

Chain of Thought (CoT) の発展手法

複雑なタスクを解くための4つのアプローチ

4つの手法：概要比較まとめ

手法名	イメージ（例え）	手間	性能	特徴
Complex-CoT	「超詳しい手本」 を見せて真似させる	大 (手書き作成)	高	思考過程まで含めた詳細な例題を与える。
Zero-shot-CoT	「魔法の言葉」 を唱えるだけ	小 (一言のみ)	中～ 高	例題なし。「ステップバイステップで」と指示。
Auto-CoT	「AIが手本を自作」 教科書作りを自動化	中 (自動化)	高	手本作成のコストを削減しつつ性能を維持。
Plan-and-Solve	「まず計画、次に実行」 手順書を作らせる	小	高	タスクを分解し、計画を立ててから解く。

1. Complex-CoT

「より多くの推論ステップを持つfew-shotプロンプト」

仕組み

- 従来の「質問と答え（Q&A）」だけの例題ではなく、**「どうやってその答えを出したか（理由・計算式）」**まで詳細に書かれた例題をAIに見せる。

初心者向けイメージ

- **「めちゃくちゃ丁寧な途中式が書いてある参考書」**を渡す。
- AIは「あ、これくらい詳しく考えなきゃいけないんだ」と察して、複雑な手順を飛ばさずに真似するようになる。

2. Zero-shot-CoT

「Let's think step by step. (ステップ・バイ・ステップで考えて)」

仕組み

- 例題 (Few-shot) は一切与えない。
- その代わりに、プロンプトの最後に**魔法の一言**を付け加えるだけで、AIの推論モードを強制的に起動させる。

初心者向けイメージ

- パニックになっている生徒 (AI) に**「深呼吸して、順番に一つずつ考えてごらん？」**と声をかける。
- たったこれだけで、当てずっぽうの回答が減り、論理的な思考が始まる。

3. Auto-CoT

「クラスタリングで分類し、推論チェーンを自動生成」

仕組み

- Complex-CoTは性能が良いが、人間が「詳しい手本」を作るのが大変。
- そこで、問題を分類（クラスタリング）し、AI自身にZero-shotで「手本（解説）」を作らせ、それをまた次の推論に利用する。

初心者向けイメージ

- 先生（人間）が解説を作るのをサボり、**「成績優秀な生徒（AI）に模範解答を作らせて、それをみんなの教科書として使う」**自動化システム。

4. Plan-and-Solve

「タスクを分解し、詳細な指示を与える」

仕組み

- いきなり「解いて」とは言わず、2段階で指示する。
 1. **Plan:** 具体的なサブタスク（やることリスト）や計画を立てさせる。
 2. **Solve:** その計画に従って実行させる。

初心者向けイメージ

- 料理をする前に、いきなり包丁を持たせず**「まずはレシピを読んで手順を書き出してください。それから料理して」**と指示する。
- 計算ミスや手順飛ばし（ハルシネーション）を防ぐ効果が高い。

【重要】なぜ思考プロセスを出力するのか？

自己回帰モデル（Autoregressive Model）の宿命

AIは**「自分が出力した文字」**を見て、次の思考を決定している。

- **出力しない（暗算）**
 - メモリに途中経過が残らないため、複雑な計算ができない。
- **出力する（筆算）**
 - **「書き出した思考」**が次の計算の足場（ヒント）になる。

“ **結論：**

「ステップ・バイ・ステップで」と指示するときは、**途中経過も画面に出力させないと、AIは賢くなれない。**

”