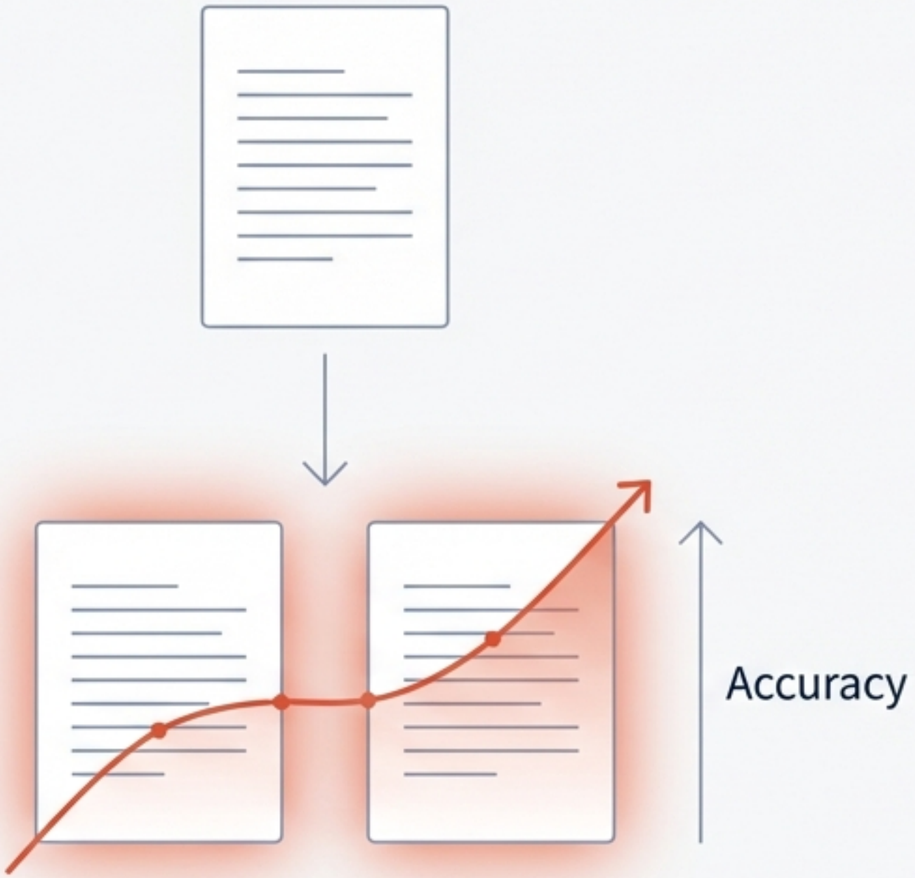
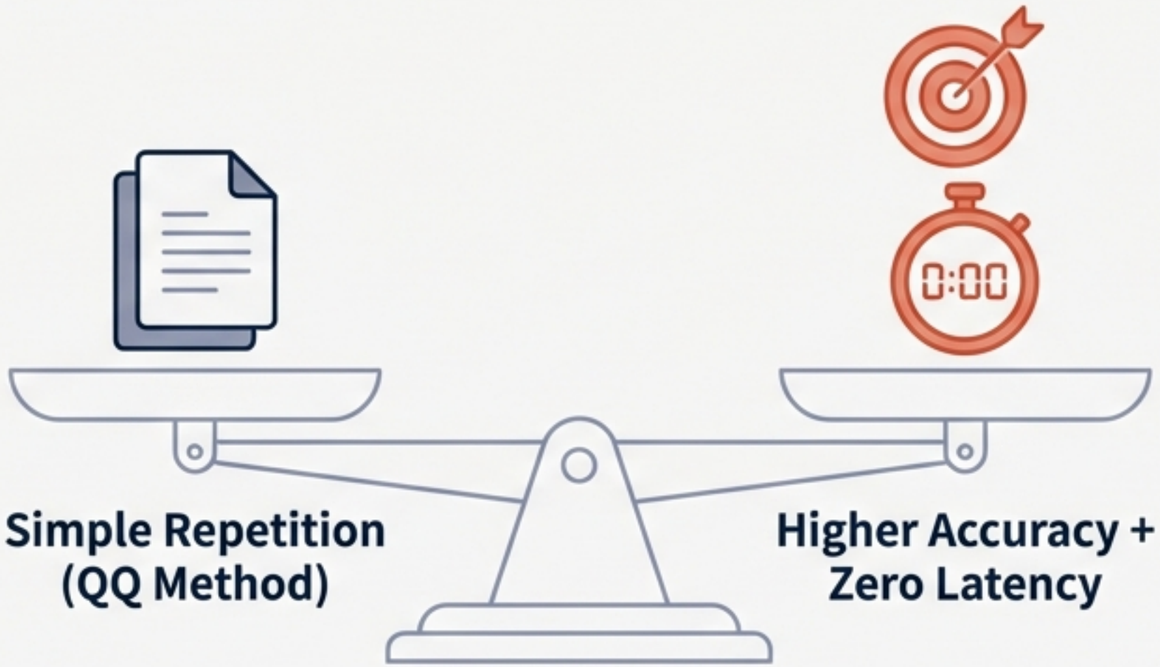
Prompt Repetition Improves LLM Accuracy

プロンプトを「2回繰り返す」だけで精度が向上するGoogle DeepMindの研究



Source: Prompt Repetition Improves Non-Reasoning LLMs (Google DeepMind, 2025)

精度向上、コスト増なし、遅延なし。
「無料のランチ」は実在した。



The Method

入力を言い換えたりせず、全く同じ内容を**2回連結**してモデルに与える。

The Result

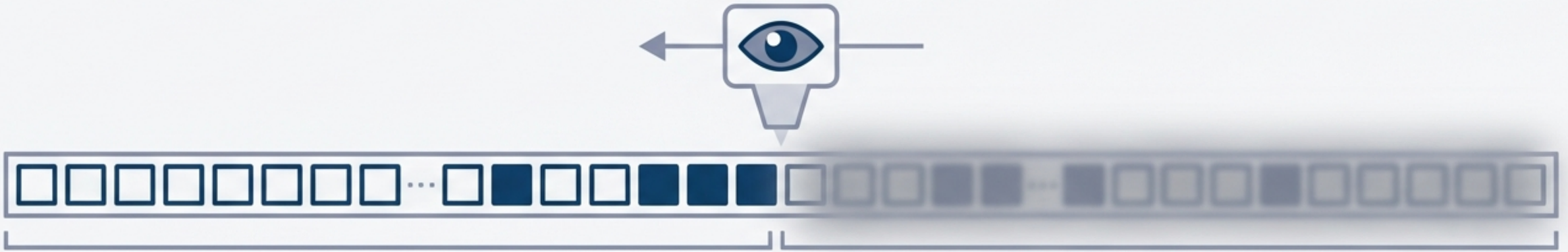
コンテキストの理解度、抽出精度、フォーマット遵守率が向上。
特定のタスク（NameIndex）では精度が**21%から97%へ向上**。

The Cost

応答速度（レイテンシ）への影響は**ほぼゼロ**。
出力トークン数は増加しない（回答が冗長にならない）。

LLMの弱点：Transformerの「片方向注意（Causal Attention）」

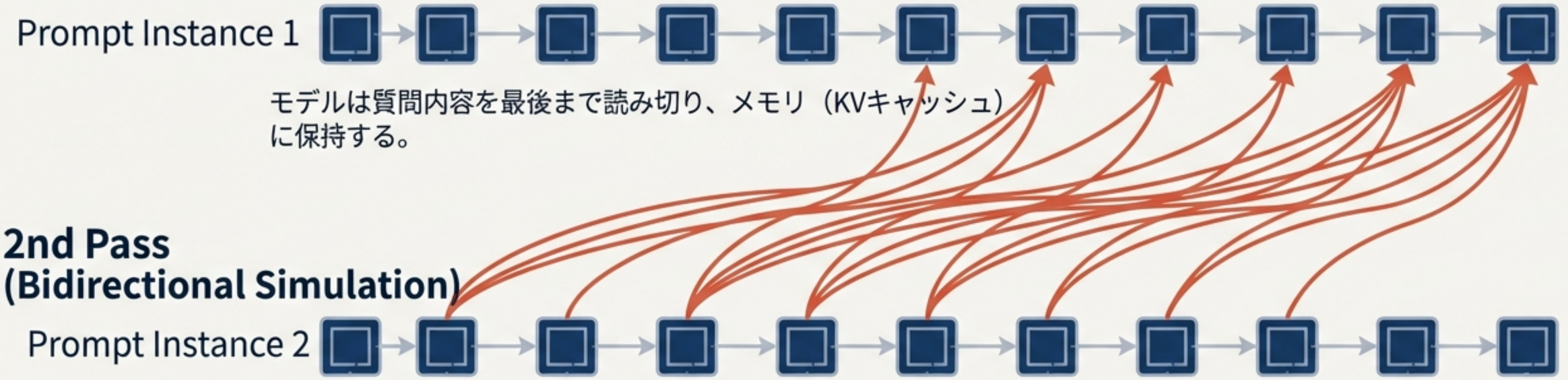
Transformerモデルは、左から右へ一方向にしかテキストを処理できない。
モデルがトークンAを読んでいるとき、それより後ろにあるトークンBの情報にはアクセスできない。



- Transformerモデルは、左から右へ一方向にしかテキストを処理できない。
- モデルがトークンAを読んでいるとき、それより後ろにあるトークンBの情報にはアクセスできない。
- 結果：文脈が**「後出し」**される場合（例：先に選択肢、後に質問）、重要性を判断できないまま読み進めてしまう。

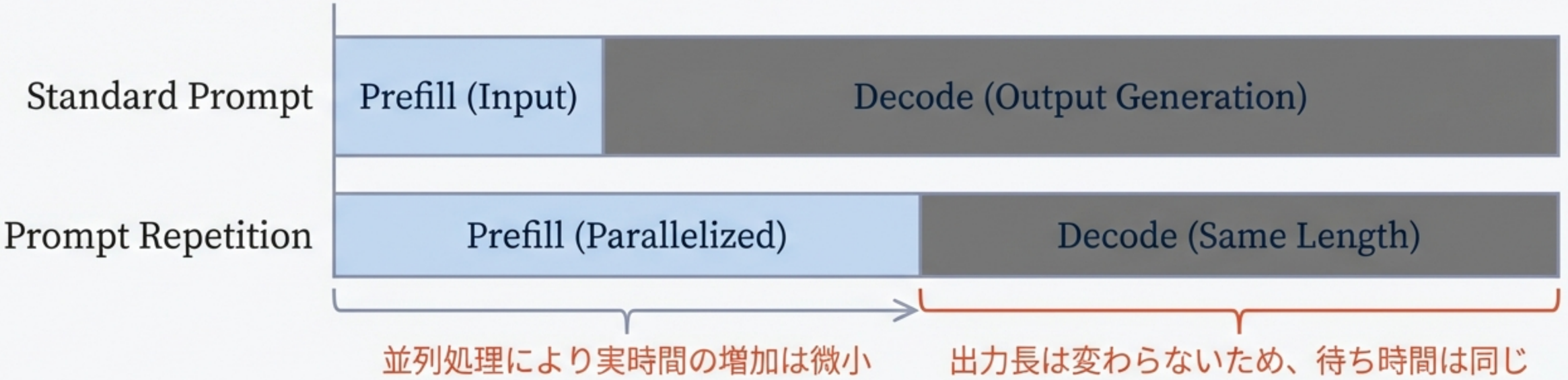
2回目は「全知」の状態で読む：擬似的な双方向注意

1st Pass (Linear Construction)



「因果的ブラインドスポット」が埋まり、
文中の曖昧さや見落としが解消される。

なぜ遅延（レイテンシ）が増えないのか？：プレフィルの並列化



LLMのボトルネックは出力生成（Decode）にある。プロンプト反復は入力処理（Prefill）のみを増やすが、これは並列化されるため、ユーザーが感じる応答速度（Wall-clock time）はほぼ変わらない。

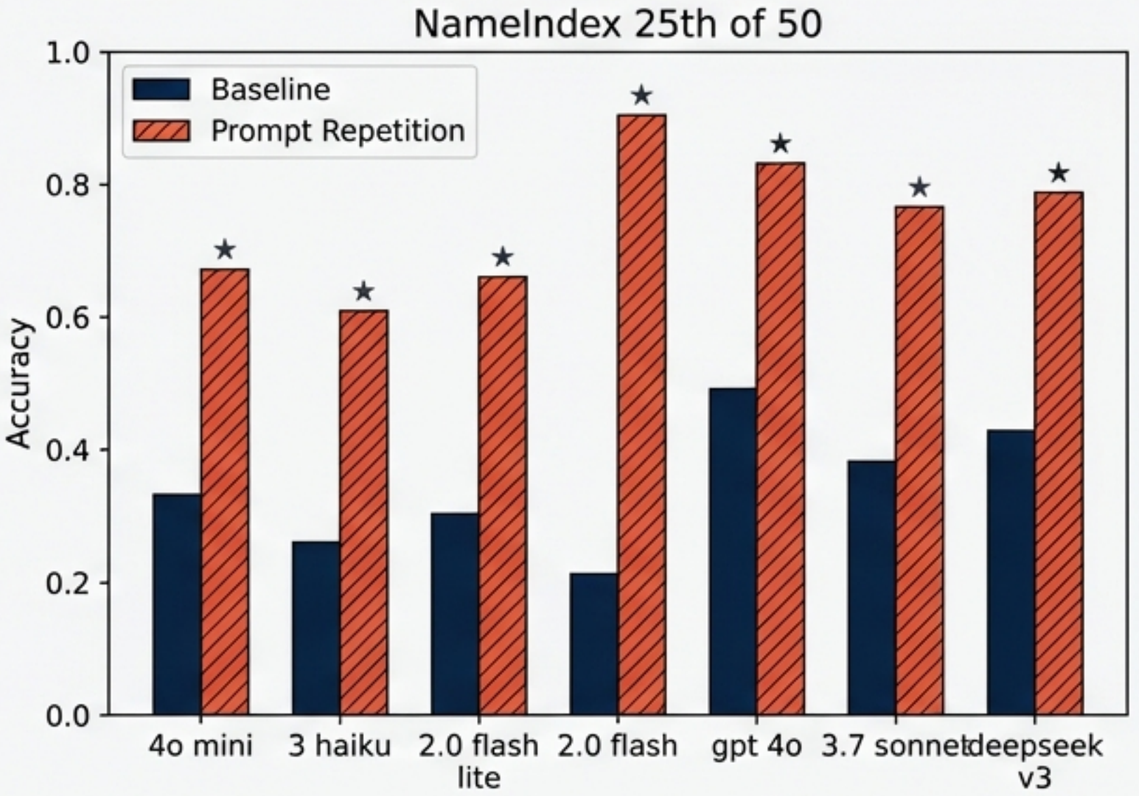
70通りの検証で47勝0敗：圧倒的な勝率

47 WINS

23 DRAWS

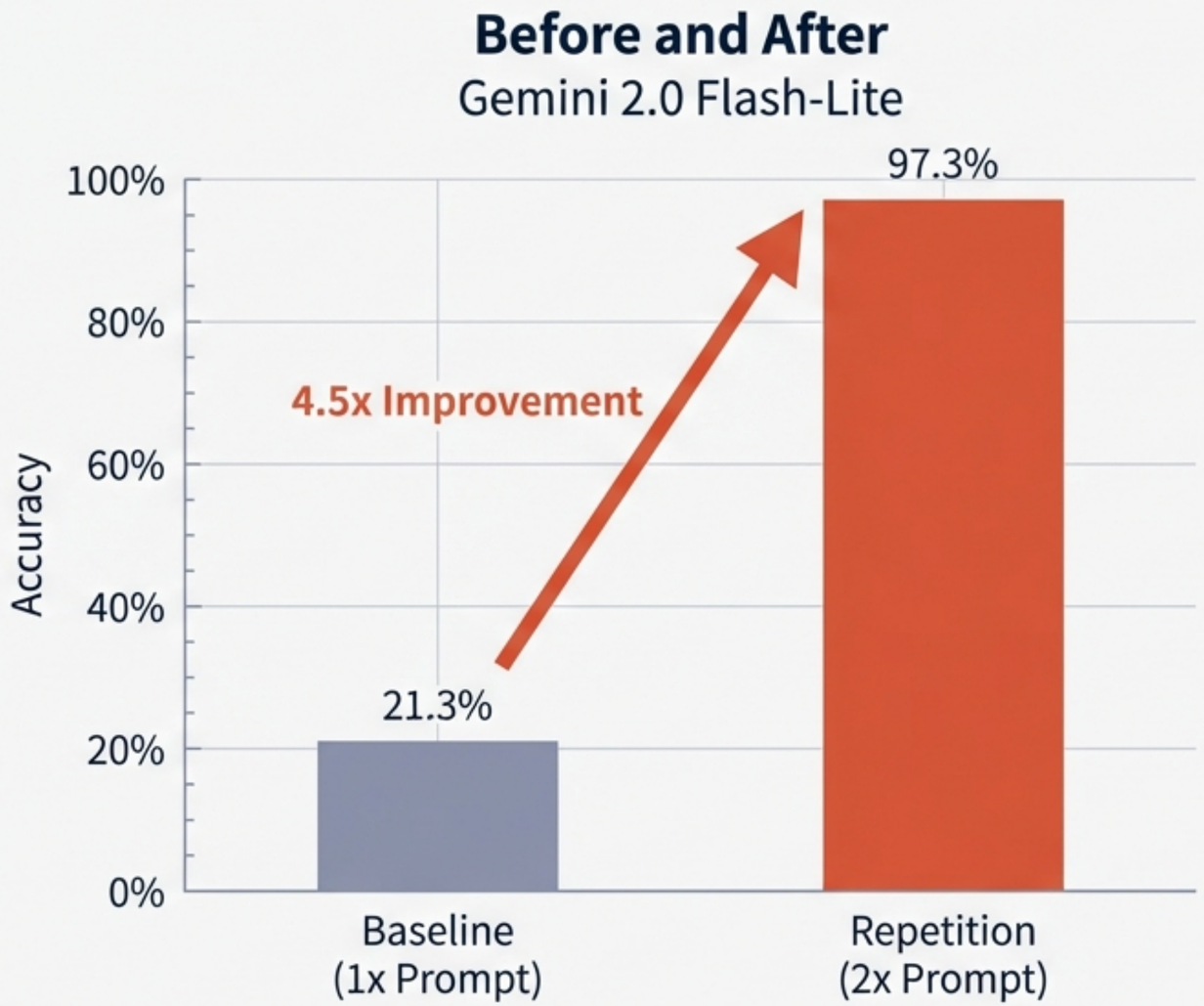
0 LOSSES

Statistically Significant Results (McNemar Test)



ARC, MMLU, GSM8Kなど7つの主要ベンチマーク、7つのモデル（Gemini 2.0, GPT-4o, Claude 3等）で検証。推論を強制しない設定において、この手法が劣化を引き起こすことはない。

「うっかりミス」が激減する：検索・抽出タスク（NameIndex）



Task: NameIndex

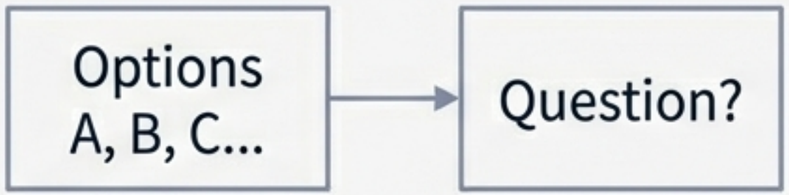
50個の名前リストから「25番目の名前」を特定するタスク。

1回の読み込みでは、モデルはカウンター（数）を保持しきれず、遠く離れた情報を正確に参照できない。

2回読むことで位置関係を正確に把握。
コンテキスト全体から特定情報を抽出するタスク（RAGなど）で最強の効果を発揮する。

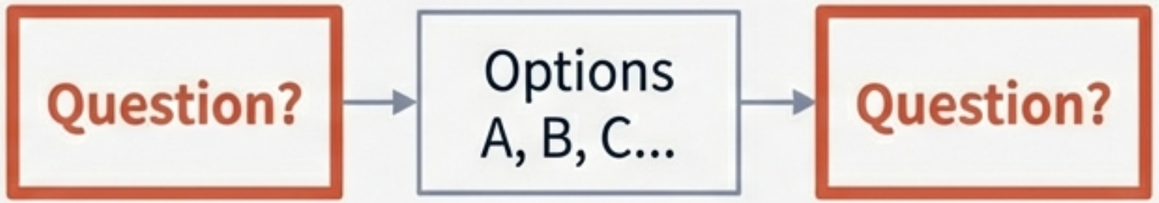
「選択肢が先か、質問が先か」順序バイアスの克服

A. Ordering Bias (Confusion)



モデルは質問を知らないまま選択肢を読むため、重要情報を捨ててしまう。

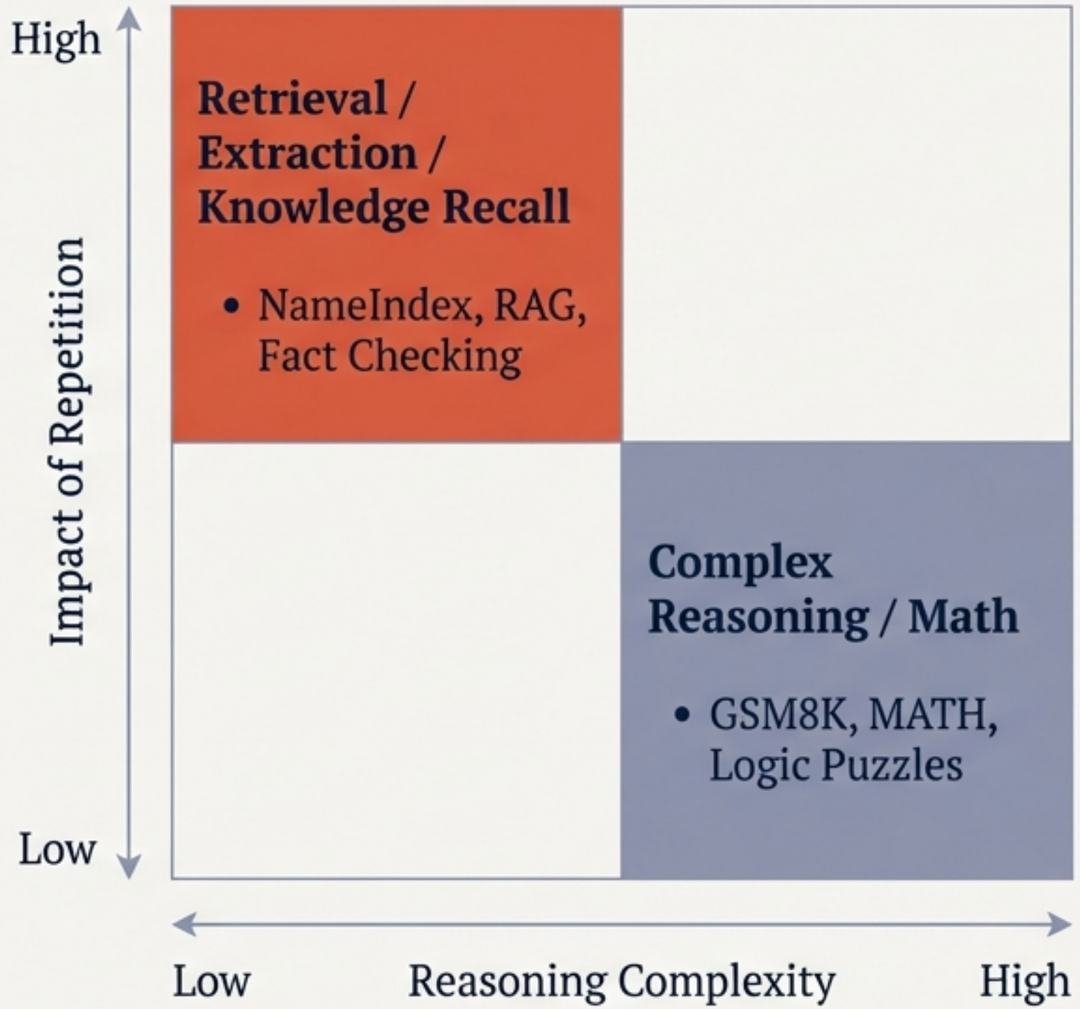
B. Repetition (Sandwich Structure)



最初の質問で意図を理解し、その文脈で選択肢を読み、最後の質問で回答を確定させる。

多くのベンチマークにおいて、質問より先に選択肢が来る「不親切な形式」でも、反復によって認知負荷を大幅に軽減できる。

限界：複雑な「推論（Reasoning）」は反復だけでは解けない



GSM8KやMATHのような多段階の論理的思考が必要なタスクでは、効果は限定的。

プロンプト反復は「理解」と「想起」を助けるが、モデルの「IQ（論理処理能力）」そのものを上げるわけではない。

解決策：推論タスクには、**CoT**（Chain-of-Thought）が必要。

モデルによる違い：「推論済み」モデルと「即答」モデル

RLHF / Chat Models (e.g., ChatGPT)

```
Let me rephrase
your question...
```

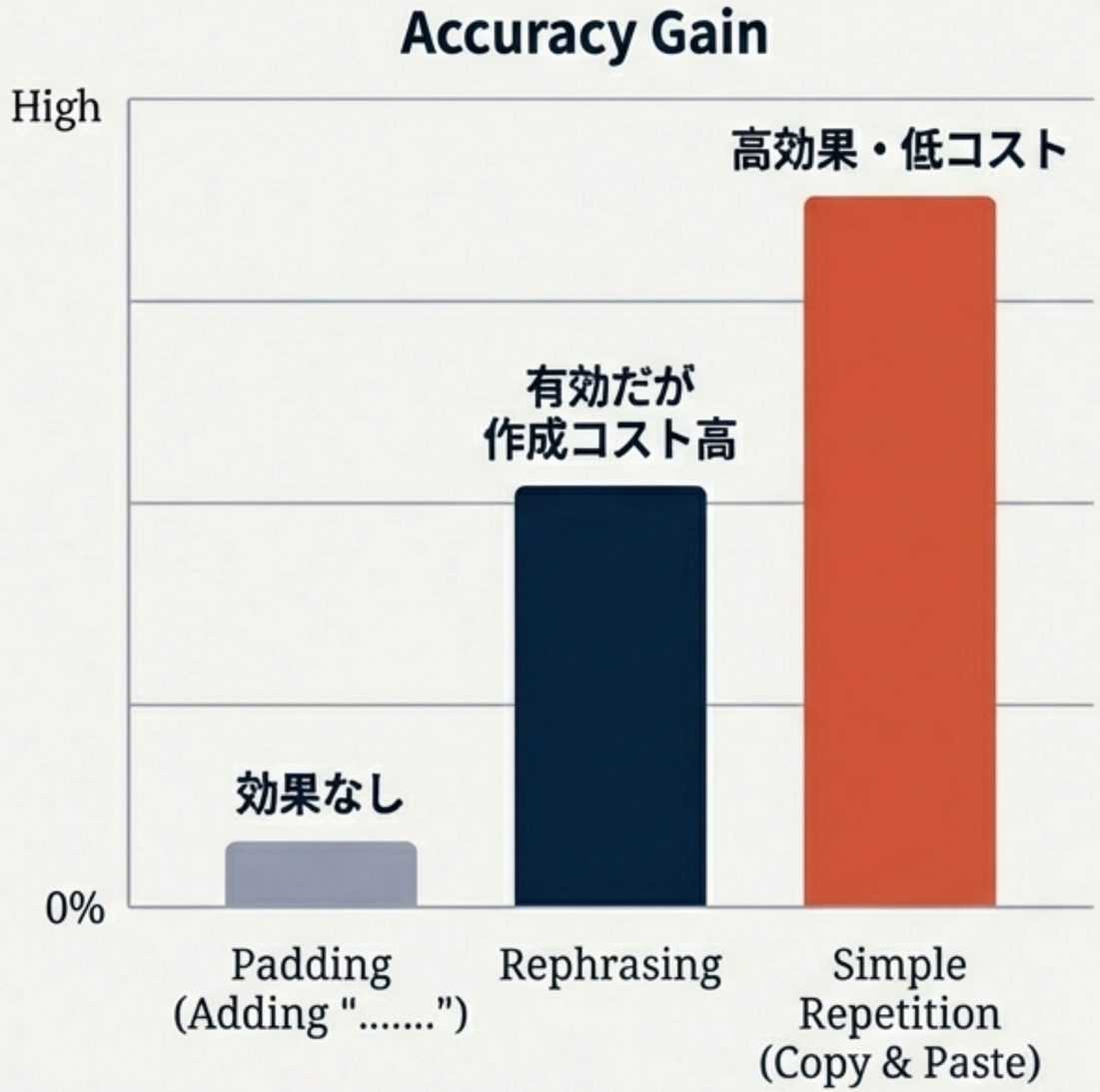
内部で既に「自己反復」や「言い換え」を行う癖がついている。明示的な反復の効果は薄い。

Base / Direct Models (e.g., Raw API)

「推論なし」で即答するモデル。反復プロンプトが不足している認知プロセスを補うため、**効果が最大化**する。

Best Use Case:
「Don’t think, just answer」のような直接的な指示、またはベースモデルの使用時。

単に長くすれば良いわけではない：パディングとの比較



プロンプトを「.」などで水増して長くしても、精度は全く向上しなかった。

重要なのは「長さ」ではなく**「情報の反復提示」**である。

極端な検索タスクでは「**3回反復**」がさら極端な検索タスクでは「**3回反復**」がさらに有効な場合もあるが、通常は**2回で十分**。

次世代アーキテクチャへの示唆：自動的な「読み直し」

Architectural Insight:

本研究はTransformerの「片方向注意」の限界を浮き彫りにした。

Future Direction:

将来のモデルは、内部的に入力を自動で再読する機構や、双方向的な処理ステップをアーキテクチャレベルで組み込む可能性がある。

「Quality = Input Quality x Contextual Attention」

今日から使えるプロンプトエンジニアリング

When to Use (推奨)	✓ RAG（検索拡張生成）におけるコンテキスト抽出 ✓ 長いドキュメントからの特定データ抽出 ✓ 選択肢が多い分類・クイズ問題
When NOT to Use (非推奨)	✗ 数学、論理パズル（CoTを使うべき）
Implementation (実装)	`prompt = query + "\n" + query` 複雑なツール統合は不要。単純な文字列連結のみ。

Simple is Strong.

複雑なシステム統合やファインチューニングの前に、
まずは「コピー&ペースト」を試すべきである。

Based on: ‘Prompt Repetition Improves Non-Reasoning LLMs’ (Google DeepMind, 2025)