

Fixing Large Language Models' Specification Misunderstanding for Better Code Generation

Zhao Tian and Junjie Chen and Xiangyu Zhang

lcse2025

背景と課題

コード生成の現状

- 大規模言語モデルの登場によりコード生成の技術は飛躍的に向上した
- しかし、ロジックが複雑になると極端に精度が落ちる。
(APPSにおける完全な成功率は約16%)
- ファインチューニングなど、精度を上げるために様々な手法が採用されている。

研究の目的

- 高精度のコード生成をプラグインプレイで利用できるようにする。

先行研究

思考喚起プロンプティング (Thought-Eliciting Prompting)

- 中間の思考プロセスや推論ステップを出力させることによって、論理的な出力をさせることを図る.
- CoTやStructured CoTなど様々な手法がある.
- 難しい問題には対応できず、一度間違った方向に進むと修正がきかない.

先行研究

フィードバックベースのプロンプティング

- 一度コードを生成したうえでテストを実行し、そのエラーを用いて改善する方法.
- 直接入力する方法や、一度説明させたうえで入力する方法などがある.
- 最初の生成があまりにも離れていると改善できない.

アプローチ

主な方針

- それぞれ効果はあるものの課題点もあるとされてきた二つの手法を組み合わせることによって、相乗効果による革新を目指す。

独自性

- これまでは生成結果について行っていたフィードバックを、LLMの理解に対して行う。

μ FiX

1. コード生成前

2. コード生成後

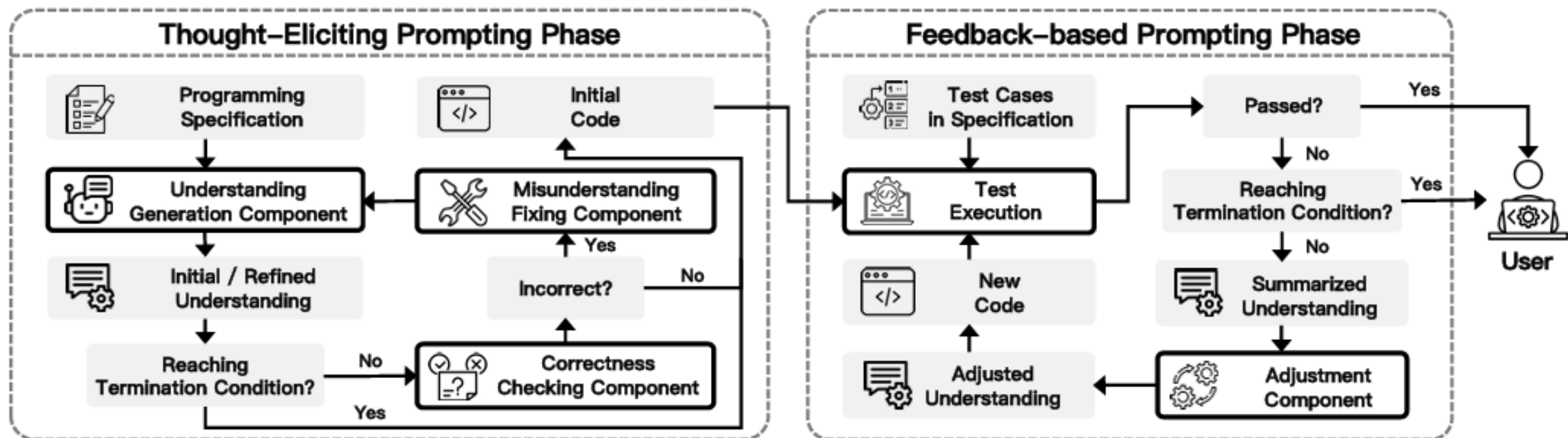


Fig. 2. Overview of μ FiX

μ FiX

1. コード生成前

- プログラムの仕様とテストケースを与え、使用をどのように理解したかをLLMに出力させる.
- その出力をテストケースと照らし合わせ、矛盾がなくなるまで修正する.

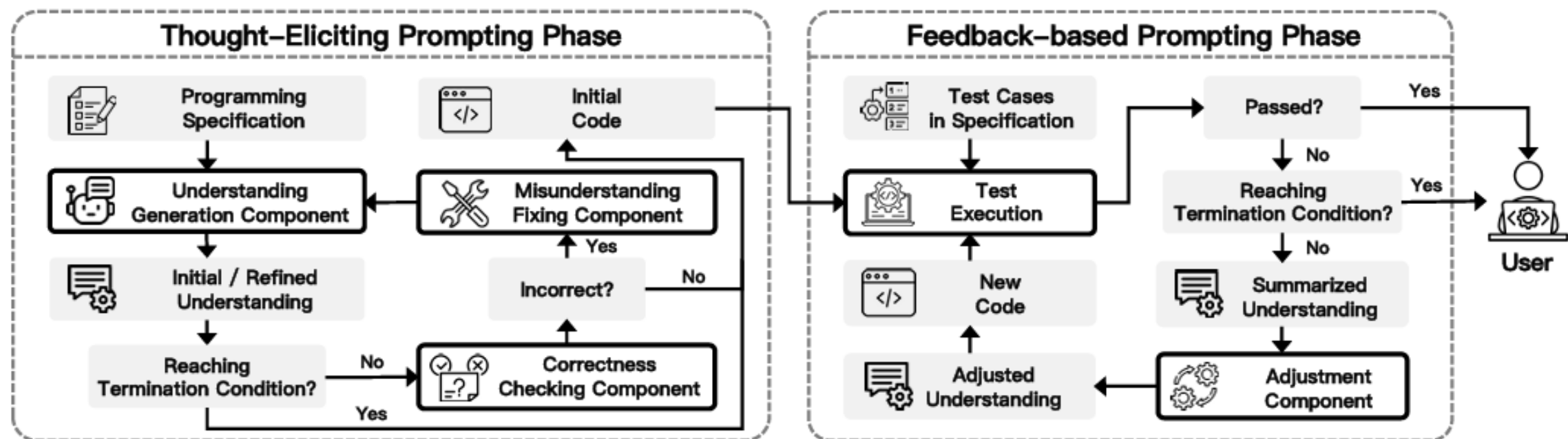


Fig. 2. Overview of μ FiX

μ FiX

2. コード生成後

- コードを出力させた後, そのコードをLLMに与えて要約させる
- ステップ1の出力と見比べさせることでずれを認識したうえで再び出力させる.

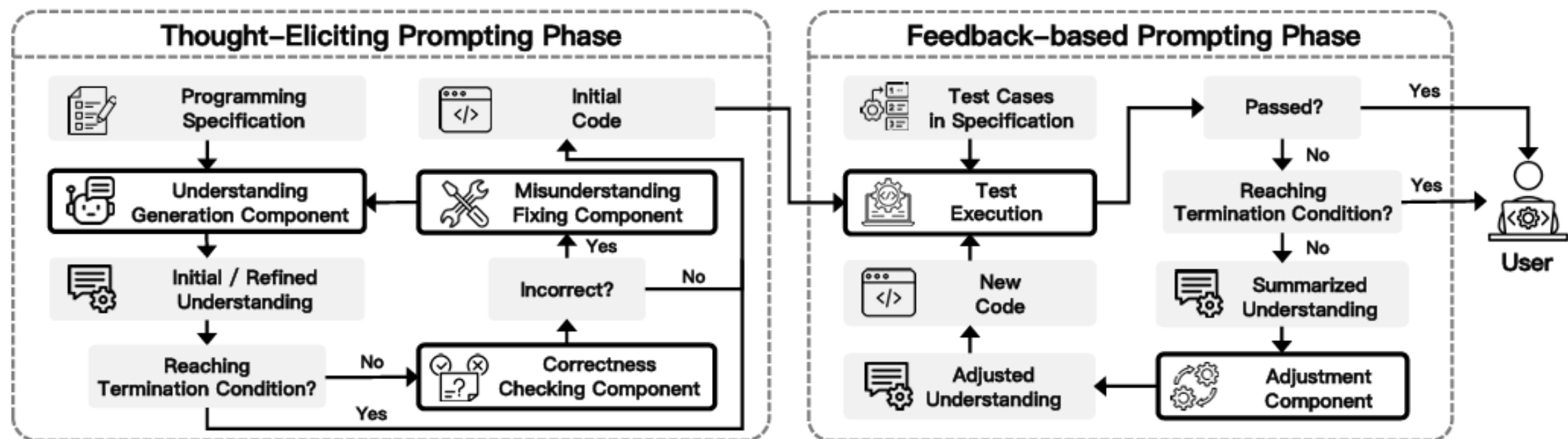


Fig. 2. Overview of μ FiX

評価設計

RQ1: 最先端の技術と比較してどうか

RQ2: 各コンポーネントの有効性

RQ3: ハイパーパラメータの設定による変化

データセット

- HumanEval, HumanEval+, APPS, HumanEval-ET, APPS-ET, BPP-ET から抽出した2066件のケース

LLM

- ChatGPT (GPT-3.5), DeepSeek-Coder

RQ1: 最先端の技術と比較してどうか

TABLE I
EFFECTIVENESS COMPARISON ON CHATGPT IN TERMS OF PASS@1 (↑) AND AVGPASSRATIO (↑). APR IS SHORT FOR AVGPASSRATIO.

ChatGPT Prompting Technique	HumanEval		HumanEval+		HumanEval-ET		MBPP-ET		APPS		APPS-ET	
	Pass@1	APR	Pass@1	APR	Pass@1	APR	Pass@1	APR	Pass@1	APR	Pass@1	APR
Zero-shot	73.78%	79.14%	66.46%	68.66%	67.07%	82.69%	58.21%	71.94%	16.33%	27.98%	6.00%	23.93%
Few-shot	75.61%	79.77%	69.51%	70.81%	68.29%	83.04%	59.34%	73.34%	17.33%	24.45%	6.33%	20.37%
Zero-shot-CoT	74.39%	79.25%	67.68%	69.69%	67.07%	83.21%	59.14%	72.91%	18.00%	30.02%	6.00%	24.97%
Few-shot-CoT	76.83%	81.62%	70.73%	72.03%	68.90%	84.41%	59.34%	73.22%	20.33%	32.65%	7.00%	27.60%
Self-planning	78.66%	81.95%	70.73%	72.36%	70.73%	83.67%	62.01%	75.22%	21.33%	33.48%	8.33%	28.97%
SCoT	79.27%	83.97%	71.95%	73.66%	71.34%	85.54%	62.22%	77.36%	22.00%	34.07%	7.67%	29.56%
Self-Debugging	77.44%	81.52%	71.34%	72.94%	70.12%	84.18%	60.37%	75.69%	18.67%	31.83%	6.33%	27.36%
Self-Edit	76.83%	81.15%	69.51%	70.82%	68.29%	84.34%	61.19%	76.35%	20.67%	33.28%	7.00%	27.49%
Self-repair	80.49%	82.61%	74.39%	75.65%	72.56%	81.25%	64.07%	75.25%	23.00%	30.21%	8.00%	25.19%
Self-planning+Self-Debugging	80.49%	83.33%	72.56%	74.20%	72.56%	85.66%	62.01%	75.40%	21.33%	33.67%	8.33%	29.23%
Self-planning+Self-Edit	79.88%	83.25%	71.34%	72.92%	71.34%	84.10%	62.01%	75.31%	21.33%	33.81%	8.33%	29.36%
Self-planning+Self-repair	82.32%	84.83%	73.78%	75.70%	73.78%	86.56%	64.17%	75.55%	23.33%	37.19%	8.67%	31.65%
SCoT+Self-Debugging	81.10%	85.50%	73.17%	74.90%	71.95%	86.83%	63.14%	78.45%	22.00%	34.16%	7.67%	29.77%
SCoT+Self-Edit	79.88%	84.27%	72.56%	74.31%	71.95%	86.12%	63.96%	78.58%	22.33%	34.43%	7.67%	29.78%
SCoT+Self-repair	81.71%	84.48%	75.00%	76.28%	73.17%	84.66%	64.37%	79.21%	25.00%	38.38%	8.67%	33.05%
μ FiX	90.24%	91.59%	80.49%	82.16%	79.88%	91.66%	69.10%	83.07%	35.67%	47.72%	10.33%	38.81%

TABLE II
EFFECTIVENESS COMPARISON ON DEEPSEEK-CODER IN TERMS OF PASS@1 (↑) AND AVGPASSRATIO (↑). APR IS SHORT FOR AVGPASSRATIO.

DeepSeek-Coder Prompting Technique	HumanEval		HumanEval+		HumanEval-ET		MBPP-ET		APPS		APPS-ET	
	Pass@1	APR	Pass@1	APR	Pass@1	APR	Pass@1	APR	Pass@1	APR	Pass@1	APR
Zero-shot	76.22%	80.92%	72.56%	73.79%	69.51%	82.97%	55.75%	68.75%	4.00%	11.38%	1.00%	10.95%
Few-shot	78.05%	82.41%	73.17%	74.91%	70.73%	84.28%	56.78%	69.86%	4.33%	6.28%	1.33%	4.81%
Zero-shot-CoT	76.83%	80.86%	72.56%	73.80%	69.51%	82.68%	55.85%	68.78%	4.67%	11.77%	2.00%	10.83%
Few-shot-CoT	78.66%	82.48%	74.39%	75.65%	71.34%	83.49%	57.08%	70.55%	4.67%	10.53%	1.67%	10.29%
Self-planning	78.65%	82.11%	73.17%	74.40%	70.73%	83.45%	57.08%	70.05%	4.00%	8.92%	1.00%	7.64%
SCoT	78.05%	82.03%	73.78%	75.04%	70.73%	83.48%	58.01%	71.16%	4.33%	10.21%	1.33%	9.29%
Self-Debugging	77.44%	82.45%	73.17%	74.78%	70.73%	83.83%	56.88%	69.73%	4.67%	12.11%	1.33%	11.60%
Self-Edit	76.83%	81.97%	73.17%	74.40%	70.12%	84.32%	57.08%	71.52%	4.67%	12.22%	1.67%	11.70%
Self-repair	79.27%	83.08%	72.56%	74.55%	70.12%	85.34%	58.73%	73.52%	5.67%	12.68%	1.33%	10.41%
Self-planning+Self-Debugging	79.27%	82.72%	73.78%	75.01%	71.34%	84.55%	59.34%	72.50%	5.33%	11.05%	1.67%	9.76%
Self-planning+Self-Edit	79.88%	83.33%	74.39%	75.62%	71.95%	84.67%	58.83%	71.96%	5.67%	13.14%	1.67%	12.46%
Self-planning+Self-repair	79.88%	83.10%	74.39%	75.62%	71.95%	84.19%	59.34%	73.84%	6.33%	13.91%	1.33%	11.35%
SCoT+Self-Debugging	79.27%	83.25%	74.39%	76.26%	71.95%	84.70%	58.21%	71.81%	5.67%	11.87%	1.67%	10.88%
SCoT+Self-Edit	78.66%	82.64%	74.39%	75.65%	70.73%	83.91%	58.21%	71.74%	4.67%	11.45%	1.67%	10.56%
SCoT+Self-repair	80.49%	84.15%	73.78%	75.77%	71.34%	86.10%	59.45%	72.94%	6.67%	15.47%	1.67%	15.00%
μ FiX	83.54%	86.17%	78.66%	80.22%	75.00%	87.12%	63.35%	78.03%	14.00%	23.59%	5.00%	19.81%

RQ2: 各コンポーネントの有効性

アブレーション

- $\mu\text{FiX}_{\text{woF}}$: 後半のフェーズを削除したもの
- $\mu\text{FiX}_{\text{SR}}$: 後半の手法を最先端のSelf-repairに置き換えたもの
- $\mu\text{FiX}_{\text{SCoT}}$: 前半のフェーズを、最先端のSCoTに置き換えたもの
- $\mu\text{FiX}_{\text{woS}}$: 前半のフェーズを削除したもの

TABLE III
COMPARISON BETWEEN μFiX AND ITS VARIANTS IN TERMS OF PASS@1 ($\uparrow$) AND AVGPASSRATIO ($\uparrow$)

Variant	HumanEval		HumanEval+		HumanEval-ET		MBPP-ET		APPS		APPS-ET	
	Pass@1	APR	Pass@1	APR	Pass@1	APR	Pass@1	APR	Pass@1	APR	Pass@1	APR
$\mu\text{FiX}_{\text{woF}}$	85.37%	87.33%	76.83%	79.11%	77.44%	88.60%	66.32%	80.12%	27.67%	38.63%	9.00%	32.75%
$\mu\text{FiX}_{\text{SR}}$	85.98%	87.94%	78.05%	79.67%	78.05%	89.54%	68.07%	82.38%	30.67%	42.80%	10.00%	35.14%
$\mu\text{FiX}_{\text{SCoT}}$	83.54%	87.16%	76.22%	77.87%	74.39%	87.83%	65.81%	80.20%	26.33%	37.72%	9.33%	32.50%
$\mu\text{FiX}_{\text{woS}}$	81.10%	84.30%	73.78%	75.09%	72.56%	85.31%	64.78%	79.36%	24.33%	40.24%	8.67%	33.58%
μFiX	90.24%	91.59%	80.49%	82.16%	79.88%	91.66%	69.10%	83.07%	35.67%	47.72%	10.33%	38.81%

RQ3: ハイパーパラメータの設定による変化

パラメータ

- N （前半の修正回数）, M （後半の修正回数）, T （温度）

デフォルト

- $N=1, M=1, T=0.7$

結果

- 回数を増やせば性能は向上するものの、コストに見合わない

TABLE IV
INFLUENCE OF HYPER-PARAMETERS N AND M IN TERMS OF PASS@1 ($\uparrow$) AND AVGPASSRATIO ($\uparrow$)

LLM	Configuration	HumanEval		HumanEval+		HumanEval-ET		MBPP-ET		APPS		APPS-ET	
		Pass@1	APR	Pass@1	APR	Pass@1	APR	Pass@1	APR	Pass@1	APR	Pass@1	APR
ChatGPT	$\mu\text{FiX}_{N=1, M=1}$	90.24%	91.59%	80.49%	82.16%	79.88%	91.66%	69.10%	83.07%	35.67%	47.72%	10.33%	38.81%
	$\mu\text{FiX}_{N=2, M=2}$	90.85%	92.88%	81.10%	82.74%	80.49%	92.04%	69.61%	83.32%	39.00%	49.81%	11.33%	40.43%
DeepSeek-Coder	$\mu\text{FiX}_{N=1, M=1}$	83.54%	86.17%	78.66%	80.22%	75.00%	87.12%	63.35%	78.03%	14.00%	23.59%	5.00%	19.81%
	$\mu\text{FiX}_{N=2, M=2}$	84.76%	86.98%	79.88%	81.44%	76.22%	88.35%	65.71%	79.59%	16.33%	27.31%	5.33%	22.84%

まとめと所感

既存手法を組み合わせたハイブリット

- いかにLLMを自由に働かせないかに注力

予備実験

- 前半をGPTに、後半をDeepSeekで行うと、DeepSeekより少し性能が向上したという記述
- 意外と実践しているのは初めて見た