DEEP LEARNING JP
[DL Papers]

Navigating the Digital World as Humans Do: Universal Visual Grounding for GUI Agents

Yuya IMAI, Matsuo Iwasawa Lab

書誌情報

【ICLR'25 Oral】 Navigating the Digital World as Humans Do: Universal Visual Grounding for GUI Agents

Authors: Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, Yu Su
Affiliation: The Ohio State University, Orby AI

TL;DR 視覚情報(スクリーンショット)のみを観測として、GUI操作を行うAIエージェントの構築

背景

GUIエージェント

- GUI操作をしてデジタル世界を行動する自律エージェント
- LLMの発展によって、webやデスクトップ、モバイルなど多様な環境でエージェントが動作可能になった
- 多くの既存手法はHTMLやアクセシビリティ(a11y)ツリーなどのテキストベースの観測を利用している

背景

テキストベース情報依存の問題点

- ノイズや不完全性
  - HTML全体には不要な情報が大量に含まれている
  - a11yツリーは省略があったり誤ったアノテーションが含まれる場合が多い
- レイテンシとコスト増大
  - HTMLなどの情報を毎ステップ取得・解析するコストが大きい

→人間のように、視覚的な観測のみを行い、ピクセルレベルの操作をするエージェントは、どこまで実現可能なのか？

背景

課題: グラウンディング

- 視覚情報のみのエージェント実現における主要なボトルネックはグラウンディング
  - テキストベースの計画（例：「送信ボタンをクリック」）を、GUI上の正確な位置（座標）に対応付けるプロセス
- グラウンディングモデルに必要な要件:
  - 高精度: 一度のグラウンディングエラーがタスク全体の失敗につながる可能性があるため、精度が非常に重要
  - 強い汎化性: デスクトップ、モバイル、Webなど、異なる種類のGUIで機能する必要がある
  - 柔軟性: 特定のプランナーに強く依存せず、様々なモデルと組み合わせて使える必要がある

背景

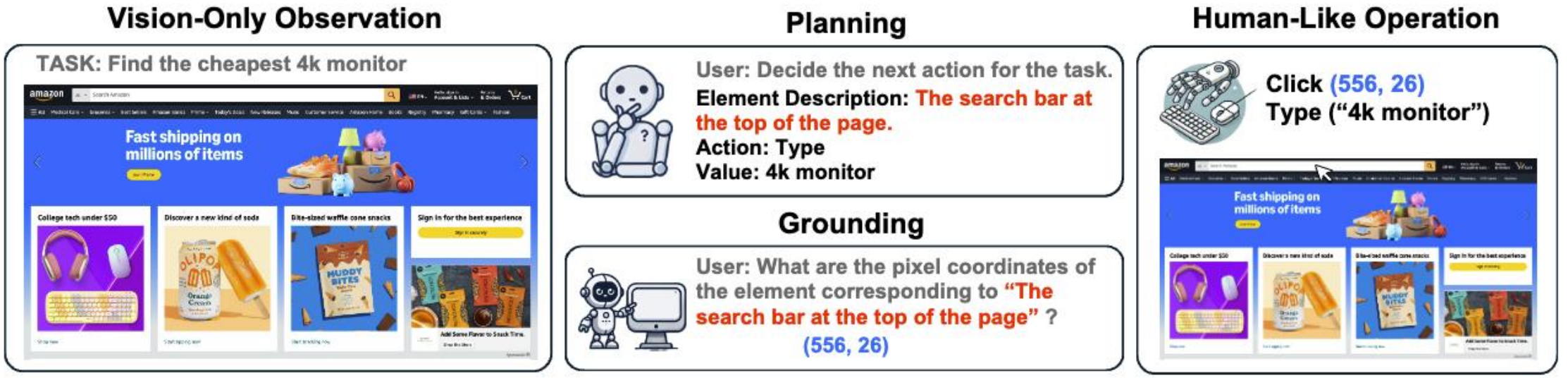
本研究の貢献

- GUIエージェントフレームワーク（SeeAct-V）の提唱
  - 人間のように視覚情報のみで環境を認識し、ピクセルレベルで操作するGUIエージェントのための汎用的なフレームワークSeeAct-Vを提案
- 大規模データセットとグラウンディングモデル（UGround）の公開
  - 過去最大規模のGUI視覚グラウンディングデータセット（130万枚のGUIスクリーンショットから1000万のGUI要素とその指示表現）を構築、公開
  - このデータセットで、普遍的な視覚グラウンディングモデルUGroundを開発、公開
- 包括的な評価
  - グラウンディング精度、オフラインエージェント評価、オンラインエージェント評価の3カテゴリにわたる6つのベンチマークを用いた、包括的なGUIエージェントの評価を実施

手法

SeeAct-Vフレームワーク

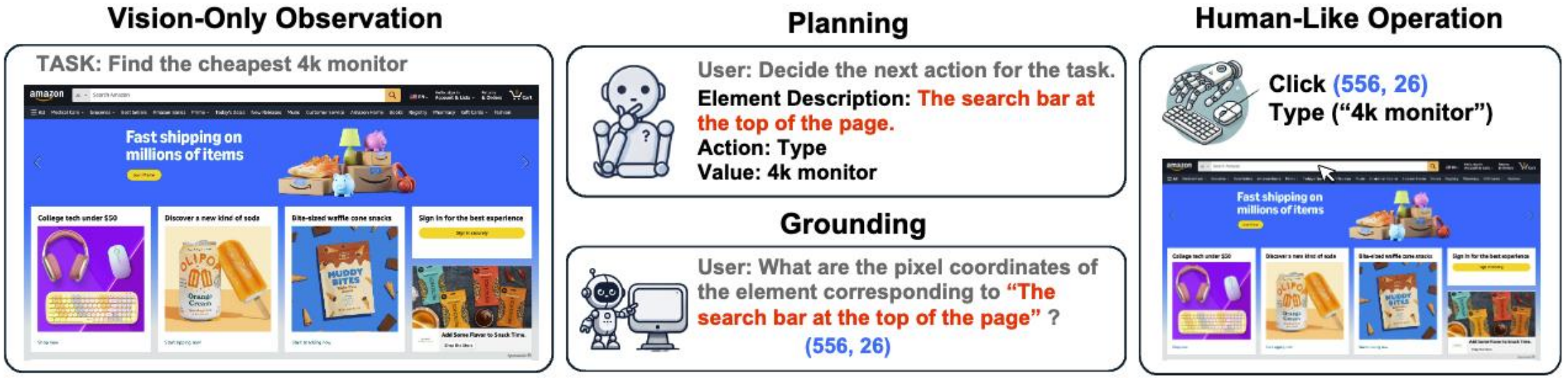
- 前提
  - スクリーンショットのみを環境観測として使用
  - タスク指示はテキストとして入力



手法

SeeAct-Vフレームワーク

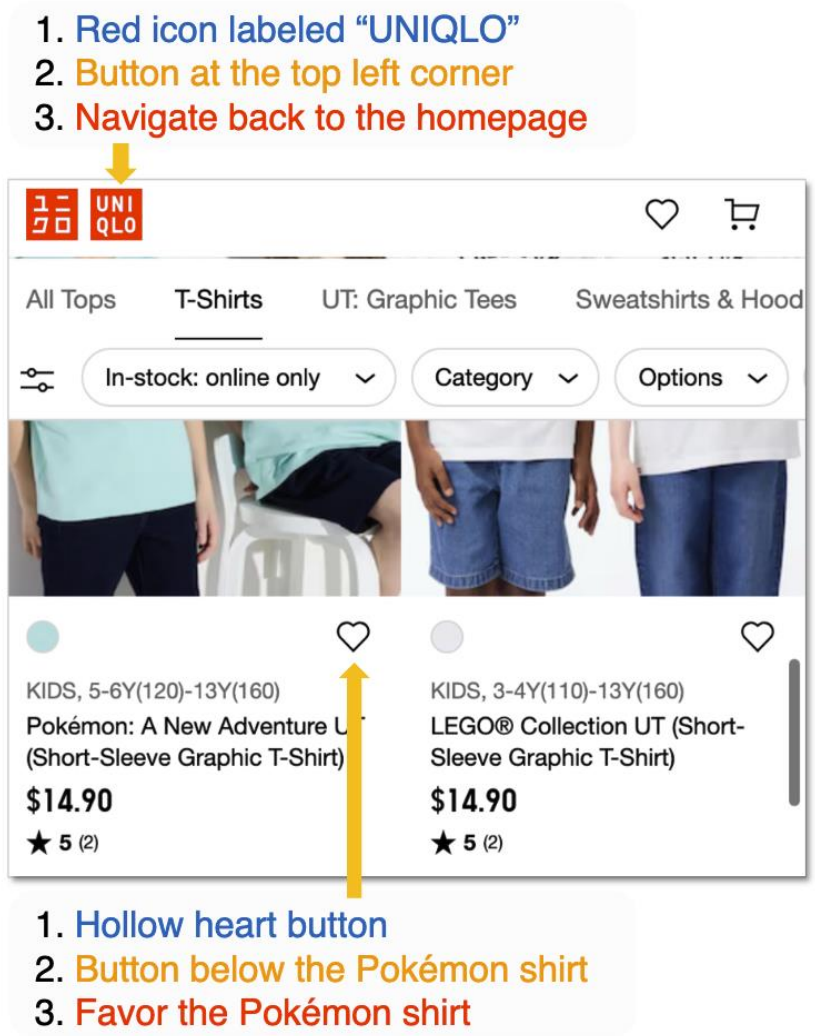
- 2つの主要コンポーネントで行動
  - Planning: 計画(テキスト)を生成する
  - Grounding: 計画をスクリーンショット上の座標に変換する



手法

データ構築

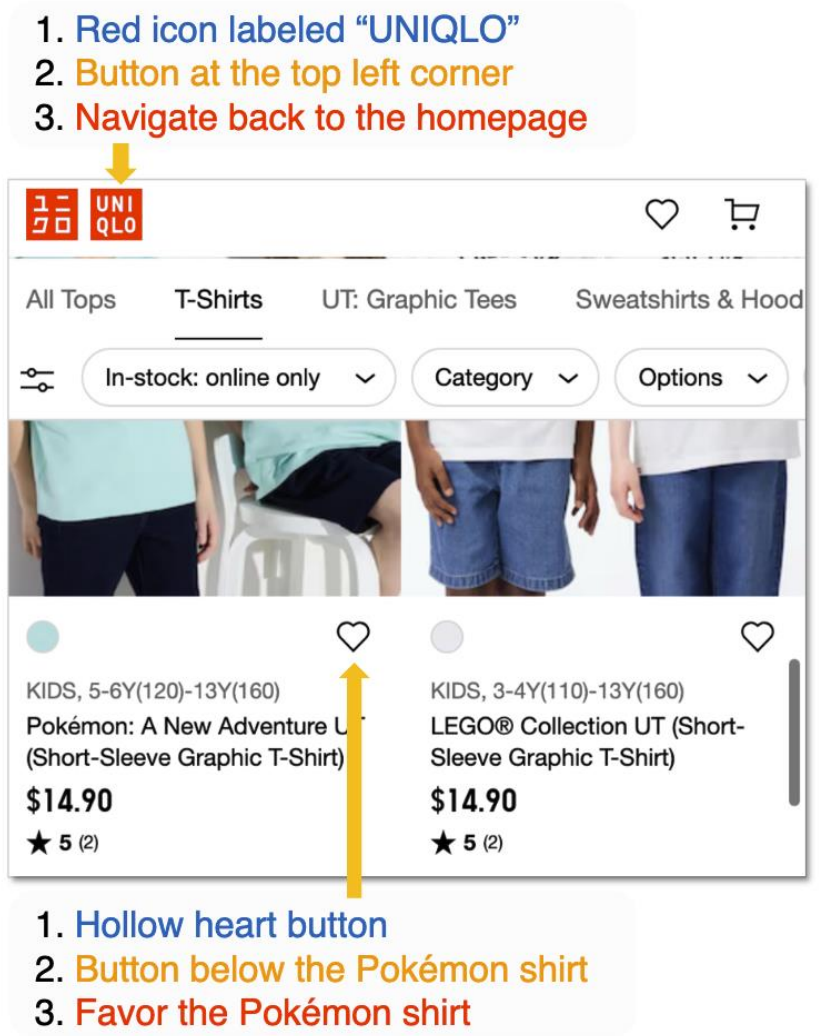
- 視覚グラウンディング用の大規模データセットをウェブから合成
  - (スクリーンショット, 参照表現, 座標)の組
- ウェブページを利用する利点
  - HTMLからバウンディングボックスや詳細な位置情報を容易に取得可能
  - CSSやアクセシビリティ属性など豊富なメタデータが活用できる



手法

データ構築

- 複数種類の参照表現の利用
  - 視覚的参照表現（Visual RE）
    - テキストや画像内容などの視覚的特徴
    - ボタンや入力フィールドなどの要素タイプ
    - 形状、色などの外観特性
  - 位置的参照表現（Positional RE）
    - 絶対位置：「ページの左上」など
    - 相対位置：「要素Xの右側」など
    - 文脈的参照：「アイテムAのための」「セクションXの下の」など
  - 機能的参照表現（Functional RE）
    - 主な機能による参照：「ホームに移動」「カートに追加」など



手法

データ構築

- 参照表現生成方法
  - HTMLから抽出(ルールベース+LLMによる洗練)
    - 視覚的表現：inner-text、altなどから要素の見た目を記述
    - 機能的表現：aria-labelなどのアクセシビリティ属性から機能を把握
    - 位置的表現：「ページの上部に」、「要素AとBの間にある」など
  - マルチモーダルLLMによる拡張
    - LLMの知識による拡張解釈（「青い鳥のアイコン→Twitterのアイコン」など）

手法

データセット

- 主要データセット「Web-Hybrid」
  - Common Crawlからの大量のスクリーンショットとメタデータ収集
  - 縦向き・横向きの多様な解像度のスクリーンショット
- 補助データセット
  - 既存のAndroidグラウンディングデータを統合
  - 「Web-Direct」：GPT-4oによる直接合成データ
    - 用途：ウェブに少ない特殊なGUI要素（トグルボタンなど）のカバー

Dataset	Annotation	# of Elements	# of Screenshots	Platform
Web-Hybrid (Ours)	Rule + LLM	9M	773K	Web
Web-Direct (Ours)	GPT	408K	408K	Web
GUIAct (Chen et al., 2024)	GPT + Human	140K	13K	Web
AndroidControl (Li et al., 2024b)	Human	47K	47K	Android
Widget Caption (Li et al., 2020b)	Human	41K	15K	Android
UIBert (Bai et al., 2021)	Human	16K	5K	Android
AITZ (Zhang et al., 2024c)	GPT + Human	8K	8K	Android
Total		10M	1.3M	Web + Android

手法

モデル設計

- オープンソースのVLM、LLaVA-NeXT (7B)をバックボーンに採用
- テキスト入力：スクリーンショット上で特定の要素を参照する言葉
  - 「スクリーンショットの中で、『{要素の説明}』に対応するピクセル座標はどこですか？」という形式
- 画像入力：柔軟に画像を分割できるCLIP@224pxをエンコーダーとして使用
- 出力：座標(x, y)を自然言語形式で出力するように調整
  - 例：「(1344, 1344)」

→前述のデータセットで学習してUGroundモデルを構築

実験

概要

- 6つのベンチマークを使用
  - 3つの主要プラットフォーム（ウェブ、デスクトップ、モバイル）をカバー
- 3つの評価設定
  - 視覚グラウンディング
  - オフラインエージェント評価（キャッシュされた環境）
  - オンラインエージェント評価（ライブ環境）

実験

視覚グラウンディング評価

- ScreenSpotベンチマーク（1272件の指示と対応するバウンディングボックス）
  - 標準設定：人間のアノテーターによる機能的説明・指示に従う
    - “set an alarm for 7:40”
  - エージェント設定：MLLMによって生成された多様な参照表現に従う

実験

視覚グラウンディング評価

- 結果
  - UGroundがすべての設定とプラットフォームで既存モデルを上回る
  - 標準設定で平均20%、エージェント設定で29%の改善

Grounding Model	Mobile		Desktop		Web		Average
	Text	Icon/Widget	Text	Icon/Widget	Text	Icon/Widget	
GPT-4	22.6	24.5	20.2	11.8	9.2	8.8	16.2
GPT-4o	20.2	24.9	21.1	23.6	12.2	7.8	18.3
CogAgent (Hong et al., 2024)	67.0	24.0	74.2	20.0	70.4	28.6	47.4
SeeClick (Cheng et al., 2024)	78.0	52.0	72.2	30.0	55.7	32.5	53.4
UGround (Ours)	**82.8**	**60.3**	**82.5**	**63.6**	**80.4**	**70.4**	**73.3**

Planner	Grounding	Mobile		Desktop		Web		Avg.
		Text	Icon/Widget	Text	Icon/Widget	Text	Icon/Widget	
GPT-4	SeeClick	76.6	55.5	68.0	28.6	40.9	23.3	48.8
	UGround	90.1	70.3	87.1	55.7	85.7	64.6	75.6
GPT-4o	SeeClick	81.0	59.8	69.6	33.6	43.9	26.2	52.3
	UGround	**93.4**	**76.9**	**92.8**	**67.9**	**88.7**	**68.9**	**81.4**

実験

オフラインエージェント評価

- ベンチマーク
  - ウェブ：Multimodal-Mind2Web
    - 100以上のウェブサイトにわたる1,013タスク
    - 評価指標: 正しい要素を選択する精度。操作の種類（クリック、入力など）の正しさは対象外
  - モバイル：AndroidControl
    - 833アプリにわたる15,000の一意なタスクを含む大規模データセット
    - 評価指標: 成功率(予測アクション、要素、引数がすべて正確な場合のみ成功)
  - デスクトップ：OmniACT
    - 38のデスクトップアプリケーションと27のウェブサイトにわたる9,802タスク
    - 評価指標: PyAutoGUIスクリプト（アクションシーケンス）の精度

実験

オフラインエージェント評価

- Multimodal-Mind2Webの結果
  - GPT-4によるChoiceやSoMといった既存のグラウンディング手法や、先行研究の資格グラウンディングモデルSeeClickを上回る
    - Choice: HTML要素の候補リストから該当の要素を選択する
    - SoM: 要素に割り当てられたマーク(ラベル)を選ぶ

Input	Planner	Grounding	Cross-Task	Cross-Website	Cross-Domain	Avg.
Image + Text	GPT-4	Choice	46.4	38.0	42.4	42.3
		SoM	29.6	20.1	27.0	25.6
Image (SeeAct-V)	GPT-4	SeeClick	29.7	28.5	30.7	29.6
		UGround	45.1	44.7	44.6	44.8
	GPT-4o	SeeClick	32.1	33.1	33.5	32.9
		UGround	**47.7**	**46.0**	**46.6**	**46.8**

実験

オフラインエージェント評価

- AndroidControlとOmniACTの結果
  - 全てのベンチマークでベースラインを上回る
  - 詳細は割愛

Table 5: Step accuracy on AndroidControl over 500 random actions from the test split. Baseline results are from Li et al. (2024b).

Input	Planner	Grounding	Step Accuracy	
			High	Low
Text	GPT-4	Choice	42.1	55.0
Image (SeeAct-V)	GPT-4	SeeClick	39.4	47.2
		UGround	46.2	58.0
	GPT-4o	SeeClick	41.8	52.8
		UGround	**48.4**	**62.4**

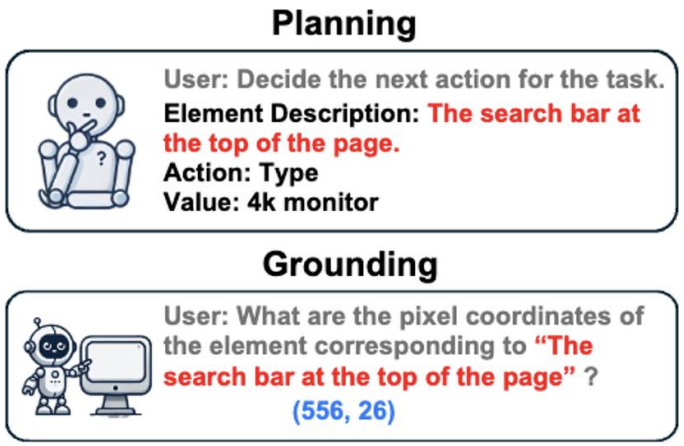
Table 6: Action scores (AS) on OmniACT. Baseline results are from Kapoor et al. (2024).

Inputs	Planner	Grounding	AS
Text	GPT-4	DetACT	11.6
Image + Text		DetACT	17.0
Image (SeeAct-V)	GPT-4	SeeClick	28.9
		UGround	31.1
	GPT-4o	SeeClick	29.6
		UGround	**32.8**

実験

エラー分析

- エラーの分類
  - 計画エラー
    - プランナーが、操作すべきUI要素について間違った説明を生成してしまうエラー
    - 例えば、「送信ボタン」をクリックすべきなのに、「キャンセルボタン」の説明を生成してしまう
  - グラウンディングエラー
    - プランナーは正しい要素の説明を生成したにもかかわらず、グラウンディングモデル間違った画面上の位置（座標）を予測してしまうエラー
- →マニュアルで分析



実験

エラー分析

- 分析結果
  - 失敗の大部分は「計画エラー」によるもの
  - 「グラウンディングエラー」も一定数見られた
    - 主な原因は、モバイルやデスクトップUIでよく使われる特異な意味を持つアイコン（例: 特定のアプリを表す独自デザインのアイコン）の理解が難しいこと

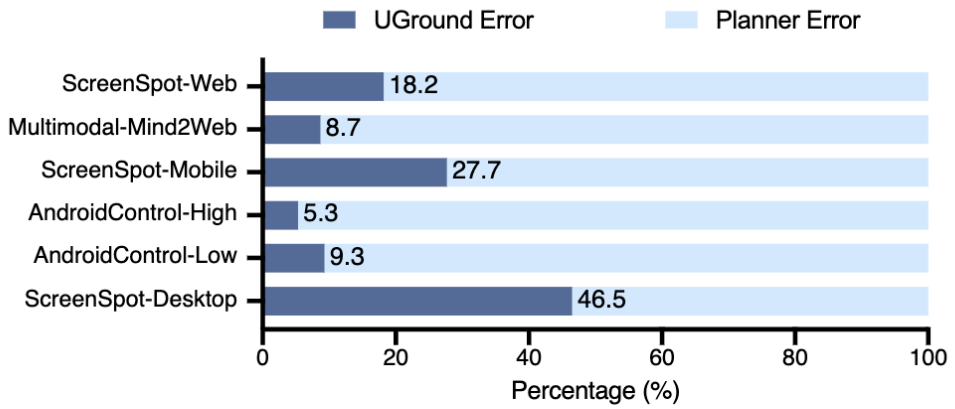


Figure 4: Error distribution from manual analysis.

実験

学習データ分析

- Web-Hybrid データセットのスケーリング分析
  - (結果) 訓練データ量が増えるにつれて、性能は一貫して向上したが、10万件を超えると、性能の伸びは鈍化する傾向が見られた

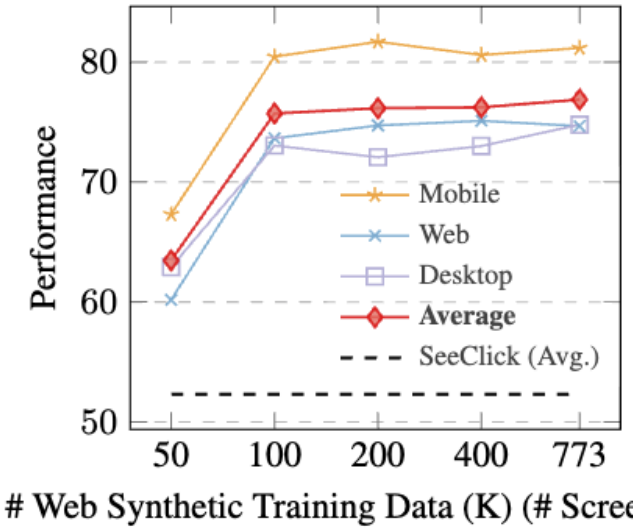


Figure 5: Scaling curve of UGround on ScreenSpot w.r.t. Web-Hybrid data size.

まとめ

手法

- 大規模なウェブベースの合成データを用いて開発された、普遍的なGUI視覚グラウンディングモデルUGround
- 視覚情報（スクリーンショット）のみを入力とし、ピクセルレベルでのGUI操作を可能にするSeeAct-Vフレームワーク

結果

- UGroundを組み込んだSeeAct-Vベースのエージェントは、オフライン（事前記録データ）およびオンライン（実環境）の両方の評価において、追加のテキスト情報に依存する既存のSoTAエージェントと比べて、同等かそれ以上の性能を達成

まとめ

限界点と今後の課題

- 訓練データの効率
  - ウェブページ間には類似した要素や繰り返しが多く存在するため、データのグルーピングや重複排除を工夫することで、訓練データの効率を改善する余地がある
- ロングテール要素への対応
  - 出現頻度の低い特殊な要素（ロングテール要素）への対応がまだ不十分。特にモバイルやデスクトップのUIには、特有の意味を持つアイコンが多く存在し、これら全てを訓練データで網羅するのは現実的ではないのでこの問題への対応は今後の課題。
- デスクトップUIデータの不足
  - 本研究では、デスクトップUIのデータは訓練に使用されておらず、デスクトップUIにおける性能の限界の一因となっている。今後のより包括的なデータセット開発が期待される。