理・ライセンス

学習データ量と内訳

発表によれば、事前学習用の総コーパスは 19.5T tokens で、その内訳は英語 17.8T、日本語 700B、その他言語 850B、コード 200B である。ここから混合比を最適化して 10.5T tokens を事前学習に投入し、さらに中間学習として合成データ＋Instruction Pre-Trainingデータ 1.2T tokens を加えている。 2

この 19.5T の“総プール”内訳（割合）は、上記数値から算出すると概ね「英語91.0%／日本語3.6%／その他言語4.3%／コード1.0%」である（算出）。 18

```
pie title 事前学習用コーパス総プール（19.5T tokens）の内訳（発表値から算出）
  "英語 17.8T（約91.0%）" : 17.8
  "日本語 0.7T（約3.6%）" : 0.7
  "その他言語 0.85T（約4.3%）" : 0.85
  "コード 0.2T（約1.0%）" : 0.2
```

ここで重要なのは、「日本語比率が総プールでは小さく見える」一方で、「実際に 10.5T をどうサンプリングしたか（最終混合比）」は未指定である点である。発表は“混合比を最適化”と述べるのみで、例えば“日本語・コードを高頻度サンプリングして比率を上げた”かどうかは一次情報から確定できない。 18

データソースと“入手可能性”の扱い

発表は、インターネット上の公開データ、政府・国会の文書、合成データ等でコーパスを整備したと述べ、さらに「OSAIDに配慮し、第三者も入手可能な良質な学習コーパスの収集・選別・構築」に言及している。 18

ただしモデルカードでは、「コーパスの大半は公開したが、ライセンス制約により一部は公開対象外」と明記される。つまり「第三者が“入手可能”であること」と「第三者が“再配布可能／同一形で再現可能”であること」は同義ではなく、ここが再現性のボトルネックになる。 19

また、モデルカードには国語研の whole-NWJC を利用した旨が明記される。 20
関連して、国立国会図書館（NDL）は NII との連携（WARPのURLリスト提供）を公開しており、ウェブアーカイブ資源が日本語データ整備に寄与している可能性が高い（ただし、LLM-jp-4 の 19.5T にどの程度・どのように組み込まれたかは未指定）。 21

ライセンスと“オープンソースAI”の含意

モデル重みと主要リポジトリは Apache 2.0 として提供される。 22

一方、「OSAID（Open Source AI Definition）に配慮」との言及はあるが、OSAID自体は Open Source Initiative [23] が公開する定義であり、“使用・研究・改変・共有”の自由を成り立たせる前提として、システムを改変するための望ましい形（preferred form）と利用の手段へのアクセスを要求する。 [24]

LLM-jp-4 は「モデル重みとコード」はApache 2.0で明確だが、「学習データが完全に公開されていない」とモデルカードが明示するため、“厳密にOSAIDの完全要件を満たすか”は未指定（少なくとも“学習データの完全開示”という観点では不足が残る可能性）と評価するのが安全である。 [25]

品質管理・前処理・フィルタリング基準

ユーザーが要望した「前処理・フィルタリング基準（重複除去、品質スコア閾値、PII除去、毒性除去、言語判定、ドメイン除外など）」は、この発表本文および公開モデルカードの範囲では未指定である。 [2]

ただし背景として、LLM-jpが“複数組織横断でモデル・コーパス・評価を公開してきた”こと、また過去の技術報告でコーパス設計や混合比の課題（質と量、どの程度のフィルタリング、言語混合比）を明示してきたことは確認できる。 [26]
このため、LLM-jp-4においても何らかの品質フィルタリング・重複除去・逸脱文書排除が行われた蓋然性は高いが、具体的閾値・手順・監査結果は未指定として扱うべきである。

トレーニング・アラインメント手法

事前学習から中間学習までの骨格

一次情報で確定できるのは「10.5T tokens の事前学習＋1.2T tokens の中間学習＝11.7T tokens」「中間学習には合成データとInstruction Pre-Trainingデータが含まれる」「計算資源としてABCI 3.0 を利用」という骨格である。 [2]

そのうえで、LLM-jpは別途ブログで“mid-training”実験の具体例を出しており、ABCI3.0上で Megatron-LM 系フレームワークを用いた事例や、学習率スケジュール（固定 vs 線形減衰）比較などを公開している。これは LLM-jp-4 における中間学習の思想（高品質データを挟む）と整合的だが、LLM-jp-4 本番学習が同一設定であることは未指定である（推定に留める）。 [27]

最適化器・学習率・混合精度・分散学習

ユーザー指定の観点のうち、以下は一次情報からは未指定である。

最適化器（AdamW/Lion/Adafactor 等）、学習率スケジュール（warmup/decay 形状、最大学習率、終端LR）、weight decay、勾配クリッピング、勾配累積、正規化（μP等）、分散方式（DP/TP/PP/EP、ZeRO/FSDP、チェックポイント方式）、混合精度の詳細（BF16/FP16/FP8、loss scaling）、ドロップアウト設定（訓練時）など。 [28]

ただし、公開 config は推論側の dtype として bfloat16 を明記しており、少なくとも“重みはBF16での配布”が前提である。 [29]

長文処理と“位相的手法”の扱い

LLM-jp-4 の長文（65,536）の根拠は、Hugging Face の config に `max_position_embeddings=65536` が明示され、RoPE設定として `rope_theta=500000`、`rope_scaling=null` が採用されていることから確認できる。これは“RoPEのパラメータ設定を変える”タイプの長文化であり、追加のrope_scaling（YaRN等）の指定は見えない。 [29]

一方、Qwen3は「32,768ネイティブ＋YaRNで131,072拡張」と明示するため、長文化の設計哲学が異なる（LLM-jp-4は65kを標準レンジ、Qwen3は32kを標準＋拡張手段を併記）。 30

蒸留・LoRA等の微調整戦略

LLM-jp-4の公開モデルに関して、「蒸留（distillation）を主要工程として使ったか」「LoRA / QLoRA を公式に推奨するか」は未指定である。 31
ただし、後段整列は“SFT＋DPO（RLなし）”が明確で、post-training datasets のリンクも提示されている（実際のデータ可用性は別途確認が必要）。 32

運用上は、8B（Llama系）／32B-A3B（Qwen3-MoE系）という“広くサポートされたアーキテクチャ”のため、一般的なPEFT（LoRA等）が適用できる見込みは高い（推定）。ただし、thinking/instructモデルはHarmony形式・専用トークナイザ・専用テンプレートを前提にしているため、微調整時にテンプレート整合と出力パースを壊さない設計が必要になる。 33

評価結果と再現性

提示ベンチマークと評価法

評価の中心は llm-jp-judge による LLM-as-a-Judge であり、MT-Bench（英語）と日本語MT-Benchに加え、安全性ベンチマーク AnswerCarefully v2.0（テストセットから336問）と、llm-jp-instructions（テストセット400問）を用いる。評価モデル（judge）は `gpt-5.4-2026-03-05` が明示され、スコアは“推論と評価を3回回して平均”としている。OpenAI 34 のAPIまたは互換APIを使う設計になっており、.env にキー設定が必要である。 35

スコア

モデルカードのテーブル、および発表本文の数値から、主要スコアを同一枠組み（gpt-5.4 judge）で整理すると以下になる。 36

モデル（評価設定）	日本語 MT-Bench	MT-Bench (EN)	AnswerCarefully	llm-jp-instructions
gpt-4o-2024-08-06	7.29	7.69	4.00	4.07 20
Qwen3-8B（発表内比較）	7.14	7.69	未指定	未指定 18
llm-jp-4-8b-thinking（reasoning_effort=medium）	7.54	7.79	3.69	3.54 20
llm-jp-4-32b-a3b-thinking（reasoning_effort=medium）	7.82	7.86	3.70	3.61 20

ここから読み取れるのは、「日本語MT-Benchでは 32B-A3B が gpt-4o および Qwen3-8B の発表値を上回る」「英語MT-Benchでも 32B-A3B が gpt-4o を上回り、Qwen3-8Bの発表値（7.69）も上回る」「安全性・品質（AnswerCarefully／llm-jp-instructions）は gpt-4o より低いが、同枠組みでモデル改善の回帰テストができる」点である。 31

再現性・統計的有意性の検証観点

再現性に関し、公開情報で確定できる強みは「評価コードが公開され、データセット入手手順（含：アクセス申請が必要なもの）と、生成→評価の分離実行が記述されている」点である。 [37]

一方で、厳密な再現性（特に統計的有意性）には以下の不足が残る。

第一に、LLM-as-a-Judge　である以上、judgeモデルの温度・プロンプト・非決定性がスコアに影響しうる。平均を3回取っていることは一定の揺れ抑制だが、分散や信頼区間、ブートストラップ等の有意性評価は未指定である。 [35]

第二に、llm-jp-judge　自身が「ライセンス都合で、論文で使ったデータセットと一部異なる」と明記しており、同名ベンチマークでも“完全一致の再現”ができないケースがある。 [38]

第三に、学習側の再現（データの完全再構築＋同一学習の追試）については、コーパスの一部が未公開であること、トークナイザ語彙が純SentencePiece学習では再現できないことが障害となる。 [15]

参考としての llm-jp-eval 図

発表資料内には llm-jp-eval による項目別評価図が掲載され、複数領域（日本語・英語・推論・安全等）の相対評価が示されている。ただし図は視覚提示が中心で、スカラー値一覧は未指定のため、厳密比較は告知された評価パイプラインでの再走が必要になる。 [39]

他モデルとの比較評価

ユーザー要望の「他国産LLM」としては、日本国内研究機関による派生・継続事前学習モデル（SwallowやQwen3 Swallow等）と、国内企業の日本語最適化モデル（例：Rakuten AI 7B）を、比較軸に沿って整理する。ここで注意点として、ユーザー候補に含まれる Qwen3 は中国企業由来で“国産（日本）”ではないが、実務上の主要比較対象として要求されているため同列に置く。 [40]

比較表（スペック・ライセンス・ベンチマーク）

下表は「一次ソース（発表・モデルカード・公式ページ）で確認できた事実のみ」を基本にし、欠落は未指定とする。

系統	モデル	アーキテクチャ要点	文脈長	ライセンス	代表スコア（MT-Bench系）
LLM-jp-4	8B (dense)	Llama系、32L/4096、GQA(KV=8)、語彙196,608、RoPE theta 500k [12]	65,536 [41]	Apache 2.0 [20]	JA 7.54 / EN 7.79（thinking, medium） [20]
LLM-jp-4	32B-A3B (MoE)	Qwen3-MoE系、128 experts / Top-8、活性3.83B、RoPE theta 500k [13]	65,536 [42]	Apache 2.0 [20]	JA 7.82 / EN 7.86（thinking, medium） [20]

系統	モデル	アーキテクチャ要点	文脈長	ライセンス	代表スコア（MT-Bench系）
Qwen3（海外ベースライン）	8B	8.2B、36L、GQA(Q=32/KV=8)、thinking切替、32k＋YaRNで131k [43]	32,768（+YaRNで131,072）[43]	Apache 2.0 [44]	発表内比較値：JA 7.14 / EN 7.69 [18]
Qwen3（海外ベースライン）	30B-A3B (MoE)	総30.5B/活性3.3B、128 experts / Top-8、32k＋YaRNで131k [45]	32,768（+YaRNで131,072）[46]	Apache 2.0 [45]	未指定（この資料セット内）
Qwen3（海外ベースライン）	32B (dense)	32.8B、64L、GQA(Q=64/KV=8)、32k＋YaRNで131k [47]	32,768（+YaRNで131,072）[47]	Apache 2.0 [48]	未指定（この資料セット内）
日本語強化（学術）	Qwen3 Swallow	Qwen3を日本語・推論で強化、8B/30B-A3B/32B、Apache 2.0 [49]	未指定	Apache 2.0 [49]	未指定（この資料セット内）
日本語強化（学術）	Swallow-MS 7B（Mistral系）	Mistral 7Bから日本語CPT、語彙拡張言及、Apache 2.0継承 [50]	未指定	Apache 2.0 [51]	未指定（この資料セット内）
日本語強化（国内企業）	Rakuten AI 7B	Mistral-7B-v0.1ベースでCPT、日本語最適化形態素解析器、Chatは追加FT [52]	未指定	Apache 2.0 [52]	未指定（この資料セット内）

推論速度・メモリの観点（推定を含む）

重みVRAM（BF16）は概算で「パラメータ数×2バイト」なので、LLM-jp-4　8Bは約16GiB、LLM-jp-4　32B-A3Bは約60GiB（GiB換算、ファイルサイズ表記ではそれぞれ約17.2GB／64.3GB）になる（算出）。 [53]

KVキャッシュ（65,536文脈、BF16）は　　8B（32L・KV=8・head_dim=128）で約8GiB、32B-A3B（32L・KV=4・head_dim=128）で約4GiBになる（算出）。設定上は sliding window が無効であるため、長文時はこのコストが素直に乗る。 [29]

このため “65k文脈をフルに使う”前提では、8Bでも　24GiB級のVRAMが実務上ほぼ必須になり、32B-A3Bは64GiB級＋実装オーバーヘッドで 80GB GPUまたは複数GPU（tensor-parallel等）が現実的、という見立てになる（推定）。一方で通常会話（8k程度）ならKVは約1GiB（8B）まで落ちるため、8Bは24GB〜48GBクラスで扱いやすい。 [54]

Qwen3側は32kネイティブであり、同一VRAMでも“標準運用時のKV負担”は小さくなる一方、YaRNで131kへ拡張する場合は別の品質劣化リスクを明示している（平均文脈が32k以下ならYaRNは推奨しない旨）。 [55]

ファインチューニング容易性・エコシステム

LLM-jp-4 の base は標準的アーキテクチャ（Llama/Qwen3-MoE）であるため、一般的なSFT/LoRA基盤との相性は良い。一方、thinking/instruct は Harmony形式の入出力と、独自トークナイザ・パーサが実装上の前提になり、`trust_remote_code`が必要になる。これは「プロダクト運用の審査（サプライチェーン、SBOM、依存監査）」のコストを増やしうる。[56]

Qwen3は thinking/non-thinking の切替（`enable_thinking`）と `<think>...</think>` ブロックの仕様、推奨デコード設定、さらに Qwen-Agent でのツール実行まで“統合体験”としてのドキュメントが厚い。[57]

国内学術系の Swallow も、公式ページで評価・透明性を前面に出しており、Mistral派生（Swallow-MS）では語彙追加により効率・文字化け抑制・few-shot詰め込みが改善するといった実運用寄りの主張がある。[58]

セキュリティ・有害出力対策

LLM-jp-4 は AnswerCarefully を評価に含めるが、モデルカード自身が「研究開発の初期段階で、人間の意図や安全性に整合するよう十分にチューニングされていない」と注意喚起している。従って、商用導入時は追加の安全アラインメント（ポリシー、拒否設計、監査ログ、レッドチーミング）が必須になる。[35]

実用上の示唆と追加一次情報・再現実験案

導入コストと推奨ユースケース

LLM-jp-4 8Bは、(1) Apache 2.0で法的障壁が相対的に小さい、(2) 65k文脈を“必要なときだけ”使う設計ならオンプレ単体GPUでも現実的、(3) 日本語MT-Benchで 7.5台という対話性能を持つ、という点から、企業内QA、文書要約、議事録・法令文書の要約・照会、日英バイリンガルの対話ボットなどに向く。[59]

LLM-jp-4 32B-A3Bは、(1) MoEで活性パラメータが小さく推論計算は軽めになり得る一方、(2) 重みサイズは大きく、運用は80GB級GPUやマルチGPUが現実的になりやすい、(3) 日本語MT-Benchで 7.82と8Bを上回る、という点から、より高品質対話・長文RAG・多段推論（reasoning_effort制御含む）を求める用途に向く。[60]

制限・リスク

第一に、学習データの完全開示が保証されていないため、第三者による完全な監査や完全再現には限界がある。OSAID準拠をどこまで満たすかは未指定であり、規制業界や公共調達では“オープンの定義”を事前に合意する必要がある。[61]

第二に、thinking/instructは trust_remote_code 前提であり、運用環境のセキュリティポリシー次第では“取り込みコードの固定化（vendoring）”が必要になる。[62]

第三に、評価が LLM-as-a-Judge（gpt-5.4）であるため、同一モデルでも judge の変更で順位が変わる可能性がある。特に日本語MT-Benchは評価器の厳しさの違いでスコアが変わり得ることが明記されている。[35]

追加で確認すべき一次情報ソースと優先度

優先度は「検証の再現性」と「実務導入のリスク低減」に直結する順とする。

最優先は、公開モデルカード（base/thinking）と config.json / tokenizer_config.json（文脈長・RoPE・語彙・MoE設定が確定する）。 63
次に、llm-jp-judge と llm-jp-eval のREADME／DATASET.md（評価手順・データ入手・ライセンス差分）。 37
次に、llm-jp-tokenizer ver4.0（語彙生成・テンプレート・パーサ。再現性上の重要ポイント）。 64
その次に、学習コーパス／SFT／DPOデータの実体（GitLabリンク先で“どこまで公開されているか”、除外部分の説明、利用条件）。モデルカードは“多くは公開、ただし一部除外”としているため、この差分の棚卸しが必要。 20
最後に、OSAID要件（何を“open”とみなすか）を導入前に明文化し、内部監査に組み込む。 65

再現実験の簡易設計案

ここでは「学習の再現」ではなく「評価の再現（モデル比較の再現性検証）」を主目的に、最小〜実務相当の3段階で設計する。学習側（11.7T tokens）は現実的に追試コストが巨大で、また一部コーパス未公開のため“厳密再現”が難しいからである。 31

段階A（最小：スコア再現の確認）
対象：llm-jp-4-8b-thinking と llm-jp-4-32b-a3b-thinking。
評価：llm-jp-judge の MT-Bench（JA/EN）＋ llm-jp-instructions（400）＋ AnswerCarefully（336）。judge は gpt-5.4 相当（利用可能な範囲で同一ID）。3回の推論・評価を実施し平均・標準偏差を出す（標準偏差はこの発表では未指定なので自前算出）。 66

段階B（比較：他モデル同条件比較）
対象：Qwen3-8B / Qwen3-30B-A3B（加えて国内派生として Qwen3 Swallow 等）。
同一プロンプト・同一推論温度・同一judgeで比較し、(1) 平均差、(2) ブートストラップCI（例：質問単位再標本化）、(3) “judge変更”感度（gpt-4o等への切替）を測る。 67

段階C（長文耐性：65kの実運用評価）
対象：LLM-jp-4（65k標準）と Qwen3（32k標準＋YaRN拡張）。
長文RAG（8k/32k/65k）で、(a) 正答率、(b) 参照整合性（引用一貫性）、(c) レイテンシとVRAM、(d) 破綻モード（反復・逸脱）を計測。Qwen3側は平均文脈が32k以下ならYaRN非推奨という注意があるので、YaRN on/off を条件化する。 68

以下は llm-jp-judge を使った最小手順の例である（環境変数や実際の judge モデル名は手元の契約・APIに合わせる）。

```
# 1) セットアップ
git clone https://github.com/llm-jp/llm-jp-judge.git
cd llm-jp-judge
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt

# 2) データ取得（llm-jp-instructions / AnswerCarefully）
bash scripts/download_llm_jp_instructions_v1.0.sh
# AnswerCarefully v2.0 はHFログイン＋アクセス申請が必要（README参照）

# 3) 生成（vLLMクライアント例。モデルはHF上のIDを指定）
MODEL_NAME="llm-jp/llm-jp-4-8b-thinking"
OUTPUT_DIR="./output/${MODEL_NAME//\//_}"
```

```
python3 -m src.llm_jp_judge.generate \
 output.dir="${OUTPUT_DIR}/generation" \
 client=vllm \
 client.model_name="${MODEL_NAME}" \
 benchmark.quality.dataset.path=./data/cache/llm-jp/llm-jp-instructions/v1.0/test.json \
 benchmark.safety.dataset.path=./data/cache/llm-jp/AnswerCarefully/v2.0/test.json

# 4) 評価（OpenAI互換クライアント例）
# .env に OPENAI_API_KEY / OPENAI_BASE_URL 等を設定
python3 -m src.llm_jp_judge.evaluate \
 input.dir="${OUTPUT_DIR}/generation" \
 output.dir="${OUTPUT_DIR}/evaluation" \
 client=openai \
 client.model_name="gpt-5.4-2026-03-05" \
 client.async_request_interval=0.5
```

統計的有意性を最低限押さえるなら、(1) 同一モデルで3回以上の再生成（seed相当の揺れ）を取り、(2) 質問単位でブートストラップして平均差の95%区間を出し、(3) judge を替えて順位の頑健性を見る、という三段が実務的である（いずれも発表内では未指定のため、再現実験側で補完する設計）。 [35]

---

[1] [2] [11] [18] [28] https://www.nii.ac.jp/news/release/2026/0403.html
https://www.nii.ac.jp/news/release/2026/0403.html

[3] https://huggingface.co/llm-jp/llm-jp-4-8b-thinking
https://huggingface.co/llm-jp/llm-jp-4-8b-thinking

[4] [7] [15] [19] [20] [22] [25] [31] [32] [35] [36] [53] [59] [60] [61] [63] https://huggingface.co/llm-jp/llm-jp-4-8b-base
https://huggingface.co/llm-jp/llm-jp-4-8b-base

[5] [12] [29] [41] [54] [68] https://huggingface.co/llm-jp/llm-jp-4-8b-base/blob/main/config.json
https://huggingface.co/llm-jp/llm-jp-4-8b-base/blob/main/config.json

[6] [27] https://llm-jp.nii.ac.jp/en/blog/mid-training-llm-jp-on-olmo2-data-setup-results-and-practical-tips/
https://llm-jp.nii.ac.jp/en/blog/mid-training-llm-jp-on-olmo2-data-setup-results-and-practical-tips/

[8] [23] [45] [46] https://huggingface.co/Qwen/Qwen3-30B-A3B
https://huggingface.co/Qwen/Qwen3-30B-A3B

[9] https://llm-jp.nii.ac.jp/release/
https://llm-jp.nii.ac.jp/release/

[10] https://llmc.nii.ac.jp/achievements/
https://llmc.nii.ac.jp/achievements/

[13] [14] [42] https://huggingface.co/llm-jp/llm-jp-4-32b-a3b-base/blob/main/config.json
https://huggingface.co/llm-jp/llm-jp-4-32b-a3b-base/blob/main/config.json

[16] https://github.com/llm-jp/llm-jp-tokenizer/blob/main/hf/ver4.0/alpha_1.0/chat_template.jinja
https://github.com/llm-jp/llm-jp-tokenizer/blob/main/hf/ver4.0/alpha_1.0/chat_template.jinja

[17] [33] [56] [62] https://github.com/llm-jp/llm-jp-4-cookbook
https://github.com/llm-jp/llm-jp-4-cookbook

21 https://current.ndl.go.jp/car/209858
https://current.ndl.go.jp/car/209858

24 65 https://opensource.org/ai/open-source-ai-definition
https://opensource.org/ai/open-source-ai-definition

26 https://llmc.nii.ac.jp/wp-content/uploads/2024/10/20240925_t1_kawahara.pdf
https://llmc.nii.ac.jp/wp-content/uploads/2024/10/20240925_t1_kawahara.pdf

30 34 40 43 44 57 67 https://huggingface.co/Qwen/Qwen3-8B
https://huggingface.co/Qwen/Qwen3-8B

37 38 66 https://github.com/llm-jp/llm-jp-judge
https://github.com/llm-jp/llm-jp-judge

39 https://www.nii.ac.jp/news/upload/nii_newsrelease_20260403.pdf
https://www.nii.ac.jp/news/upload/nii_newsrelease_20260403.pdf

47 48 55 https://huggingface.co/Qwen/Qwen3-32B
https://huggingface.co/Qwen/Qwen3-32B

49 https://swallow-llm.github.io/qwen3-swallow.en.html
https://swallow-llm.github.io/qwen3-swallow.en.html

50 51 58 https://swallow-llm.github.io/swallow-mistral.ja.html
https://swallow-llm.github.io/swallow-mistral.ja.html

52 https://corp.rakuten.co.jp/news/press/2024/0321_01.html
https://corp.rakuten.co.jp/news/press/2024/0321_01.html

64 https://github.com/llm-jp/llm-jp-tokenizer/tree/main/hf
https://github.com/llm-jp/llm-jp-tokenizer/tree/main/hf