FT-LLM 2026 チューニングコンペ 事後分析レポート

チーム: 043_chattsogpt

A 7-day sprint (Jan 29 – Feb 4, 2026)

大規模言語モデルのファインチューニング技術と評価（Mathematical Task）

課題設定と解決へのアプローチ全体像

- ベースモデル: **llm-jp-4-8b (v4-8b-decay2m-ipt_v3.1-instruct4.zip)**
- タスク: 日本語数学問題のLaTeX解答出力

Block 1

合成データによるSFT

アルゴリズムを用いた
高品質データの生成

Block 2

2段階学習

「推論力（Phase A）」と
「出力形式（Phase B）」の分離

Block 3

多数決推論

N=16のサンプリングによる
Majority Voting

最終スコア
67%
(dev評価)

コンペティションにおける過酷な制約

インターネット遮断

推論時の外部アクセス不可（RAG利用不可）

外部LLMの制限

推論時にGPT-4などの別モデルを利用不可
（独立モデルのみ）

ハードウェア制約

Google Colabを使用

~~ABCI 3.0（NVIDIA H200 GPU 8枚）、
実行時間上限500分~~

Docker提出

容量上限64GB（※当チームは最終23GBで構築）

教訓 (Lesson): リソースが限定されたオフライン環境では、単一モデルの「純粋な推論力」と「フォーマット遵守力」の両立が不可欠。

最大の罠：「89%の幻」と過学習の露呈

驚異的スコアからの劇的崩壊

初期の誤信

初回の2段階SFTでdev評価89%という驚異的スコアを記録。

真相（データリーク）

訓練データ内に dev.jsonl（評価用100問）と極めて類似した問題が混入。

過学習の証拠

未見の高校数学の精度は極端に低く、中学数学のみ高高精度。「問題を解く力」ではなく「暗記」だった。

教訓 (Lesson):「高いスコア ≠ 強いモデル」。評価データとの厳密な重複・類似度チェックはデータ生成パイプライン において絶対不可欠。

データ戦略の変遷と「秘伝のタレ」の発見

Phase 1: 外部データ (JaMARD)

結果：17–26%

ドメイン不一致。入試形式と合わず失敗。

Phase 2: Webスクレイピング

結果：0%

CoT（思考過程）が欠落し、推論できず失敗。

Phase 3: 合成データ (Python+SymPy)

結果：89%

問題生成は成功するも、dev重複による過学習。

Phase 4: クリーンデータ再構築

結果：67%

重複排除＋バランス調整で真の汎化性能を獲得。

勝利の鍵：アルゴリズムによる合成データ生成と精製

合成データ生成（Generation）

LLM生成 ✗ Unpredictable

Python+SymPy生成 ✓ Deterministic

- LLM生成に依存せず、Pythonスクリプトによるテンプレートベース生成を採用。
- `random.seed(42)` と sympy により、品質を数学的に保証。
- 全20単元以上、約5,200問を生成。

精製パイプライン（Cleaning Pipeline）

丁寧語の除去
「ご参考になれば幸いです」等のAI的表現を徹底削除。

↓

長文圧縮
900文字超のデータは「先頭60% + 末尾40%」のみ保持して要約。

↓

重複排除
繰り返しの説明を自動検出。

思考と形式を分離する「2段階SFT」アーキテクチャ

Phase A: 推論力（CoT）の習得

- **データ:** 思考過程付きの合成データ
- **比率:** 70:30（訓練:検証）
- **目的:** モデルに自由に「考え方」を学ばせる。

Phase B: 出力形式の矯正

- **データ:** LaTeX形式のみの公式形式データ
- **比率:** 80:20（訓練:検証）
- **目的:** 学んだ推論を、厳格なLaTeXフォーマットに「書き直す」訓練。

教訓 (Lesson): 推論と形式を同時学習させると相互干渉が生じる。段階を分けることで、両方の精度が劇的に向上する。

学習パラメータ設定とインフラの落とし穴

Rank (r)	16
Alpha	32
Target Modules	全線形層 (q/k/v/o/gate/up/down_proj)
Learning Rate	1e-4 / Dropout: 0.05 / Epoch: 1
Environment	Google Colab (A100)
Docker Image	23GB (043_chattsogpt.tar)

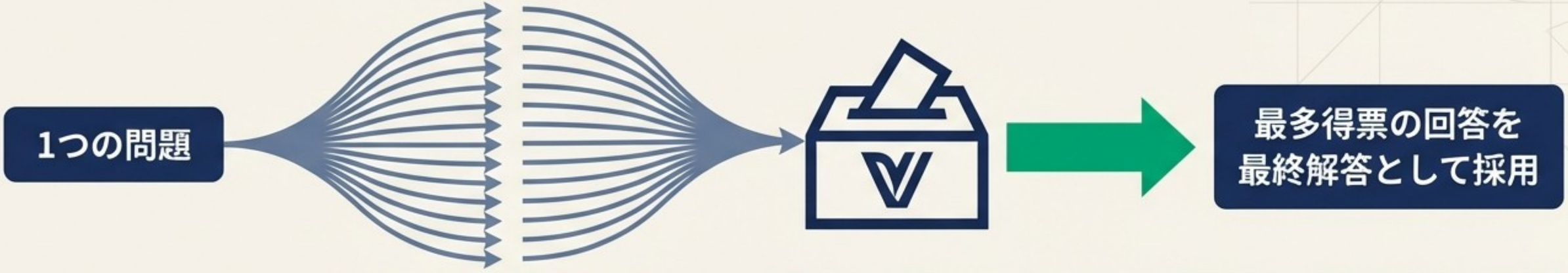
インフラの落とし穴と回避策

Issue (問題):
Google Driveの同期遅延により、16GBのモデルファイル(.safetensors)が0バイトに破損。

Solution (解決策):
Emergency_Remerge_V2.ipynbを作成。Colabローカルで一時保存・整合性チェック後にDriveへコピーする2段階保存を実装。

精度を最大化する多数決推論と安全機構

Majority Voting Mechanism



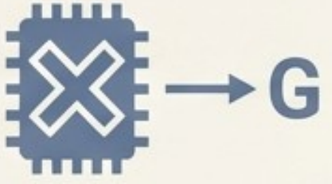
N=16回のサンプリング生成
(Temperature=0.5)

Result: この手法単体で精度が **+7% 大幅改善**

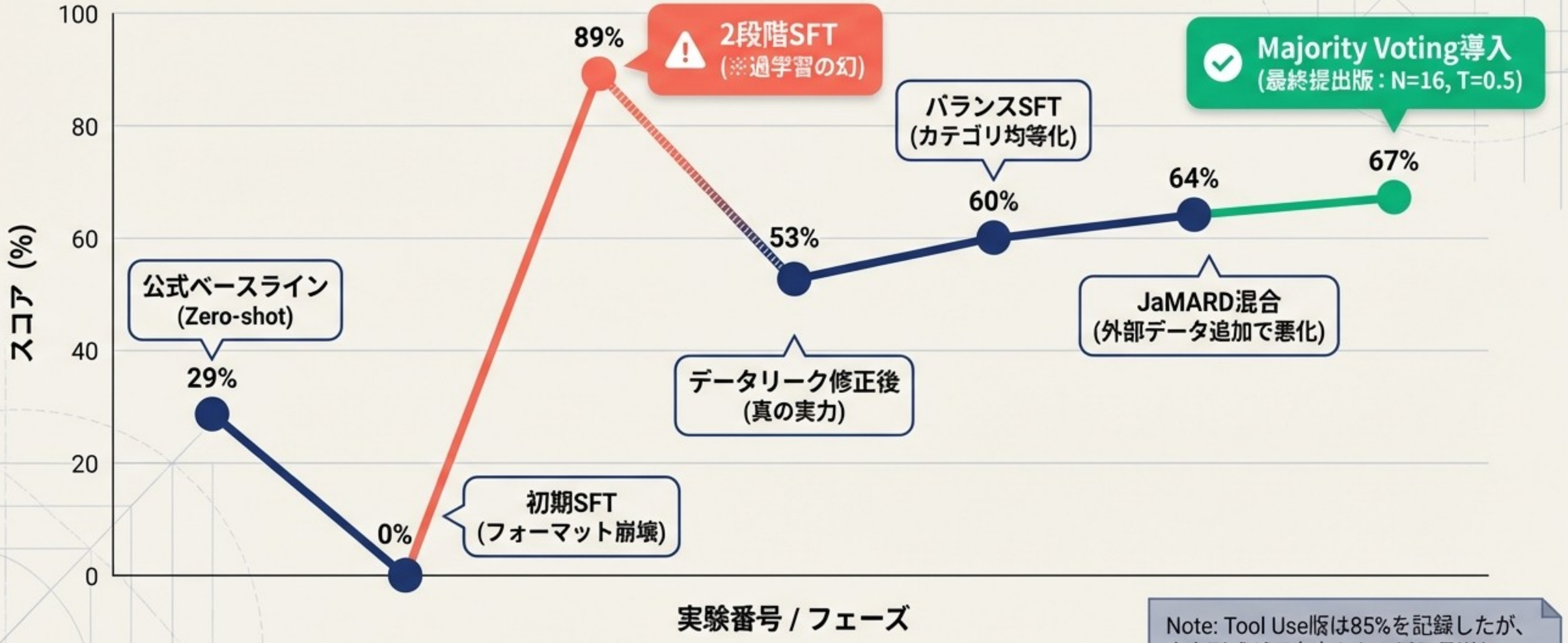
Safety Mechanism: OOM Fallback

OOM（メモリ不足）フォールバック

N回のサンプリング中にVRAMが枯渇した場合、自動的にGreedyデコーディング (1回生成)に切り替える堅牢なエラーハンドリング。

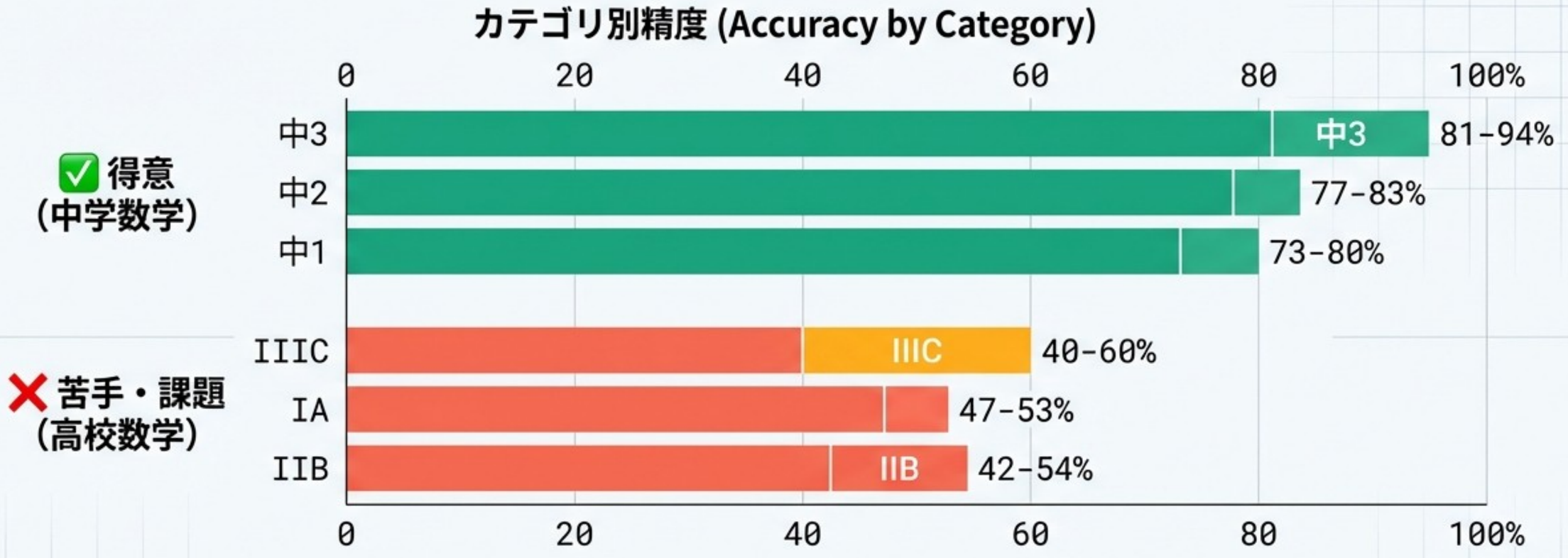


スコア推移と実験結果の全記録



Note: Tool Use版は85%を記録したが、
出力形式が不安定なため採用見送り。

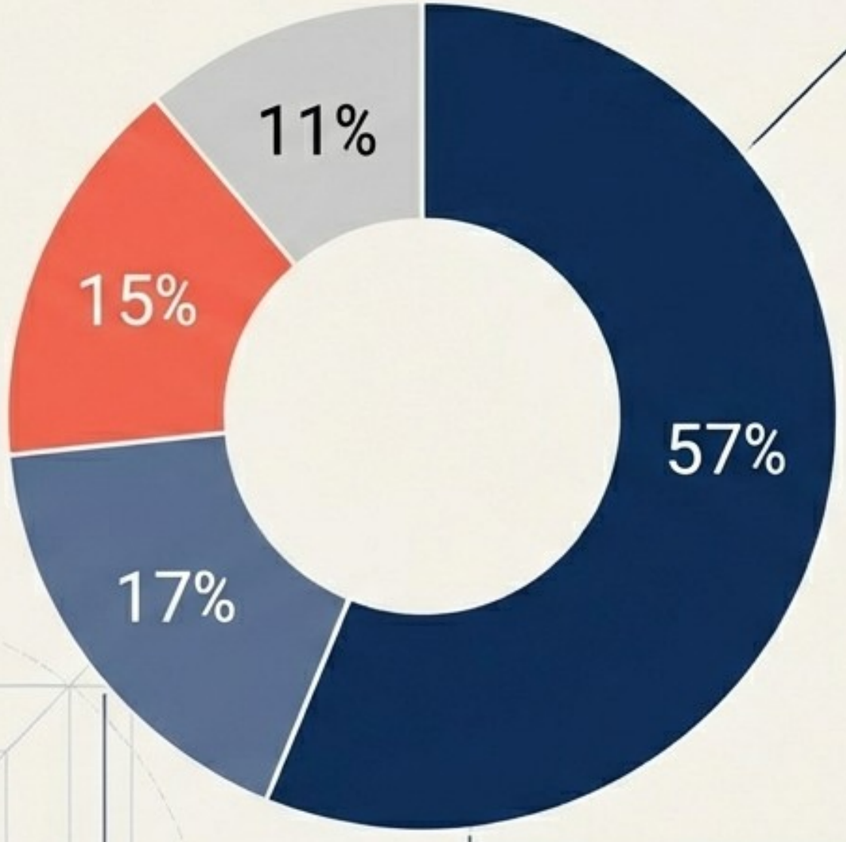
カテゴリ別精度評価：得意領域と弱点の可視化



Specific Weaknesses（弱点単元）

数列、指数・対数、三角関数、2次関数、数と式、整数の性質における推論力が今後の課題。

エラー分析：「なぜ間違えたのか？」



57% (27件) - 計算・論理エラー
途中計算の誤りや、問題文の誤解。

17% (8件) - LaTeXエスケープ問題
\\ の重複出力による文字列不一致。

15% (7件) - 条件解釈ミス
≤ と < の境界条件などの取り違え。

11% (5件) - フォーマット不一致
数学的には同値だが表記順序（因子の順序など）が異なる。

技術的総括：成功要因と失敗要因

🟢 成功 (What Worked)

推論と形式を完全分離した
2段階SFT。

LLMに頼らず品質を担保した
アルゴリズム合成データ。

推論のブレを吸収する
多数決推論（N=16）。

AI的表現（丁寧語等）を削る
厳格なデータクリーニング。

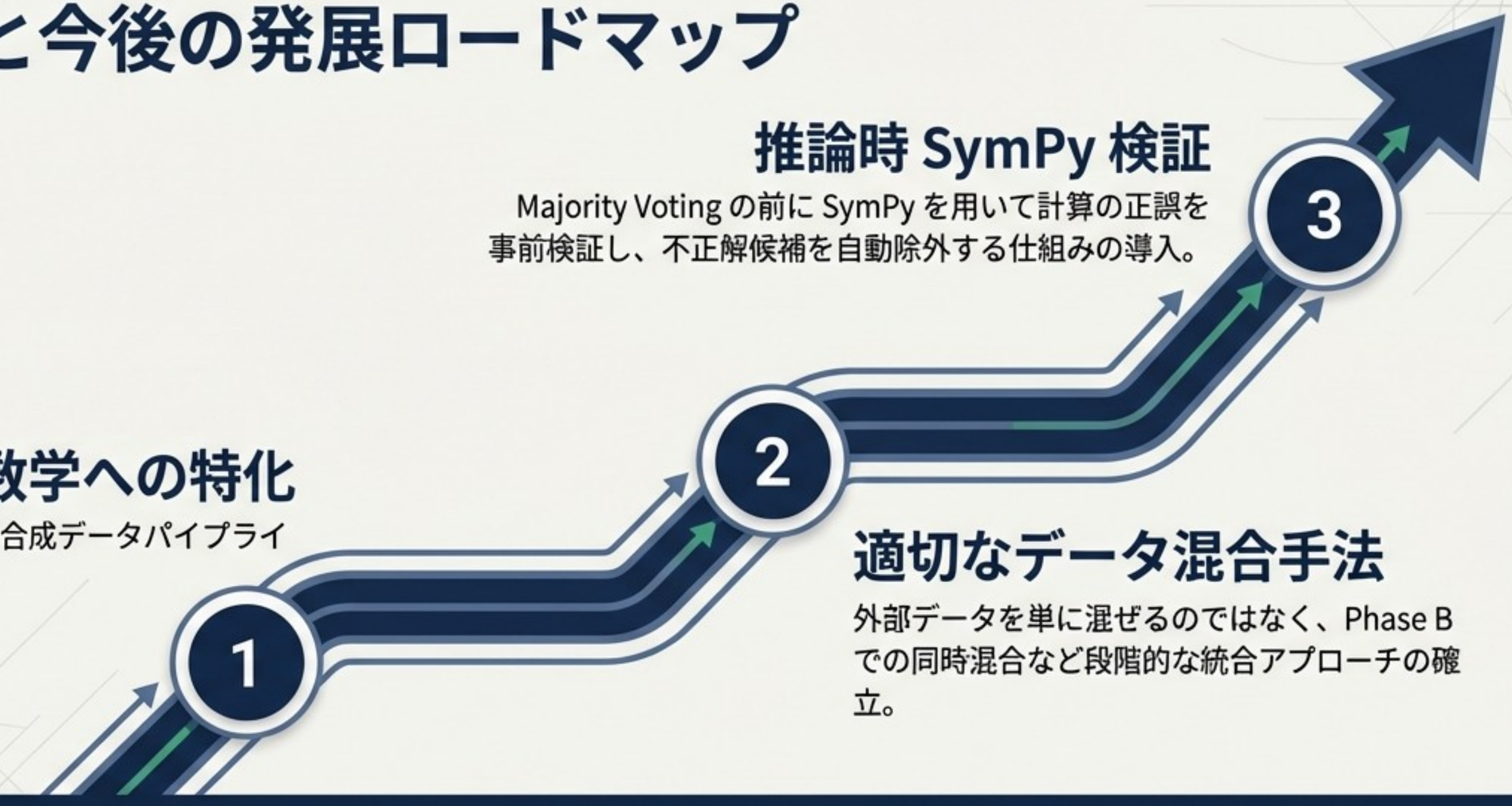
🔴 失敗 (What Didn't Work)

外部データの安易な混合:
JaMARD追加による破滅的忘却
（高校数学の知識破壊）。

Tool Use（計算機呼び出し）:
プロンプト感度が高すぎ、
出力形式が不安定化。

DPO: 効果なし（-1%低下）。
分布の偏りを引き起こすリスク。

結びと今後の発展ロードマップ



高校数学への特化

IA/IIB向け合成データパイプラインの拡充。

適切なデータ混合手法

外部データを単に混ぜるのではなく、Phase Bでの同時混合など段階的な統合アプローチの確立。

推論時 SymPy 検証

Majority Voting の前に SymPy を用いて計算の正誤を事前検証し、不正解候補を自動除外する仕組みの導入。

「品質の高いクリーンな合成データと、推論・形式を分離した2段階学習が、限られたリソース下でLLMの数学的性能を最大化する最適解である」