A Modular Web UI Grounding System for Automated Social Media Post Scheduling

Hui Gi Wai
23444584

Abstract

End-to-end Vision-Language-Action (VLA) models achieve strong GUI grounding but incur high compute costs and expose opaque failure modes. We present a modular three-stage pipeline – YOLOv8 proposal generator, CLIP ViT-L/14 retriever, Qwen2.5-VL-3B LoRA reranker – that enables stage-wise bottleneck analysis. Trained on ShowUI-Web, the detector reaches recall@200 of 0.801 and fine-tuning lifts retriever hit@20 from 0.214 to 0.554 (+157%). The grounder is paired with a Qwen3-8B LoRA post generator and a Playwright executor to automate tech-news scheduling on Threads. For out-of-domain adaptation, continuing LoRA fine-tuning on a 1,684-sample Viraly trace lifts hit@1 from 14.8% to 60.7% (+45.9 pp), recovering clock-dial interactions that enable end-to-end scheduling.

1. Introduction

Vision-Language-Action (VLA) models such as SeeClick [1] and ShowUI [2] ground natural-language instructions directly onto GUI screenshots, but their monolithic design hides *where* errors originate and is expensive at inference time. We decompose grounding into three interpretable stages – proposal, retrieval, rerank – each trainable and measurable in isolation, and wrap the grounder with an LLM content generator and a browser executor to form a practically deployable tech-news post scheduler.

The system has three AI sub-systems: **(A)** a Qwen3-8B LoRA fine-tuned for Threads-style post generation; **(B)** a three-stage modular grounder trained on ShowUI-Web [2]; **(C)** a site-specific LoRA adaptation of the Stage 3 reranker to Viraly. Figure 1 summarises the integrated pipeline.

2. Task A: Threads-Style Post Generation

Dataset. Public posts from the Threads account `theartificialintelligence` were scraped and paired with the underlying tech-news article they reference. The released dataset[1] contains 1,016 (instruction, post) pairs (896 train / 120 val). Articles average 457 characters; posts median 122, max 481 (all <500, the Threads limit).

	Task A (Qwen3-8B)	Stage 3 (Qwen2.5-VL-3B)	Task C (cont. Stage 3)
Quantisation	4-bit NF4	—	—
LoRA rank r	8	8	8
α	16	16	16
Dropout	0.05	0.05	0.10
LR	1×10^{-4}	1×10^{-4}	5×10^{-5}
Eff. batch	16	8	8
Epochs	2	3	6
Max seq. len	1024	—	—

Table 1. LoRA hyperparameters across all fine-tuning stages.

Model and Training. We LoRA fine-tune `Qwen/Qwen3-8B` under 4-bit NF4 QLoRA; hyperparameters are listed in Tab. 1 (Task A column). Train loss falls from 1.78 to 1.68; validation loss converges smoothly.

Evaluation. On 50 greedy-decoded held-out samples the adapter achieves 100% JSON validity and 98% compliance with the 500-character Threads limit. The base Qwen3-8B emits verbose chain-of-thought before loosely structured text; the LoRA adapter emits clean JSON directly, eliminating all formatting failures.

3. Task B: Modular Web UI Grounding

3.1. Dataset: ShowUI-Web

We adopt the original ShowUI-Web dataset [2][2] (21,988 grounding pairs, 17,022 screenshots, 22 websites, 8 UI-element classes) and release a processed version with

[1] E. Hui, *Threads English Tech-News SFT Dataset*, 2026. https://huggingface.co/datasets/e1879/threads-english-tech-news-sft

[2] https://huggingface.co/datasets/showlab/ShowUI-web

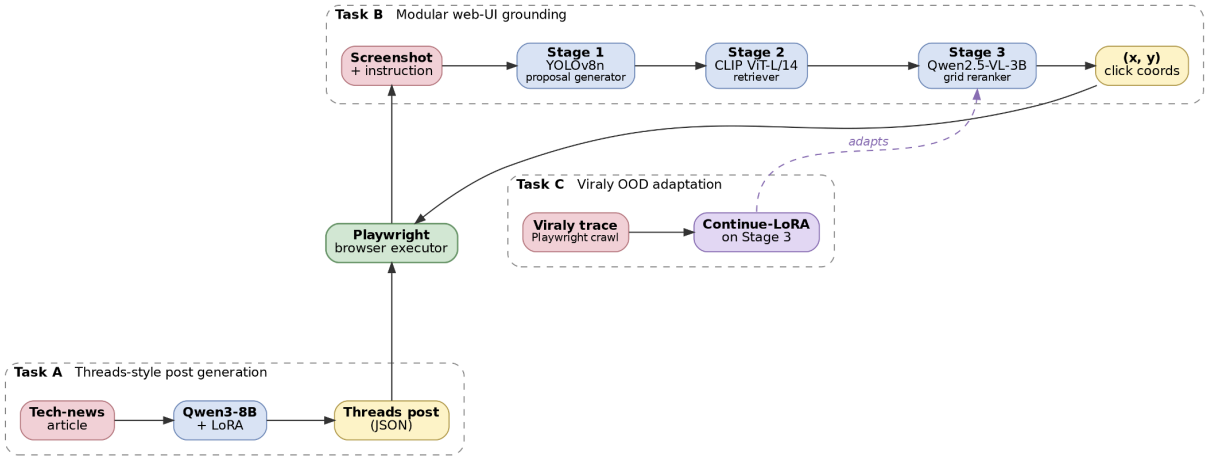


Figure 1. Task A fine-tunes Qwen3-8B with QLoRA on Threads tech-news pairs, emitting a JSON-valid post from a news article. Task B grounds UI elements with a three-stage modular pipeline trained on ShowUI-Web: a YOLOv8n class-agnostic proposal generator, a CLIP ViT-L/14 retriever, and a Qwen2.5-VL-3B LoRA grid reranker that predicts the target tile. Task C continues LoRA fine-tuning of Stage 3 on a Viraly trace to adapt the reranker to site-specific clock-dial interactions. A Playwright browser executor closes the loop: it renders screenshots for Task B, receives the predicted click coordinates, and types the Task A post body to complete a 10-step scheduling workflow.

scenario-wise splits and detector proposals[3]. We resplit by *scenario* (website) to measure domain generalisation rather than image memorisation: train (17,022 pairs / 17 sites), val (1,687 / {apple.com, hotels.com}), test (3,279 / {bbc.com, espn.com, steampowered.com}). Bounding boxes occupy a mean of 0.86% of the canvas – small, dense targets dominate.

3.2. Stage 1 – YOLOv8 Proposal Generator

Design. YOLOv8n is retrained as a class-agnostic UI element detector. The objective is *high recall@K*, not precision: the ranker downstream chooses among proposals, so missing the true box is fatal but extra boxes are cheap.

Training. Table 2 lists the training and inference parameters.

Inference sweep. We sweep confidence threshold, NMS-IoU, max detections, and image size (Tab. 2). Table 3 shows the resulting recall@K curve on val. We adopt K=200 (recall@200 = **0.801**), where recall saturates; K=500 only gains 1.5 pp at additional ranker cost.

3.3. Stage 2 – CLIP ViT-L/14 Retriever

Design. Each Stage 1 proposal is expanded with contextual padding and cropped; a dual-encoder ranks the K crops against the instruction. Only the vision tower is fine-tuned (text tower frozen) to preserve instruction semantics while adapting to UI pixels.

[3]E. Hui, *ShowUI-Web Processed*, 2026. https://huggingface.co/datasets/e1879/showui-web-processed

Training		Inference	
Base weights	YOLOv8n	Conf. thresh	5×10^{-5}
Epochs	30	NMS-IoU	0.6
Optimiser	SGD	Max detections	1,200
LR schedule	Cosine	Image size	1,280
Warm-up	10%		
Image size	960		
Batch size	16		

Table 2. YOLOv8 proposal generator configuration.

K	50	100	**200**	500
recall@K	0.654	0.766	**0.801**	0.816

Table 3. YOLOv8 proposal recall@K on val (IoU$\geq$0.5). K=200 adopted.

Training. Table 4 lists the retriever training parameters. Two padding regimes were compared: R1 (pad 0.08, 384 px) vs. R2 (pad 0.12, 448 px); **R1 epoch 2** is selected.

Evaluation. Table 5 shows fine-tuning more than doubles hit@1 and nearly triples hit@20 over zero-shot CLIP, confirming that UI-element semantics require adaptation despite CLIP's web-scale pretraining. The fine-tuned hit@20 of 0.554 recovers 69% of the detector oracle ceiling (0.801), meaning the retriever successfully surfaces the gold element into the top-20 for two-thirds of all queries; the remaining misses are mainly tiny or visually ambiguous controls (e.g. icon-only buttons, toggle switches) whose crop appearance

Parameter	Value
Fine-tuned tower	Vision only (text frozen)
Negatives per positive	63 (mixed pool)
Learning rate	5×10^{-6}
Weight decay	0.01
Epochs	3
Selected checkpoint	R1 epoch 2

Table 4. CLIP ViT-L/14 retriever training configuration.

Model	hit@1	hit@5	hit@10	hit@20
Zero-shot CLIP	0.065	0.148	0.194	0.214
Ours (R1 ep. 2)	**0.176**	**0.324**	**0.443**	**0.554**

Table 5. CLIP ViT-L/14 retriever on val (K=200, oracle ceiling 0.801).

Exact-match	hit@1	hit@5	hit@10	MRR@20	Oracle
5.25%	0.176	0.324	0.443	0.224	0.801

Table 6. Full three-stage pipeline on ShowUI-Web val. Exact-match is the reranker alone (ep. 3); hit@K and MRR are end-to-end; Oracle is proposal recall@200.

overlaps with many competing candidates.

3.4. Stage 3 – Qwen2.5-VL LoRA Grid Reranker

Design. The top-M=20 CLIP crops are composed into a 5×4 numbered grid (224 px tiles, 8 px padding). Qwen2.5-VL-3B-Instruct is prompted to output the integer index of the target tile, reducing a vision regression task to structured text generation.

Training. LoRA hyperparameters follow the Stage 3 column of Tab. 1. Best-checkpoint selection uses validation exact-match. Training data are rebuilt from the refreshed Stage 2 retriever: 15,461 / 934 train/val rows kept after filtering cases where the gold box has no retrieved neighbour.

Evaluation. Table 6 reports the full pipeline results. Exact-match on val rises from 3.5% (ep. 1) to **5.25%** (ep. 3). End-to-end hit@1 is 0.176 with MRR@20 of 0.224 – *Stage 3 is the current bottleneck*, consistent with the median gold rank of 5 entering the reranker: a larger base model, chain-of-thought prompting, or iterative cropping are natural next steps. The low absolute accuracy despite the gold element being present in the grid (median rank 5 of 20) suggests that the 3B-parameter VL model struggles to resolve fine-grained spatial distinctions among visually similar crops at 224×224 tile resolution; scaling to a 7B backbone or providing the full screenshot alongside the grid may yield substantial gains.

4. Task C: Viraly OOD Adaptation

Dataset. A Playwright crawler records the ten-step Viraly scheduling workflow – type post, open date picker, open time picker, click hour, click minute, click AM/PM, confirm, schedule – producing 330 screenshots with 999 DOM-level annotations over 10 action types. Running the Task B detector+retriever on these screenshots emits 1,684 reranker training rows (val: 61).

Adaptation. We continue-LoRA fine-tune the Stage 3 checkpoint using the Task C column of Tab. 1; key changes from Stage 3 are a halved learning rate, doubled dropout, and 6 epochs.

Action taxonomy. The ten-step Viraly scheduling workflow contains three functional groups of click targets:

1. *General web UI* – Create Post button, Scheduled-For timestamp, calendar date cell, time-display opener, OK confirmation (5 actions). These resemble standard web elements present in ShowUI-Web.
2. *Clock-dial* – selecting hour and minute on a Material-UI analog TimePicker (2 actions). The radial layout, overlapping tick marks, and rotating selection arm have no analogue in the source domain.
3. *Site-specific controls* – AM/PM toggle and Schedule Post button (2 actions). Their custom styling departs from ShowUI-Web conventions.

A tenth action, typing the post body, is handled by the keyboard executor and does not involve the reranker.

Evaluation. Zero-shot transfer of the ShowUI-trained reranker to Viraly achieves only 14.8% hit@1 – the analog clock dial is essentially unseen in the source domain. Site-specific LoRA lifts hit@1 to **60.7%** (+45.9 pp). Table 7 breaks the gain down by action type.

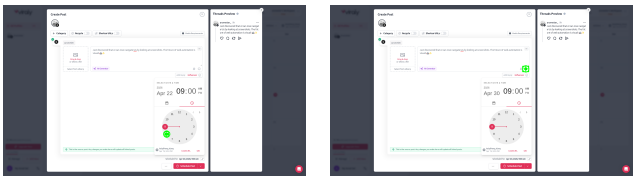
The reranker val set (N=61) consists entirely of clock-dial and schedule-button instances (clock-hour: 20, clock-minute: 39, Schedule Post: 2). The five general-UI actions are absent because the ShowUI-Web-trained retriever rarely ranks Viraly's custom UI components into its top-20 candidate set, preventing formation of valid reranker inputs for those actions. Their correctness is instead validated qualitatively through the end-to-end Playwright demo (Fig. 2).

Clock-minute shows the largest absolute gain (+53.8 pp to 74.4%), likely because the minute dial's 12 evenly spaced tick marks at the circle's perimeter produce visually distinctive crops; by contrast, the hour dial's 12 labels cluster in a smaller radius, creating more inter-candidate ambiguity and capping adapted accuracy at 35.0%. The Schedule-Post button jumps from 0% to 50%, indicating that even simple buttons can fail zero-shot when their visual style departs from the ShowUI-Web distribution.

Catastrophic forgetting. An initial LoRA run trained only on clock-dial actions exhibited catastrophic forgetting: clock accuracy improved but the model lost the ability to ground standard buttons and text links that the ShowUI-

Action	N	Zero-shot	Adapted	Δ
Clock hour	20	5.0%	35.0%	+30.0
Clock minute	39	20.5%	74.4%	+53.8
Schedule Post	2	0.0%	50.0%	+50.0
Overall	**61**	**14.8%**	**60.7%**	**+45.9**

Table 7. Viraly reranker hit@1 before vs. after site-specific LoRA. General-UI actions are absent from the val set (see text) and are validated only in the end-to-end demo.



(a) Adapted LoRA (b) Baseline (zero-shot)

Figure 2. Clock-hour click comparison. The instruction is to click hour "7" on the analog dial. (a) The adapted LoRA places its click (green circle) near the 8 position on the clock face – close but not exact, consistent with the 35% hour accuracy in Tab. 7. (b) The baseline clicks on the emoji button in the text editor, entirely outside the clock widget, illustrating the domain gap on site-specific UI components.

Web checkpoint originally handled. We resolved this by oversampling all eight click-action types during continued LoRA training (up to 25× for the rarest class), ensuring the adapter acquires clock-dial competence without sacrificing general-UI grounding. One gap remains: `click_ampm` was excluded from both training and evaluation because the annotation pipeline did not produce reranker-eligible samples for the AM/PM toggle; the end-to-end demo confirms the adapted model handles this action correctly, but offline quantification is lacking.

End-to-end Demo. Figure 2 shows a representative clock-hour step: the baseline clicks the emoji button (outside the clock entirely), while the adapted LoRA clicks near the correct sector. Once the clock step fails, all downstream actions operate on incorrect UI state, so only the adapted agent completes the full ten-step workflow.

5. Limitations

Task A: The Threads SFT dataset contains only 1,016 pairs scraped from a single account (`theartificialintelligence`); style transfer may not generalise to other tones, languages, or platforms. Evaluation is limited to format compliance (JSON validity, character count) – no human or LLM-as-judge assessment of post quality, factual faithfulness, or engagement potential was performed.

Task B: The Stage 3 reranker is the clear bottleneck at 5.25% exact-match on val, yet we evaluate only a 3B-parameter base model; scaling to 7B or exploring chain-of-thought prompting are untested. The ShowUI-Web test split (3,279 samples from bbc.com, espn.com, steampowered.com) was not evaluated through the full three-stage pipeline, so cross-site generalisation remains unmeasured. Pipeline latency is not benchmarked, limiting practical deployment assessment.

Task C: OOD adaptation is demonstrated on a single production site (Viraly) with a small val set (N=61 covering only 3 of 10 action types); the `click_ampm` action is absent from both training and offline evaluation. No head-to-head comparison with monolithic VLA baselines (e.g. SeeClick, ShowUI) is conducted on the same Viraly data, preventing direct accuracy–cost trade-off analysis. The Viraly training annotations depend on DOM-level element coordinates extracted by Playwright, which assumes access to the page DOM – a privilege unavailable in screenshot-only deployment scenarios.

6. Conclusion

The modular design makes stage-wise error attribution trivial: detector recall saturates near 0.80, the retriever is well-adapted, and the reranker is the remaining bottleneck. OOD results show LoRA adaptation on <2 k records more than quadruples hit@1 on an unseen production site, validating modularity as a practical alternative to retraining a monolithic VLA.

Project Member Contributions

This project is conducted by a single team member, Hui Gi Wai (23444584).

References

[1] Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing GUI grounding for advanced visual GUI agents. In *ACL*, 2024. 1

[2] Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Weixian Lei, Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for GUI visual agent. In *CVPR*, 2025. 1