# [C5-2] Compressed-a11y: 視覚的文脈の再構成と冗長性削減による GUI エージェント観測の効率化

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


- GUI エージェント向け観測表現 **Compressed-a11y** を提案
- GUI の視覚的レイアウト情報の保持と冗長性削減を両立
- 入力トークン数の削減とタスク成功率の向上を実証


## Background

### GUI エージェント

GUI 操作を行う AI エージェント (GUI エージェント) は,
マルチモーダル大規模言語モデル (MLLM) の発展に伴い,
高度なプランニングを要求されるタスクを遂行可能に

### オープンソースモデルを用いた GUI エージェント

クローズドモデルと比較し, データプライバシーの確保や
通信レイテンシの低減などメリットがある一方,
タスク指示と UI 要素の対応付け (**grounding**) が課題
→ **モデルが理解しやすい GUI 観測表現が必要**

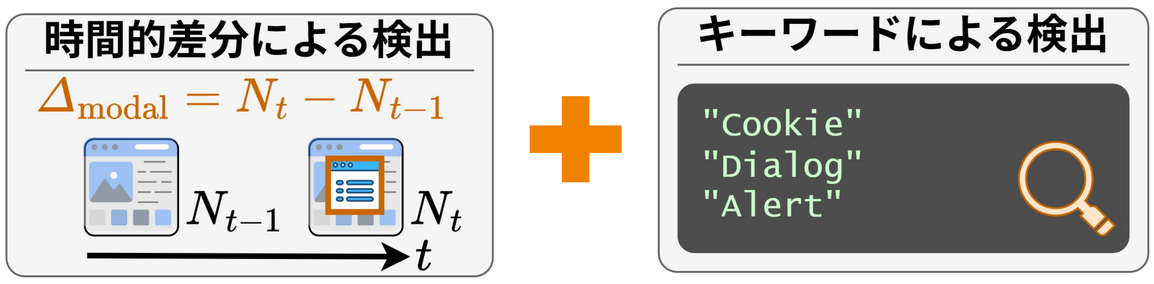
### テキストベースの GUI 観測表現

- **Accessibility (a11y) tree**：
  - 😊 UI要素の構造情報を階層構造で表現し, grounding に有効
  - 😭 冗長でモデルの入力コンテキスト長を圧迫
  - 😭 視覚的レイアウト情報 (UI 要素の重なり等) が欠落
- **Linearized a11y tree** [Xie+,NeurIPS'24]：
  - 😊 a11y tree の階層構造を一次元のテキスト列へ線形化
  - 😊 タグによるフィルタリングで冗長の削減
  - 😭 **依然として視覚的レイアウト情報の欠落と冗長性が残存**

→ **新たな観測表現 Compressed-a11y を提案**

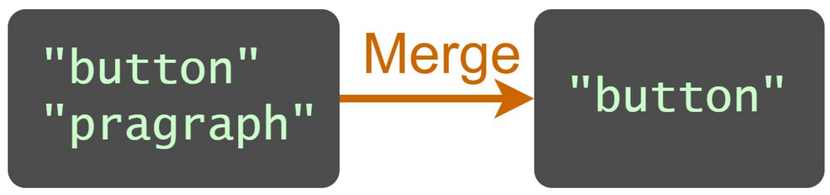
## Methods

### モーダル (前面に出現する UI 要素) の検出



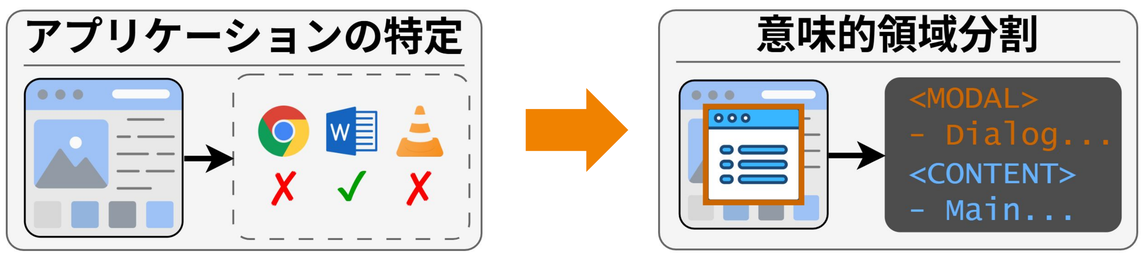

### 冗長性削減

- a11y tree 上でタグが異なる重複 UI 要素の統合




- タスク指示に基づく pragraph の動的圧縮

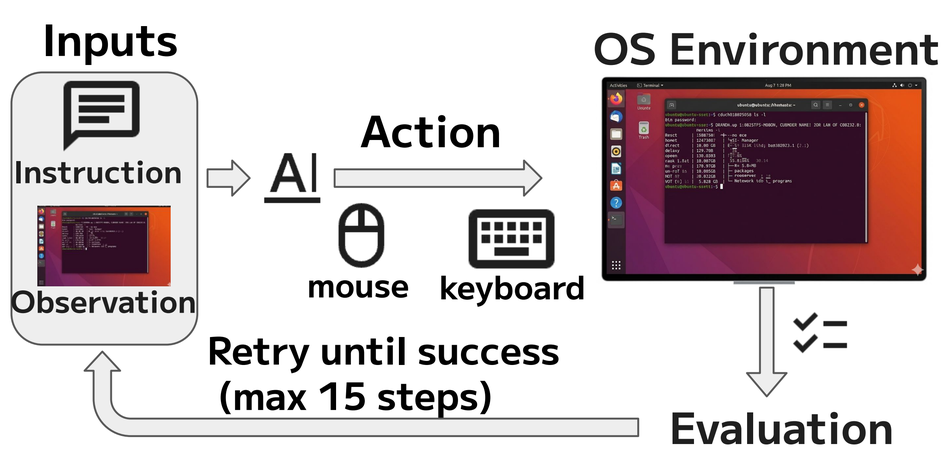

### 意味的領域分割




## Experiments & Results

### GUI エージェント評価ベンチマーク OSWorld [Xie+,NeurIPS'24] で評価

- 評価タスク：全 358 タスク (各タスク最大15ステップ)
  - Chrome: 44, Writer: 23, Calc: 46, Impress: 47, Thunderbird: 15, GIMP: 26, VLC: 17, VS Code: 23, OS: 24, Multi Apps: 93
  - 非決定性を考慮し, 各タスク2回試行, いずれか成功で成功判定
- 観測表現：Compressed-a11y (ours), Linearized a11y tree, Screenshot
- 使用モデル：Qwen3-VL-32B (temperature = 1.0, top_p = 0.9)




### 実験結果：1ステップごとのモデルへの平均入力トークン数

| Method | Chrome | GIMP | Calc | Impress | Writer | Thunderbird | VLC | VS Code | OS | Multi Apps | Overall |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Linearized a11y tree | 5538 | 3352 | 30181 | 5585 | 4110 | 3040 | 1609 | 7228 | 820 | 8607 | 8758 |
| **Compressed-a11y(ours)** | **1741** | **1110** | **3438** | **1859** | **2321** | **2045** | **791** | **2767** | **536** | **1709** | **1910** |

### 実験結果：アプリケーションドメイン別タスク成功率

| Method | Chrome | GIMP | Calc | Impress | Writer | Thunderbird | VLC | VS Code | OS | Multi Apps | Overall |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Screenshot | 0.045 | 0.115 | 0.000 | 0.021 | 0.043 | 0.000 | 0.118 | 0.043 | 0.208 | **0.108** | 0.070 |
| Linearized a11y tree | 0.182 | 0.192 | 0.000 | 0.149 | 0.087 | 0.267 | **0.294** | 0.304 | 0.333 | **0.108** | 0.156 |
| **Compressed-a11y(ours)** | **0.250** | **0.231** | **0.043** | **0.191** | **0.304** | **0.467** | **0.294** | **0.348** | **0.375** | **0.108** | **0.207** |

## Discussion & Conclusion

- Compressed-a11y は, モデルへの平均入力トークン数を削減し, ほとんどのアプリで最も高い成功率を達成
- トークン数の削減率はアプリ依存であり, 画面が複雑でトークン数が多いアプリほど高くなる傾向を確認
- 情報圧縮は成功率向上に寄与しており, GUI 操作に必要な本質的情報の抽出が grounding の向上に重要
- 複数アプリが同一画面に共存する Multi Apps 環境では改善は限定的であり, 単一アプリ前提の設計が要因

### Future Work

単一アプリ前提の設計から複数のアプリ混在環境への適応, モーダル検出精度の向上