UI-Redline-bench: 赤入れ指示による WebUI コード修正ベンチマーク

肥合 智史 [1]　藤井 諒 [1]　岸波 洋介 [1]　森下 睦 [1]
[1] フューチャー株式会社
team-nlp-research@future.co.jp

概要

本研究では，WebUI 開発における赤入れプロセスを模倣した，スクリーンショット上の視覚的指示に基づいてコードを修正するタスクを提案し，ベンチマークを構築する．既存研究が主に言語的指示を扱うのに対し，本研究では手書きやデジタルの描画による直感的な修正指示を直接入力として扱う点が特徴である．主要な VLM を用いた評価実験の結果，指示と Web コンテンツを適切に区別する能力の重要性や，矢印による参照等，離れた位置の情報を統合的に理解する能力に課題があることを確認した．

1　はじめに

近年，Vision-Language Model (VLM) に代表されるマルチモーダルモデルの発展により，画像とテキストを統合的に処理することが可能となっている [1, 2, 3]．VLM の応用範囲は，画像のキャプション生成や視覚質問応答だけでなく，表やチャートの解析，GUI 画面の操作支援といった幅広いタスクへと拡大している．特に，Web サイトやモバイルアプリのユーザインタフェース (UI) は，ボタンやフォーム，テキストなどを要素とする構造化された視覚情報として VLM の有力な応用対象となっている．先行研究では，ウェブサイト上でのタスク実行ベンチマーク [4] や，フロントエンド UI コード編集ベンチマーク [5, 6] が提案され，WebUI に対する言語的指示による操作・編集能力が評価されている．

一方で，実務における WebUI 開発の現場では，仕様や修正内容が必ずしも言語的指示だけで伝達されるわけではない．例えば，「右上のボタンを少し左へ」等の微細な調整は言語化が難しく，認識の齟齬が生じやすい．Newman ら [7] は，デザインの検討過程において，Web サイトのデザイナーがプリントアウトされた図面に手書きのスケッチや注釈を加えて同僚へフィードバックする事例を報告している．そのため，手書きの意図を解釈し，デジタルな実装へ反映させるプロセスを自動化できれば，迅速な試行錯誤が可能になり，開発効率の向上が期待できる．このような背景から，視覚的指示を直接的にコード修正へと繋げる能力は，VLM の極めて有用な応用シナリオの一つになると考えられる．

図 1　タスク概要．VLM は修正指示が描き込まれた画像と対応する HTML/CSS コードを入力として，修正後コードを出力する．

そこで本研究では，WebUI のスクリーンショット上に描かれた手書き修正指示に基づくコード編集を行うタスクを提案し，その能力を評価するベンチマーク **UI-Redline-bench** を構築する (図 1)．本稿の貢献として，手書き修正指示に基づく Web UI コード編集タスクの提案，および実務のデザインレビューを模したデータセットを含むベンチマークの構築が挙げられる [1)]．さらに，既存の VLM を用いた評価を実施し，手書き指示の解釈や UI 要素の特定における課題を明らかにした．

1) 構築したデータセットは https://huggingface.co/datasets/future-architect/UI-Redline-bench に公開した．

2 関連研究

WebUI の開発プロセスを支援するためのベンチマークが近年提案されている．WebUIBench [5] は，Web フロントエンド開発におけるコード生成能力を評価する包括的なベンチマークである．ここでは，自然言語による機能要件に基づいてフロントエンドのコードを生成・修正するタスクが設定されており，VLM のコード実装能力に焦点を当てている．また，DesignBench [6] は，マルチモーダルモデルのデザイン能力を多角的に評価する枠組みである．色選定，レイアウト生成，タイポグラフィなどの観点から評価を行うが，その指示入力は主にテキストプロンプトやバウンディングボックスなどの構造化されたデータを用いている．これらの研究は，言語的指示やメタデータに基づく生成・編集を主眼としている．一方で，スクリーンショット上に直接描かれた「手書きの修正指示」という，より直感的かつ構造化されていない視覚情報を直接の入力としてコード編集を行う能力については評価されていない．本研究は，実務を踏まえた視覚的な修正指示フローをタスク化する点でこれらと異なる．

3 視覚的指示追従タスクの定義

本研究では，WebUI 開発における「赤入れ」修正プロセスを模倣した，視覚的指示に基づくコード編集タスクを定義する．

レンダリング可能なフロントエンドコードの空間を $\mathcal{C}$，任意の RGB 画像からなる空間を $\mathcal{I}$ と定義する．また，ソースコードを Web サイトの画面イメージに変換するレンダリング処理を，写像 $\mathcal{R} : \mathcal{C} \to \mathcal{I}$ とする．本ベンチマークの構築では，修正前の Web サイトのソースコード $C_{src} \in \mathcal{C}$ をレンダリングして得られた画像 $\mathcal{R}(C_{src})$ に対して，人間が視覚的な修正指示 A_{vis} を追記する．なお，A_{vis} は，手書きの線，囲み，矢印，テキスト等からなり，画像編集ソフトウェアを用いて元の画像に直接描画される．ここで，画像に対する可能なアノテーションの集合を $\mathcal{A}$ とすると，この書き込み操作は $\mathrm{Overlay} : \mathcal{I} \times \mathcal{A} \to \mathcal{I}$ と定式化される．モデルは，C_{src} および，そのレンダリング結果に視覚的修正指示を上書きした画像 $\mathrm{Overlay}(\mathcal{R}(C_{src}), A_{vis})$ を入力として，指示の意図が反映されたソースコード C_{tgt} を生成する．

評価においては，モデルによって出力されたコード C_{tgt} をレンダリングした結果 $\mathcal{R}(C_{tgt})$ が，視覚的指示 A_{vis} で意図された修正内容 (例：要素の移動，色の変更) を正しく反映しているかを判定する．本タスクの難しさは，言語のみの指示とは異なり，「このボタンを右に動かす」「ここの色をこれにする」といった指示対象や変更内容が，画像上の空間的な位置関係や描画内容に依存して定義される点にある．

4 データセット構築

構築手順は，ベース Web サイト生成，修正案の生成，視覚的指示アノテーションの 3 段階からなる．各段階の詳細を以下に述べる．

4.1 LLM によるベース Web サイトの生成

まず，修正前のベース Web サイト C_{src} を作成した．Web サイトの多様性を確保するため，Web サイト 3 種類 (ポートフォリオ，オンラインストア，ニュースサイト)，使用する CSS フレームワーク 3 種類 (フレームワークを使用しない Vanilla, Bootstrap, Tailwind CSS) を組み合わせた計 9 種類の構成を用意した．各構成について，GPT-5 を用いて HTML/CSS コードを生成した．画像アセットは，上記のコード生成時にはプレースホルダを使用した．あてはまる画像の説明もコード生成と共に出力しておき，それを基に GPT-5 で適切な画像を生成し，配置した．

4.2 LLM による修正案の作成

次に，画像への描画を行う前段階として，各ベース Web サイトに対して，テキスト記述による修正案 A_{txt} と，それを反映した修正後 Web サイト C_{ref} を作成した．修正案の策定にあたっては，UICrit [8] で定義されているデザイン批評の観点を参考に，レイアウト，配色，可読性，ボタンのユーザビリティ，視認性・学習可能性の 5 カテゴリを採用した．

GPT-5 により，各カテゴリについての修正案を生成した．この修正案はテキストのみによって記述されており，視覚的な情報は含まれていない．生成にあたっては，ページ全体の大幅な改変ではなく，「特定のボタンの色を変更する」「見出しのフォントサイズを大きくする」といった，特定のコンポーネントに対する視覚的な変更を伴う修正指示を生成するようプロンプトを記述した．

4.3 人手による視覚的指示の作成

最後に，生成された修正案 A_{txt} および修正後 Web サイト C_{ref} の品質を人間が検証し，視覚的指示 A_{vis}

表 1 指示のスタイルごとの評価結果.

モデル	All	Hand-written	Digital
GPT-5	84.7	82.6	86.9
Claude Sonnet 4	36.7	30.3	43.1
Gemini 2.5 Pro	76.9	76.3	77.4
Qwen3-VL-32B	62.1	65.7	58.6

を作成した.

まず，前節の修正案 A_{txt} が修正後 Web サイト C_{ref} に正しく反映されているかを著者 4 名で検証した．次に，未反映や大幅なレイアウト崩れが無いことが確認されたペア (C_{src}, C_{ref}) に対してのみ，修正前画像 $\mathcal{R}(C_{src})$ 上に修正箇所や内容を示す視覚的指示 A_{vis} を描画した．なお，指示のスタイルによる影響を分析するため，同一の修正内容に対して 2 種類の指示画像を作成した．一つは，フリーハンドで囲みや矢印，指示テキストを書き込んだ**手書き指示 (Hand-written)** であり，もう一つは，それを図形ツールやテキストボックスを用いて忠実に再現した**デジタル指示 (Digital)** である.

以上の手続きで，著者 4 名が，9 種類のベース Web サイトに対し生成された，5 カテゴリ分の異なる修正案 A_{txt} に対してアノテーションを行った．最終的に，構築したデータセットは合計で 350 インスタンス (175 修正事例 × 2 スタイル) から構成される [2)].

5 実験

5.1 実験設定

構築したデータセットを用いて，主要な VLM である GPT-5, Claude Sonnet 4，Gemini 2.5 Pro [9], Qwen3-VL-32B-Thinking [10] の 4 モデルを評価した．データセットに含まれる全 350 インスタンスに対し，各モデルで推論を行い，合計 1,400 件 (350×4) の出力コードを得た.

評価においては，修正後画像 $\mathcal{R}(C_{tgt})$ と指示画像 Overlay$(\mathcal{R}(C_{src}), A_{vis})$ を比較し，指示が概ね正しく反映された**成功**，反映が不完全または副作用が生じた**部分成功**，全く反映されないかコード破損により描画できない**失敗**の 3 段階で評価した．生成された 1,400 件の結果すべてを人手で評価することはコストが大きいため，LLM を用いた自動評価を採用した．自動評価の妥当性検証は付録 A に記載する.

2) 検証で棄却されたケースは 5 事例であった.

表 2 描き込み要素ごとのスコア平均.

描き込み要素	有り	無し
矢印	60.0	69.5
位置を示す囲い	70.8	53.9
追加 UI の概形	57.0	67.6

表 3 矢印と概形についてのスコア平均詳細.

	概形有り	概形無し
矢印有り	48.9	66.1
矢印無し	76.6	68.5

5.2 定量評価

定量評価値として，成功を 100，部分成功を 50，失敗を 0 としてスコア化し，その平均値を採用した.

表 1 の All にモデルごとのデータ全体におけるスコア平均を示した．GPT-5 が最も高く，次いで Gemini 2.5 Pro が高スコアであった．一方，Claude Sonnet 4 は全体的にスコアが低迷した．Claude では，長辺が規定の閾値を超える画像はアスペクト比を維持したままリサイズされる [3)]．本実験で用いた修正前 Web サイト画像は縦長の形状をしており，過度に縮小され，手書きの修正指示などの微細な情報が失われたことが，著しい性能低下を招いたと考えられる.

また，指示のスタイルで比較すると，GPT-5, Claude Sonnet 4，Gemini 2.5 Pro では，Digital の方が Hand-written よりも高いスコアを示した．これは，手書き文字やフリーハンドによる図形の認識精度が影響しているためだと考えられる．対照的に，Qwen3-VL-32B においては手書き指示の方が高スコアであった．誤答事例において画像内にアノテーションされた視覚的指示を読み取る実験を行ったところ，デジタル指示の場合，モデルが指示文だけでなく周囲の無関係なコンテンツまで過剰に読み取る傾向が確認された．Waseda ら [11] は，文脈に応じて画像内の必要なテキストのみを利用する「選択的テキスト理解」の重要性を指摘しており，本タスクにおいても，指示と Web ページのコンテンツを適切に分離・取捨選択する能力が性能を左右することを示唆している.

ベース Web サイト種類や CSS フレームワーク，修正カテゴリのスコアへの影響は付録 B に記載する.

3) https://platform.claude.com/docs/en/build-with-claude/vision

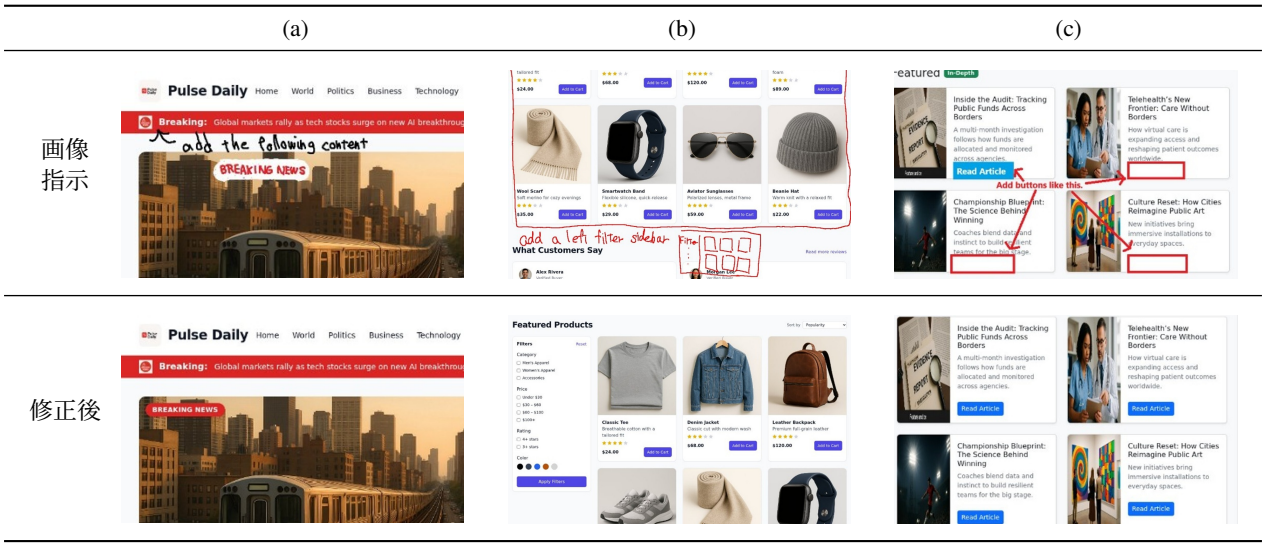

図 2　分析事例．矢印と追加 UI の概形が描かれた指示に対する失敗事例 (a) と，修正位置を文字で記入 (b) もしくは囲み線で示した (c) 指示に対する成功事例．

5.3　事例分析

失敗事例として，図 2 (a) のように修正位置から矢印を引いて離れた位置に概形を描写する事例が多く観察された．この要因に関して，視覚的指示における描き込み要素から検証した．描き込み要素として，矢印，位置を示す囲い，追加 UI の概形の三要素それぞれの有無ごとに全モデルのスコア平均を集計した．その結果，表 2 に示すように，矢印もしくは追加 UI の概形が描き込まれた場合，スコア平均が低くなった．そこで，さらに矢印と概形描写の共起に関して分けてスコアを集計し，表 3 に示した．図 2 (a) のように矢印と概形が同時に描かれた場合のスコアが著しく低くなった．このような例に関しては，矢印が示す修正位置と，実装すべき概形の紐づけに失敗していると考えられ，離れた場所にある視覚情報を統合することが VLM にとって困難であることを示唆している．一方で，矢印無しで概形のみが描かれた場合は高スコアとなった．これは，図 1 や図 2 (b) のように修正位置に隣接して概形を直接描写したり，位置を文字で示したものであり，指示と対象の空間的距離が近いことが精度向上に寄与していると考えられる．

また，表 2 に示す通り，位置を示す囲いがある場合，高スコアとなった．これは，図 2 (c) のような例で，囲み線による明示的な領域指定が，空間的に離れた情報の結びつきを強化したと考えられる．

以上より，VLM は矢印によって示される修正位置と実装すべき概形を，一連の意味的な塊として正しく認識・統合できていないことが示唆される．修正位置に上書きもしくは隣接した位置に描写することで精度は改善するものの，スペースが限られる実際の UI 画面では描き込みが困難な場合もある．また，そのような指示をユーザに強いることは手書きによる直感的指示の利点を損なう結果となる．したがって，矢印を用いた離れた位置への参照や概形描写を含む，より直感的かつ抽象的な視覚的指示の意図を汲み取る能力の獲得が，VLM の今後の実用化に向けた重要な課題であるといえる．

6　おわりに

本研究では，WebUI 開発における赤入れプロセスを模倣した，視覚的指示に基づくコード修正タスクを提案し，その能力を評価するベンチマーク **UI-Redline-bench** を構築した．実験の結果，指示文と Web サイトのコンテンツを適切に区別・分離する能力の重要性や，指示と対象の空間的距離がモデルの解釈を阻害する要因となることが明らかになった．今後の展望としては，直感的かつ抽象的な指示への対応を実現するため，VLM の空間的な推論能力の強化が不可欠である．具体的には，矢印の指し示す先や，描き込みの意味的な一塊の認識といった，マルチモーダルな照応解析技術の検討が挙げられる．また，データセットに関して，現時点ではアノテータ 4 名による 350 事例の小規模なものであるため，デザイナーを含む多様なアノテータの採用やサンプル数の増加によるデータセットの拡張を行っていく．

謝辞

本研究成果の一部は，データ活用社会創成プラットフォーム mdx を利用して得られたものです.

参考文献

[1] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 48, No. 8, pp. 5625–5644, 2024.

[2] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In **Proceedings of 2021 IEEE Winter Conference on Applications of Computer Vision (WACV)**, pp. 2199–2208, 2021.

[3] Zeyuan Huang, et al. SketchGPT: A sketch-based multimodal interface for application-agnostic llm interaction. In **Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST)**, pp. 1–18, 2025.

[4] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. In **Advances in Neural Information Processing Systems (NeurIPS)**, pp. 28091–28114, 2023.

[5] Zhiyu Lin, Zhengda Zhou, Zhiyuan Zhao, Tianrui Wan, Yilun Ma, Junyu Gao, and Xuelong Li. WebUIBench: A comprehensive benchmark for evaluating multimodal large language models in WebUI-to-code. In **Findings of the Association for Computational Linguistics (ACL)**, 2025.

[6] Jingyu Xiao, Ming Wang, Man Ho Lam, Yuxuan Wan, Junliang Liu, Yintong Huo, and Michael R. Lyu. Designbench: A comprehensive benchmark for mllm-based front-end code generation. In **arXiv:2506.06251**, 2025.

[7] Mark W. Newman and James A. Landay. Sitemaps, storyboards, and specifications: a sketch of web site design practice. In **Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS)**, pp. 263–274, 2000.

[8] Peitong Duan, Chin-Yi Cheng, Gang Li, Bjoern Hartmann, and Yang Li. UICrit: Enhancing automated design evaluation with a UI critique dataset. In **Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST)**, pp. 1–17, 2024.

[9] Google Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. In **arXiv:2507.06261**, 2025.

[10] Qwen Team. Qwen3 technical report. In **arXiv:2505.09388**, 2025.

[11] Futa Waseda, Shojiro Yamabe, Daiki Shiono, Kento Sasaki, and Tsubasa Takahashi. Read or ignore? a unified benchmark for typographic-attack robustness and text recognition in vision-language models. In **arXiv:2512.11899**, 2025.

[12] J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. **Biometrics**, Vol. 33, No. 1, pp. 159–174, 1977.

A　評価手法の検証

LLM による自動評価の信頼性を担保するため，本評価に先立ち，以下の手順で人手評価との相関を検証した．評価モデルには，GPT-5 を利用した (以下，GPT-5-Score)．全 1,400 件の推論結果から検証用データとして 128 件をランダムに抽出した．この 128 件を，データセット構築に参加した 4 名のアノテータが評価した．各アノテータは自身が作成に関与していない事例を評価対象とし，各事例につき 3 名の評価者が割り当てられるようにした．検証の結果，3 名のアノテータ間の一致率は Fleiss' kappa 係数で 0.853 となり，ほぼ完全な一致 [12] を示した．また，各事例について多数決により正解ラベルを決定し，これと GPT-5-Score との一致度を測定したところ，Cohen's kappa 係数で 0.639 となった．これはかなりの一致 [12] に相当し，本タスクの評価において GPT-5 が人間の代用として一定の信頼性を持つことを示唆している．以上の結果に基づき，残りのデータを含む全件の評価に GPT-5-Score を適用した．

B　結果の詳細

表 4 に，ベース Web サイトの種類ごとのスコアを示す．Portfolio が最も高いスコアとなり，News と Store が続いた．Store や News は，同一構造のリストコンポーネントが大半を占めるため，共通クラスへの変更による意図しない波及が生じやすい．特に News は，図 3 のように，サイドバーがある非対称なグリッドレイアウトや深いネスト構造といった複雑な構造も相まって対象の特定が困難であったと考えられる．対して Portfolio は，機能ごとに独立した構成のため，修正対象の特定が容易であったことが高スコアに繋がったと考えられる．

表 5 の CSS フレームワーク別の結果では，Vanilla と比較して Bootstrap や Tailwind のスコアが低い傾向が見られた．各フレームワークを利用したコードの修正には，フレームワーク特有のクラス名の意味を正確に把握する必要があり，これがタスクの難易度を高めたためだと考えられる．また，フレームワーク間では Tailwind が Bootstrap を上回る傾向にあった．特に配色カテゴリの修正事例においてその差が顕著であった．具体的には，Tailwind では，クラス名を HTML に追加して色を指定する HTML の変更が支配的であったのに対し，Bootstrap では，CSS ファイル内でセレクタを記述してスタイルを上書きする CSS の記述量が支配的であった．したがって，Bootstrap におけるセレクタ記述の生成難易度の高さが，スコア差の主因と考えられる．

表 6 に，各モデルの修正カテゴリごとの結果を示す．すべてのモデルにおいて「レイアウト (L)」のスコアが他のカテゴリに比べて著しく低い傾向であった．「可読性 (R)」や「配色 (C)」といったタスクは，特定のタグ内のテキストや CSS プロパティを局所的に変更することで達成可能である．対してレイアウトの変更は，DOM 構造の並び替えや，親要素へのスタイル適用など，大域的な構造理解と依存関係の考慮が必要となるため，より困難な課題であると考えられる．

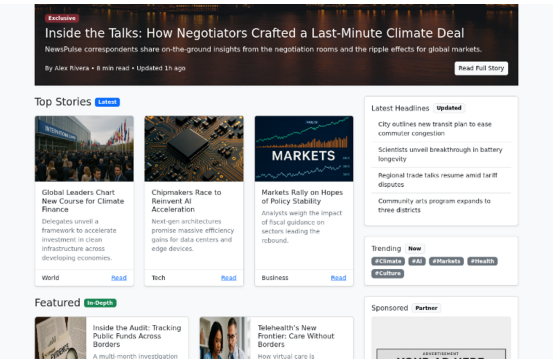

図 3　Bootstrap を利用した修正前 news サイト．

表 4　ベース Web サイト種類ごとの評価結果．

モデル	All	Portfolio	Store	News
GPT-5	84.7	86.9	86.0	81.1
Claude Sonnet 4	36.7	39.8	39.4	30.7
Gemini 2.5 Pro	76.9	80.9	75.8	73.7
Qwen3-VL-32B	62.1	66.9	64.0	55.3

表 5　CSS フレームワークごとの評価結果．

モデル	Vanilla	Bootstrap	Tailwind
GPT-5	89.6	77.6	86.6
Claude Sonnet 4	45.0	28.1	36.6
Gemini 2.5 Pro	79.6	73.2	77.6
Qwen3-VL-32B	65.0	57.9	63.4

表 6　モデル・カテゴリごとの評価結果 (L: レイアウト, C: 配色, R: 可読性, B: ボタンのユーザビリティ, V: 視認性・学習可能性)．

モデル	L	C	R	B	V
GPT-5	73.4	86.4	91.7	84.0	86.8
Claude Sonnet 4	31.2	29.3	43.8	37.5	41.0
Gemini 2.5 Pro	71.9	77.9	86.1	72.9	75.0
Qwen3-VL-32B	45.3	65.0	81.2	60.4	56.9