[S4-P11] ローカルLLMを用いたAIエージェントの現状と課題

竹下 理斗[1], 川田 拓朗[2], 大橋 巧[2], 北田 俊輔[2], 彌冨 仁[1,2]
[1]法政大学 理工学部, [2]法政大学大学院 理工学研究科

■ Summary

ローカルLLM の AIエージェント性能評価

- GUI 操作タスク (ブラウザ操作・画像編集など) をローカルLLM に実行させ成功率を評価
- モデル規模・アーキテクチャごとに性能を比較
- 失敗分析をもとにプロンプトを改善し再評価

■ Background

AIエージェントとは

- 目標に沿い自律思考しタスクを実行・評価する技術
- 強化学習 → LLM へ (追加学習なしで利用可能に)

クローズドLLM vs ローカルLLM

vs

- 高性能
- 高コスト
- 外部依存
- セキュリティ懸念
- ブラックボックス

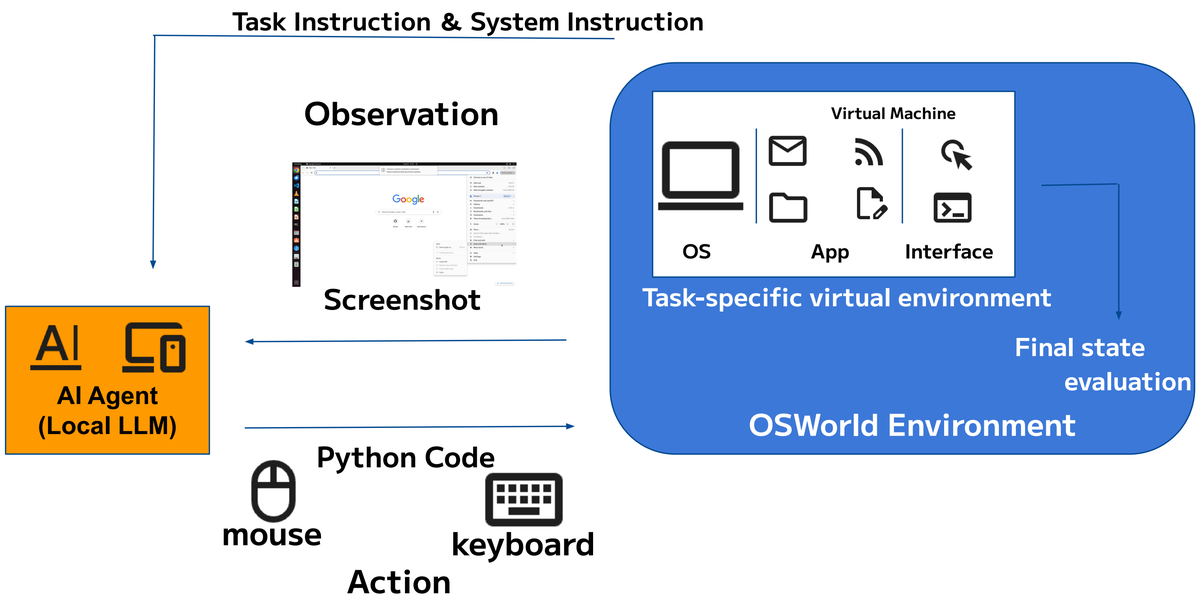
- 軽量でも動作
- 低コスト
- 内部依存
- データ安全
- 調整が容易

😢 **ローカルLLM のエージェント性能評価はまだ不十分**

■ Method

OSWorld [Xie+,NeurIPS'24] を用いたローカルLLM 評価

- GUI 操作ベンチマーク環境
- VM 上でブラウザ操作等を含む 369 タスク
- 観測方法：Screenshot 、a11y_tree など
- 評価：実行スクリプトに基づき成功か判定



失敗分析に基づく最小プロンプト修正

- 行動ログをもとに典型的な失敗パターンを抽出
- GPT を用いてプロンプト修正・再評価

失敗と対策

- 無限ループ → 同一操作の連続禁止・戦略切替
- 座標依存 → テキスト主導操作へ切替
- ポップアップ遮断 → まずオーバーレイ処理
- 空振り操作 → 短い待機を挿入
- 早発 DONE/FAIL → 宣言は条件を満たした場合のみ

など計 13 個の対策

■ Experiments & Results

実験環境

評価環境：OSWorld (Ubuntu VM、360/369 タスク)

- 各タスクは同一 VM スナップショットから開始
- 解像度 1920×1080 の Screenshot を観測

推論環境：ローカルLLM (OpenAI 互換 API 経由で通信)

- モデル入力：Screenshot + プロンプト
- モデル出力：Python コード (スクリーン上の操作を模倣)

実験結果 (モデル別成功率、steps=15)

Model	Params	Success Rate (%)	Unintended Success Rate (%)
Qwen-2.5-VL-3B	~3B	0.28	0.00
Qwen-2.5-VL-7B	~7B	2.78	0.56
Qwen-2.5-VL-32B	~32B	11.67	0.28
DeepSeek-VL2-small	~2.8B	0.28	0.28
InternVL2-8B	~8B	0.56	0.56
Llama 3.2-Vision	~11B	0.83	0.28
LLaVA-v1.6-Mistral-7B	~7B	0.00	0.00
Simular Agent S2 (Mar 13, 2025)	-	27.00	-
Human	-	72.40	-

😆 **同一アーキテクチャでは規模により性能向上**
😢 **アーキテクチャが異なる場合比例せず**

失敗分析に基づくプロンプト修正前後の比較

Qwen-2.5-VL-32Bの平均成功率 (各2回) を比較

Prompts	Average Success Rate (%)
Original	10.4
Fixed	10.7

😆 **プロンプト修正はエージェントとしての行動に確実に作用し、行動パターンの質は上昇**
😢 **「座標頼みの探索」などモデル固有の弱点は残り、成功率の底上げには直結せず**

■ Conclusion & Future Work

ローカルLLMエージェント性能

- 同一アーキテクチャでは規模により性能向上
- 異なるアーキテクチャ間では規模と性能は比例せず
  → アーキテクチャの適合性が重要
- 成功率は低く、依然としてクローズドモデルに劣る

Future Work

Self-correction による誤動作修正

- 行動の誤りを検出し、戦略を切り替える仕組みを導入

Screenshot より成功率が高い a11y_tree へ入力切替

- ローカルLLM は入力制限により利用不可
- 圧縮／抽出でローカルLLM へ適用させ、効果を検証