FOCUSED OMNI-MODAL GUI AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graphical User Interface (GUI) agents increasingly act over streaming omni-modal observations spanning screenshots, audio, video, interaction history, and device states. However, exposing this complete context uniformly at every step ignores that relevance is transient and sparse: the signal determining one action can become clutter at the next. Because executed actions feed back into the context, these errors compound into Sequential Distraction, driving the agent into irreversible trajectory drift. We formulate the underlying requirement as Focused Perception: a representation that is step-adaptive, action-sufficient, and non-intrusive. We realize this in **FocusGUI**, a lightweight adapter that dynamically prioritizes multimodal channels against a step-specific decision reference. Instead of uniformly processing all inputs, it compresses the most relevant evidence into a compact latent and subtly injects it into the frozen decoder's embeddings, guiding the model's attention without altering its native parameter space. FocusGUI trains from action supervision alone, requiring neither modality-relevance labels nor backbone fine-tuning. Across four omni-modal backbones, it raises overall Type Match by $+33.2$ and Exact Match by $3.0\times$ over the frozen backbone on average, and Type Match by $+30.1$ over the strongest existing enhancement method.

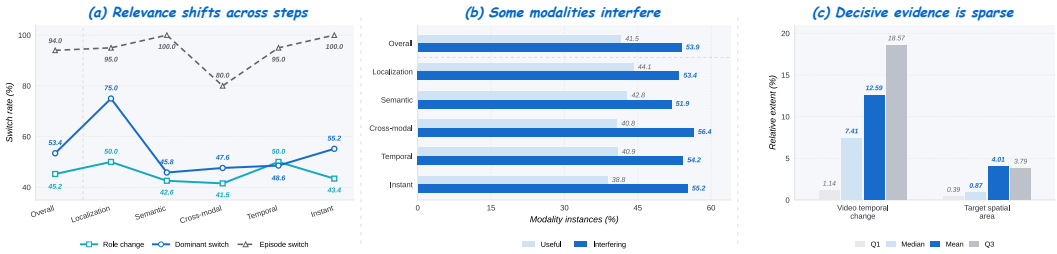

Figure 1: **Relevance is transient across steps and sparse within them.** Across multi-step episodes, we systematically ablate individual modalities from the input of Qwen2.5-Omni and record the resulting change in the log-likelihood of the ground-truth action. A modality is *useful* if its ablation drops this likelihood, and *interfering* if its ablation increases it; the *dominant* modality is the one whose absence causes the steepest drop. (a) The dominant modality switches across 45.3% of adjacent steps, with 94.0% of episodes containing at least one switch. (b) 90.1% of steps contain at least one interfering modality. (c) Based on annotations, the target element occupies a median of a mere 0.87% of the screen area, and video content changes by a median of just 7.41% within a step.

1 INTRODUCTION

Graphical User Interface (GUI) agents are shifting from parsing static screenshots to acting on dynamic, streaming observations across multiple modalities (Henry et al., 2026; Jang et al., 2025; Yao et al., 2026). In realistic computing environments, the evidence needed for the next action need not be confined to the current visual frame. An unexpected notification chime might demand an immediate context switch, a specific transition in a live video can dictate when to click, and spoken instructions must be continuously mapped to visual changes on the screen (ByteDance Seed, 2026; Li et al., 2026; Yuan et al., 2026; Chen et al., 2025). Contemporary omni-modal models can ingest screenshots, video streams, audio, interaction history, and device states within a single network (Xu et al., 2025; Fu et al., 2025; Cui et al., 2026). This has naturally led to a default paradigm: ex-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

posing the complete multimodal observation to the model at every interaction step and leaving it to implicitly determine what matters while generating the action.

Leaving this selection entirely to the decoder assumes that it can reliably rediscover what matters from the complete observation at every interaction step. Our analysis reveals why this is difficult (Fig. 1). When individual modalities are removed during multi-step interaction, the dominant modality switches across nearly half of adjacent steps; even within a single step, only a small fraction of the observation may support the decision, while the rest can become distracting. Relevance is therefore fundamentally a property of the *step*, not a static property of the observation. Yet the decoder is forced to rediscover this relevance at every step, without any explicit signal of what has changed. In a closed-loop agent, such misreads do not end with the current prediction: the executed action changes the environment and feeds back into the observation and history of the next step, allowing errors to compound over time. We refer to this compounding failure as **Sequential Distraction**.

The most intuitive remedy is to perform *subtraction* on the inputs. This is exactly what existing methods attempt: cropping parts of the current screen to retain only candidate regions of interest (Lu et al., 2024; Luo et al., 2025; Liu et al., 2026b; Cheng et al., 2024). However, this strategy of giving less fundamentally misaligns with the nature of streaming interaction in two ways. First, the decisive signal could emerge from audio, video, or device status at any moment, channels entirely outside the scope of spatial cropping. Second, even when the signal resides on the screen, the crudely cropped background often contains the essential contextual evidence needed to ground accurate coordinates and text parameters. The problem is not that the decoder is given too much, but that it lacks guidance on how to prioritize it. A proper focus mechanism should change how the decoder *reads* the observation, rather than altering what it *is given*.

We realize this principle in **FocusGUI** (Fig. 2). Since the decoder can already read the full observation and what it lacks is a signal of what matters at the current step, FocusGUI leaves both the backbone and its input untouched and supplies that signal alongside them. At every step, it exposes the observation as separable perceptual channels, scores each against a reference derived from the current observation as a whole, and compresses the prioritized channels into a compact focus latent. The latent enters as a zero-initialized residual at the boundary where the decoder begins to generate the action. This residual reallocates attention over the entire observation without altering it, guiding both the choice of action and the grounding of its arguments. Training uses next-action supervision alone, with no labels indicating which modality matters.

On average across four omni-modal backbones, FocusGUI raises Type Match from 31.0% to 64.2% and Exact Match from 11.2% to 33.5% over the frozen backbone, and reaches 1.9× the Type Match of the strongest enhancement method. The gains grow with interaction depth, as Sequential Distraction predicts: from the first step to the fourth, the frozen Qwen2.5-Omni loses half of its Type Match while FocusGUI retains 86.8%. Our contributions are as follows:

- An empirical account of **Sequential Distraction**: in multi-step omni-modal GUI interaction, relevance is a step-level property, transient across steps and sparse within them, and because actions re-enter the observation, per-step misreads compound into trajectory drift.
- A distinction between the evidence an agent is given and the way its decoder reads it, showing why spatial refinement cannot satisfy both the selection of an action and the grounding of its arguments, and deriving from this the form a solution must take.
- **FocusGUI**, a zero-initialized residual adapter at the action boundary of a frozen backbone, trained from action supervision alone, whose gains across four backbones and five task categories are distributed as the account predicts.

2 METHOD

Problem Formulation and Context-Preserving Focus At the interaction step t, the agent receives an omni-modal observation $O_t = \{g, I_t, V_t, A_t, H_t, S_t\}$, consisting of the user goal g, current interface I_t, temporal visual observation V_t, acoustic observation A_t, interaction history H_t, and device state S_t. It predicts an executable action $y_t = (y_{t,1}, \ldots, y_{t,L_t})$, represented as a token sequence specifying both the action primitive and its arguments. Let F_{Θ_0} denote a pretrained omni-modal backbone whose parameters Θ_0 remain frozen, and let $E_t \in \mathbb{R}^{N_t \times d_{\text{model}}}$ denote the embedding se-

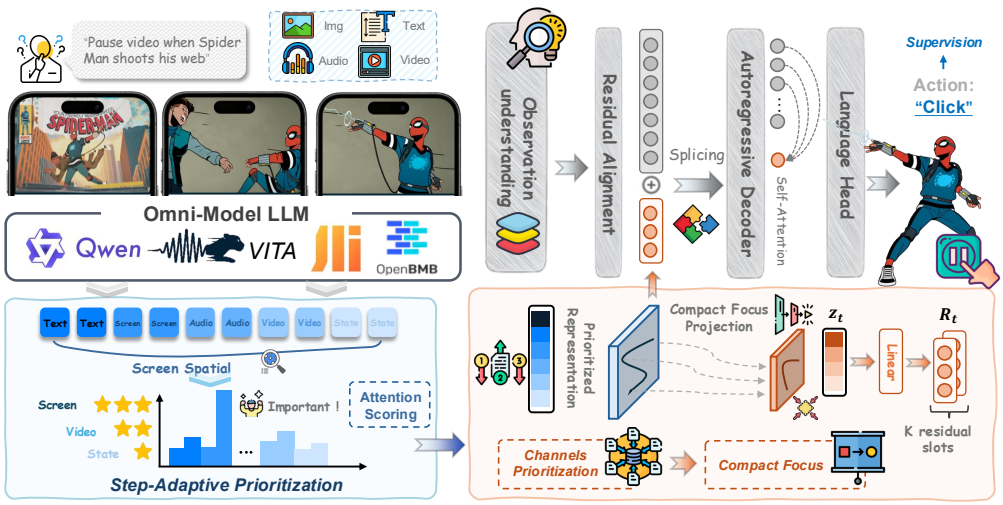

Figure 2: **FocusGUI.** The frozen backbone's original pathway is left intact (top). From the same observation, FocusGUI exposes perceptual channels, scores them against a decision reference recomputed at every step, compresses the result into a compact focus latent, and adds it as a zero-initialized residual at the K positions preceding the action boundary.

quence presented to its decoder. Section 1 committed to realizing focus in how the decoder reads rather than on what it is shown, and that commitment fixes the form of the mechanism. FocusGUI neither replaces nor prunes any part of O_t. From the same observation, it constructs a compact focus representation $z_t = f_\phi(O_t)$, maps it to a residual $R_t = A_\phi(z_t)$, and adds that residual to a small set of existing embedding positions. The decoder therefore operates on

$$\widetilde{E}_t = E_t + \mathcal{S}_{J_t}(\gamma R_t), \qquad p_{\Theta_0,\phi}(y_t \mid O_t) = p_{\Theta_0}(y_t \mid \widetilde{E}_t), \tag{1}$$

where $\mathcal{S}_{J_t}(\cdot)$ places the residual at positions J_t and is zero elsewhere. This separates *observation preservation* from *decision guidance*. The original pathway retains the complete multimodal context, while the FocusGUI pathway learns how that context should be emphasized for the next action. Two properties of z_t follow from this division of labor. It is not required to reconstruct the observation or to encode any execution detail, since every pixel, coordinate, and transcript remains available to the decoder through the unmodified pathway. It can therefore be made small, because its role is to condition a read rather than to carry evidence. Appendix G shows that such a residual selects the primitive by changing what the decoder reads, but corrects arguments only by reallocating attention over evidence positions Eq. (1) leaves untouched.

Step-Adaptive Evidence Organization A priority cannot be defined inside a fused representation, so the observation must first be exposed as separable evidence. We write this as perceptual channels $X_t = \{x_t^{(m)}\}_{m\in\mathcal{M}}$, where $\mathcal{M}$ indexes the sources' FocusGUI scores, including the backbone's integrated hidden state and a spatial screen channel alongside the individual modalities. Each channel is mapped into a shared d-dimensional space through a source-specific projector, $e_t^{(m)} = P_m(x_t^{(m)}) \in \mathbb{R}^d$. The separate projectors accommodate the native statistics of different sources while making their representations comparable. The usefulness of a channel, however, depends not only on what it contains but also on the surrounding task state. We therefore derive a decision reference $q_t = P_q(e_t^{(\mathrm{all})})$ from the integrated representation of the complete observation. Here, $e_t^{(\mathrm{all})}$ provides a *joint semantic context*: it summarizes the current observation against which source-specific evidence is evaluated. For each channel m, FocusGUI computes its compatibility with this joint context and normalizes the resulting scores into step-specific priorities:

$$\begin{aligned} s_t^{(m)} &= v^\top \tanh\Big(W_e(e_t^{(m)} + r_m) + W_q q_t\Big), \\ \alpha_t^{(m)} &= \frac{\exp(s_t^{(m)}/\tau)}{\sum_{j\in\mathcal{M}} \exp(s_t^{(j)}/\tau)}, \qquad c_t = \sum_{m\in\mathcal{M}} \alpha_t^{(m)} e_t^{(m)}, \end{aligned} \tag{2}$$

where r_m is a learned channel-identity embedding and τ is the softmax temperature. Because both the perceptual channels and the joint semantic context are recomputed at every interaction step, the relative emphasis among information sources can change as the task evolves. Nothing in Eq. (2) carries state across steps: drift is produced one step at a time, so removing the corruption at each step removes its accumulation. The priorities $\alpha_t^{(m)}$ should be interpreted as how FocusGUI organizes evidence within its auxiliary pathway, rather than as a complete explanation of how the frozen backbone internally uses each modality.

Compact Focus and Residual Alignment The prioritized representation c_t determines how different evidence sources contribute to the auxiliary pathway, but still lies in the full shared channel space. We therefore compress it into a lower-dimensional focus representation: $z_t = P_s(c_t) \in \mathbb{R}^{d_z}, \quad d_z < d$. This projection constrains the capacity of the *auxiliary conditioning signal*, rather than the information accessible to the backbone. Because the complete observation remains available through the original multimodal pathway, z_t need not independently encode every visual detail, coordinate, or text argument required for execution. Instead, it provides a compact summary of what should receive greater emphasis for the current action. To make this focus representation directly usable by the frozen decoder, we align it with the decoder's native embedding space at the action boundary. Let b_t denote the position at which action generation begins, and let $J_t = \{b_t - K, \ldots, b_t - 1\}$ denote the K preceding embedding positions. FocusGUI expands z_t into a residual over this window and adds it to the existing embeddings:

$$R_t = \mathrm{Reshape}(W_p z_t + b_p) \in \mathbb{R}^{K \times d_{\mathrm{model}}}, \qquad \widetilde{E}_{t,J_t} = E_{t,J_t} + \gamma R_t. \tag{3}$$

Injecting the residual immediately before action generation makes the focus signal available to all subsequent action tokens without introducing a separate prediction head or modifying the output vocabulary. The additive form is equally important: it preserves the original embeddings and uses the focus representation only as an additional conditioning signal. We initialize W_p and b_p to zero, such that $R_t = 0$ and $\widetilde{E}_t = E_t$ at the beginning of training. FocusGUI therefore starts from the original frozen-model input and gradually learns observation-conditioned deviations through optimization. Importantly, J_t consists of existing prompt positions: FocusGUI neither appends textual hints nor removes any multimodal input.

Learning Focus from Action Supervision FocusGUI requires no annotation specifying which modality or perceptual channel should be emphasized. Instead, all components are learned jointly from supervision on the next executable action. Let ϕ denote the trainable parameters of the channel projectors, step-adaptive prioritization, compact projection, and residual alignment modules. We optimize the average action-token negative log-likelihood

$$\mathcal{L}_{\mathrm{act}}(\phi) = -\frac{1}{|\mathcal{D}_{\mathrm{tr}}|} \sum_{t \in \mathcal{D}_{\mathrm{tr}}} \frac{1}{L_t} \sum_{k=1}^{L_t} \log p_{\Theta_0,\phi}\left(y_{t,k} \mid y_{t,<k}, O_t\right). \tag{4}$$

The loss is evaluated only on action tokens. Gradients propagate through the frozen decoder to ϕ, while the omni-modal backbone, decoder, and language-model head receive no parameter updates. Thus, a single action-generation objective jointly learns *which evidence to emphasize*, *how to summarize it*, and *how to express that summary in the decoder's native representation space*. At inference time, FocusGUI constructs the perceptual channels from the current observation, evaluates them under the current joint semantic context, forms the compact focus representation, and generates the corresponding residual. The frozen decoder then autoregressively produces the next executable action through its original generation pathway.

3 Experiments

3.1 Experimental Setup

Dataset and Metrics. We evaluate on OmniGUI (Henry et al., 2026), a benchmark of omni-modal smartphone GUI interactions across 29 applications; we use 708 episodes with 2,572 action steps. Each step pairs a screenshot with synchronous audio, video, and action history, so the decisive signal

Table 1: Main results across omni-modal backbones and GUI task categories. FocusGUI rows are highlighted, and the best score in each column within each backbone block is bolded.

	Overall			Localization			Semantic			Cross-modal			Temporal			Instant		
Methods	TM↑	EM↑	GP↑	TM↑	EM↑	GP↑	TM↑	EM↑	GP↑	TM↑	EM↑	GP↑	TM↑	EM↑	GP↑	TM↑	EM↑	GP↑
Qwen2.5-Omni	49.61	12.60	5.50	57.32	12.20	4.94	48.65	17.12	6.75	42.20	9.17	3.17	45.97	12.10	7.40	58.54	12.20	5.07
+DRS-GUI	49.41	6.69	2.97	51.22	6.10	3.63	50.45	5.41	1.23	45.87	5.50	2.79	49.19	9.68	4.95	51.22	6.10	1.87
+OmniParser	48.43	7.68	3.87	51.22	7.32	4.51	49.55	7.21	1.87	44.95	5.50	2.41	46.77	10.48	5.83	51.22	7.32	4.48
+RegionFocus	48.23	7.68	4.07	52.44	4.88	3.82	49.55	7.21	2.65	44.04	6.42	3.46	46.77	11.29	5.21	50.00	7.32	5.13
+GUI-Actor	50.20	13.78	5.54	58.54	13.41	5.18	47.75	18.02	6.36	43.12	10.09	3.63	46.77	12.90	7.08	59.76	14.63	5.47
+FocusGUI	**71.65**	**40.16**	**17.49**	**78.05**	**45.12**	**30.06**	**68.47**	**39.64**	**14.24**	**67.89**	**33.03**	**7.47**	**76.61**	**46.77**	**26.35**	**67.07**	**35.37**	**7.97**
VITA-1.5	13.98	9.65	6.09	12.20	7.32	**7.04**	19.82	13.51	6.59	11.01	6.42	2.69	12.90	9.68	8.85	13.41	10.98	**5.07**
+DRS-GUI	10.43	6.30	3.14	8.54	3.66	1.26	9.01	6.31	3.05	10.09	3.67	1.76	12.90	10.48	5.58	10.98	6.10	3.89
+OmniParser	17.13	7.48	3.82	18.29	3.66	1.26	17.12	7.21	3.77	14.68	4.59	2.93	17.74	12.10	6.40	18.29	8.54	4.59
+RegionFocus	12.20	7.68	3.69	9.76	4.88	1.44	10.81	7.21	3.63	10.09	4.59	2.75	15.32	12.10	5.77	14.63	8.54	4.78
+GUI-Actor	14.57	11.02	5.39	17.07	9.76	1.23	22.52	18.02	**11.14**	9.17	6.42	**3.82**	12.90	11.29	5.94	10.98	8.54	4.35
+FocusGUI	**58.66**	**26.38**	**6.84**	**56.10**	**18.29**	6.79	**54.05**	**28.83**	8.76	**55.96**	**24.77**	3.71	**64.52**	**30.65**	**9.69**	**62.20**	**26.83**	4.71
MiniCPM-o-4.5	36.02	13.58	7.55	35.37	13.41	9.97	35.14	9.91	5.15	35.78	15.60	3.98	34.68	12.90	11.04	40.24	17.07	7.68
+DRS-GUI	36.42	12.60	5.47	37.80	10.98	3.40	36.94	14.41	5.24	31.19	8.26	4.41	37.90	16.94	8.62	39.02	10.98	5.37
+OmniParser	37.20	12.40	6.43	41.46	9.76	4.84	36.04	12.61	5.84	30.28	8.26	5.02	38.71	16.94	9.26	41.46	13.41	6.96
+RegionFocus	37.01	13.19	6.69	40.24	10.98	5.62	36.94	15.32	7.27	30.28	9.17	5.32	39.52	17.74	9.32	39.02	10.98	5.57
+GUI-Actor	36.61	14.17	7.64	36.59	14.63	9.47	34.23	10.81	5.63	36.70	16.51	4.23	35.48	12.10	10.84	41.46	18.29	8.03
+FocusGUI	**67.72**	**41.54**	**16.76**	**81.71**	**47.56**	**21.73**	**68.47**	**45.05**	**19.94**	**60.55**	**34.86**	**9.15**	**67.74**	**41.13**	**18.68**	**62.20**	**40.24**	**14.49**
Baichuan-Omni-1.5	24.41	8.86	4.32	26.83	4.88	0.00	16.22	8.11	1.72	22.02	11.93	6.77	28.23	10.48	7.66	30.49	7.32	4.71
+DRS-GUI	27.76	5.12	2.26	32.93	3.66	2.76	26.13	5.41	2.08	21.10	2.75	1.19	29.03	8.06	2.89	31.71	4.88	2.29
+OmniParser	32.09	6.10	2.21	37.80	4.88	3.01	31.53	6.31	0.78	26.61	5.50	1.82	30.65	8.06	3.26	36.59	4.88	1.96
+RegionFocus	24.61	5.51	2.47	28.05	3.66	2.41	22.52	5.41	1.57	19.27	3.67	0.87	25.81	8.06	4.45	29.27	6.10	2.84
+GUI-Actor	31.10	11.02	5.35	34.15	8.54	2.91	24.32	10.81	3.11	28.44	12.84	**6.92**	33.06	12.10	8.57	37.80	9.76	**5.24**
+FocusGUI	**58.86**	**25.98**	**8.73**	**56.10**	**18.29**	**3.70**	**58.56**	**33.33**	**15.27**	**55.05**	**21.10**	4.09	**62.10**	**31.45**	**15.21**	**62.20**	**21.95**	3.62

can come from any of the four streams. We report Type Match (TM), Exact Match (EM), and Goal Progress (GP). TM checks whether the predicted action primitive matches the ground truth; EM further requires predicted coordinates to fall within the target element's bounding box and generated text to match the reference. GP is the fraction of correctly executed steps per episode, averaged across episodes.

Models. We evaluate four representative open-source omni-modal backbones: Qwen2.5-Omni (Xu et al., 2025), VITA-1.5 (Fu et al., 2025), MiniCPM-o 4.5 (Cui et al., 2026), and Baichuan-Omni-1.5 (Li et al., 2025b). The four backbones support textual, visual, auditory, and temporal inputs but differ in architecture and multimodal representation, which lets the comparison probe the generality of FocusGUI across backbone families. For each backbone, the base model stays frozen, and training updates only the lightweight FocusGUI head and action predictors, using the same episode-disjoint split across all four models.

Baselines. We compare against four methods, each applied to all four backbones without modifying their weights. DRS-GUI (Liu et al., 2026b), OmniParser (Lu et al., 2024), and RegionFocus (Luo et al., 2025) refine the visual input at inference by searching over candidate regions, parsing the screenshot into structured elements, and zooming into attended regions. GUI-Actor (Wu et al., 2025a) instead learns an action head over the backbone's visual tokens, trained on the same annotations and split as FocusGUI with the backbone frozen. All methods receive the same observation and decoding procedure, differing only in how that observation is organized before the action is generated. The trained head separates two explanations for any gain: trainable parameters on this data, which GUI-Actor also has, and step-adaptive organization, which FocusGUI adds.

Implementation Details. Evidence channels are projected into a shared 128-dimensional space and weighted by the full-modal query with softmax temperature $\tau = 1.0$; the weighted representation is compressed to a 64-dimensional action-sufficient embedding and expanded into $K = 4$ zero-initialized residual tokens added to the last four prompt embeddings before the action bound-

Table 2: Component ablation on Qwen2.5-Omni. CD: channel decomposition. SP: step-adaptive prioritization. AP: action-sufficient projection. RA: residual alignment. Each task category reports Type Match (TM), Exact Match (EM), and Goal Progress (GP).

Components				Overall			Localization			Semantic			Cross-modal			Temporal			Instant		
CD	SP	AP	RA	TM↑	EM↑	GP↑	TM↑	EM↑	GP↑	TM↑	EM↑	GP↑	TM↑	EM↑	GP↑	TM↑	EM↑	GP↑	TM↑	EM↑	GP↑
✗	✗	✗	✗	49.61	12.60	5.50	57.32	12.20	4.94	48.65	17.12	6.75	42.20	9.17	3.17	45.97	12.10	7.40	58.54	12.20	5.07
✗	✗	✓	✓	66.26	35.00	12.20	68.49	40.56	14.55	66.67	33.64	11.28	60.55	30.86	6.58	72.58	38.58	20.00	60.29	31.46	7.13
✓	✗	✓	✓	67.38	37.78	11.56	71.95	42.68	16.82	64.86	37.84	9.65	64.22	32.04	6.35	74.42	45.97	18.02	59.76	28.05	5.80
✓	✓	✗	✓	71.14	38.75	16.11	78.05	42.68	25.43	66.67	37.84	13.84	67.72	32.70	6.54	78.23	47.58	27.50	65.03	32.93	7.25
✓	✓	✓	✗	70.45	34.86	12.11	74.15	41.34	28.09	67.57	33.33	13.43	67.56	32.04	5.05	79.84	39.52	10.73	63.11	28.05	3.26
✓	✓	✓	✓	**71.65**	**40.16**	**17.49**	**78.05**	**45.12**	**30.06**	**68.47**	**39.64**	**14.24**	**67.89**	**33.03**	**7.47**	**76.61**	**46.77**	**26.35**	**67.07**	**35.37**	**7.97**

ary, introducing 1.15M–1.45M trainable parameters depending on the backbone. With the encoder, decoder, and LM head frozen, we train only these modules for three epochs with AdamW under the teacher-forced action-token negative log-likelihood, selecting the learning rate 2×10^{-4} and weight decay 10^{-3}. At inference, the frozen decoder generates actions autoregressively with greedy decoding. Additional implementation details are provided in the Appendix D.

3.2 Main Results

Consistent gains across backbones. Table 1 evaluates four omni-modal backbones under identical frozen weights. Averaged across them, FocusGUI adds 33.2 TM, 22.3 EM, and 6.6 GP points over direct inference and 30.1, 23.6, and 8.0 points over the strongest baseline at each backbone. It holds the best TM and the best EM in all 24 backbone-category cells.

Why screen refinement backfires. The baselines fail in a way that is diagnostic rather than incidental. DRS-GUI, OmniParser, and RegionFocus are near-neutral on TM (+0.9 points on average) yet lower EM by 28% and GP by 35% in relative terms, in 12 of 12 cases for each metric. Narrowing a screenshot makes the action primitive more salient, which is what TM rewards, while discarding the surrounding context that fixes coordinates and text, which is what EM requires. None of the three reaches audio, video, or device state, so a decision resting on those channels stays unrecoverable however the screen is processed. This is what our framing predicts: once relevance is a property of a step rather than of a region, refining the region cannot restore it and may cost more than it returns.

Where the gains come from. The structure of the gains indicates what the representation supplies. Per-category TM gains run from 37.3 points on Temporal, where the decisive signal persists across frames and a single screenshot serves worst, to 27.8 on Instant, where the decision is reactive and largely present on the current screen; EM follows the same order. Across backbones, the overall TM spread narrows from 35.6 points to 13.0, indicating that choosing the correct primitive becomes substantially less backbone-dependent under a better-organized observation. EM variation instead reflects backbone-specific argument-generation capacity. At episode level, improvements in per-step prediction translate into Goal Progress when they are sustained across the trajectory. Episode progress is gated by the weakest step in a trajectory.

3.3 Ablation Study

All four components contribute, mostly to Goal Progress. Table 2 removes each component from the full model. Overall Type Match falls by 0.51, 1.20, 4.27, and 5.39 points when action-sufficient projection, residual alignment, step-adaptive prioritization, and channel decomposition are removed, but GP falls by 1.38, 5.38, 5.93, and 5.29 points on a base roughly a quarter as large. In relative terms, every component is worth four to five times more to episode progress than to single-step action type: at most 7.5% of TM against up to 33.9% of GP. Their disproportionately larger effect on GP shows that the components primarily keep a trajectory from unravelling, which is what Sequential Distraction predicts.

GP reveals contributions beyond TM. Decomposing the observation into channels without prioritizing them raises Type Match by 1.12 and Exact Match by 2.78 points over a single integrated channel. Prioritization converts this richer channel access into episode-level stability by deciding

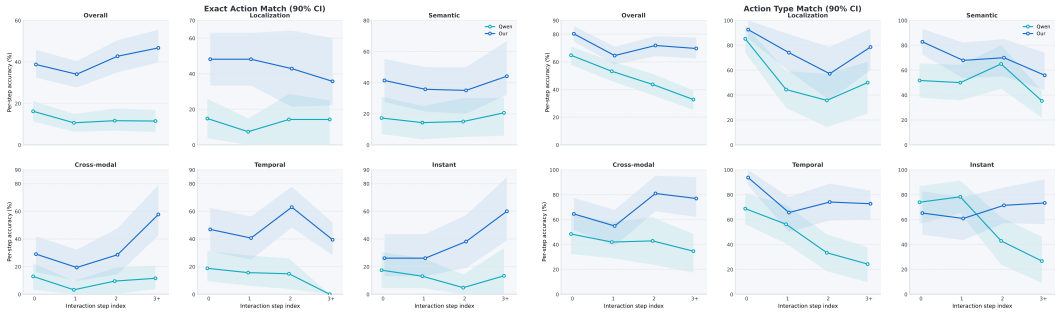

Figure 3: **FocusGUI holds accuracy as trajectories lengthen.** Per-step Exact Match (left) and Type Match (right) on Qwen2.5-Omni, with steps binned by their index within the episode and 3+ collecting the fourth step onward. Shaded regions are 90% episode-clustered bootstrap intervals.

which channel to trust. Residual alignment shows the same metric asymmetry more sharply: replacing the additive residual with prompt-embedding substitution reduces EM by 5.30 and GP by 5.38 points, compared with a 1.20-point TM change. Preserving the original prompt embeddings therefore matters most for grounding the selected action primitive into correct arguments.

3.4 Performance Across Interaction Depth

Sequential Distraction predicts that interference accumulates. A method that suppresses it should separate from the backbone as a trajectory lengthens, whereas a uniformly stronger predictor would hold a constant margin. We bin every evaluation step by its index within the episode and report per-step accuracy with 90% episode-clustered bootstrap intervals (Fig. 3).

Qwen2.5-Omni falls from 64.79 TM at the first step to 32.79 beyond the third, retaining half of its initial accuracy. FocusGUI retains 86.8% over the same range, and its exact-match accuracy rises rather than falls, from 38.73 to 46.72. The margin therefore widens with depth: EM gains increase monotonically from $+22.54$ to $+35.25$, and all intervals exclude zero. This locates the episode-level improvement of Table 1: Goal Progress does not rise evenly along a trajectory but comes predominantly from steps the frozen backbone has already lost. The effect is sharpest under temporal reasoning, where the backbone scores 0.00 EM beyond the third step and FocusGUI reaches 39.39.

Instant-response tasks provide a negative control. At the first two steps, where these reactive tasks carry little accumulated history, FocusGUI is statistically indistinguishable from the backbone (-8.70 and -17.39 TM; both 90% intervals include zero). At depths 2 and 3+, once interaction history can accumulate, FocusGUI gains $+28.57$ and $+46.67$ TM. This depth-dependent separation is the behavior expected of a mechanism targeting accumulation, rather than a uniformly stronger predictor. Restricting the analysis to episodes with at least four steps holds the episode set constant across bins, and the widening margin persists.

Learned priorities reorder across an episode. Figure 4 averages $\alpha_t^{(m)}$ over test steps grouped by interaction phase. Interaction state receives 30.9% of the priority mass at the opening step, more than twice a uniform share, and falls to 7.7% by the closing phase, moving from the highest-priority channel to the lowest; audio and video move the other way, rising from 13.4% and 11.3% to 18.8% and 18.3%. No fixed weighting reproduces both ends of that range; only a representation that reweights its inputs as the decision changes can, which is exactly the step-adaptive behavior Focused Perception requires. The screen is never dominant: it stays between 10.9% and 12.8%, below

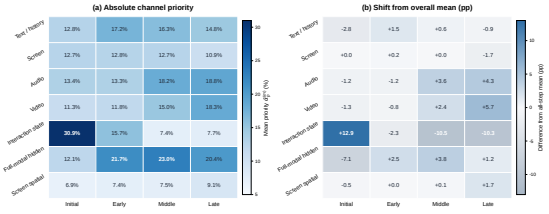

Figure 4: **Learned channel priorities reorder across interaction phases.** (a) Mean priority $\bar{\alpha}^{(m)}$ per channel and phase on Qwen2.5-Omni; uniform allocation would be 14.3%. (b) Deviation of each cell from that channel's all-step mean. FocusGUI trains on action annotations alone, with no modality-relevance supervision.

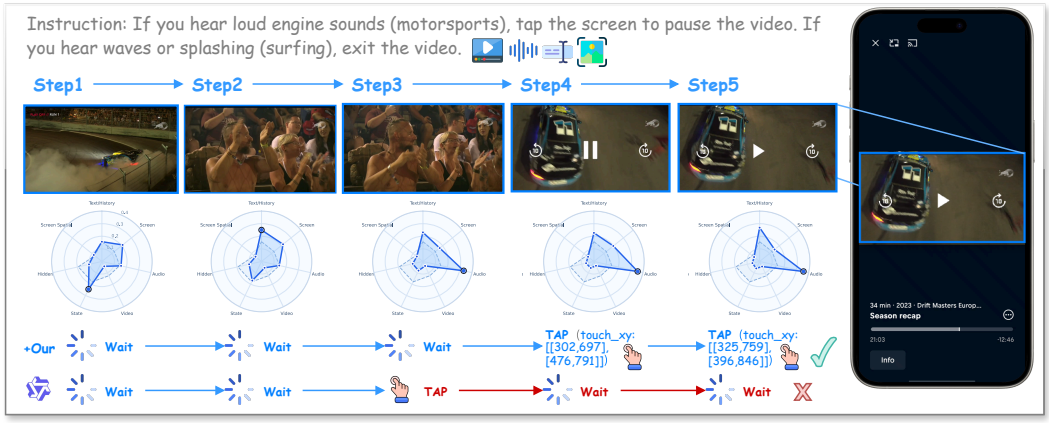

Figure 5: **Case study: a decision that never appears on the screen.** The instruction conditions the action on what is heard, so the screen looks the same at every step while the deciding evidence sits in audio. FocusGUI waits and taps once the engine sounds arrive; the frozen backbone acts early on visual salience and cannot recover, since its own action re-enters the context of every later step. Radar plots show the channel priorities $\alpha_t^{(m)}$ per step.

uniform at every phase, and the spatial screen channel never exceeds 9.1%. Prior work refines exactly these two channels, while the channels that gain priority late are the streaming ones it cannot reach. None of this is supervised: the ordering is recovered from action annotations alone, and Fig. 5 traces the same shift within a single episode.

Capacity sensitivity. Table 3 varies each capacity parameter while holding the other at its default. Across a sixteenfold range of K, Overall TM spans 2.6 points, EM 2.4, and GP 3.0, with no trend in any of them. This matches the role K plays in the design: R_t is a deterministic function of z_t, so a wider boundary window redistributes the same compressed representation over more positions without admitting more of the observation. Capacity is set by d_z instead, and the sweep locates where it binds. At $d_z = 32$ the latent is too small to carry the action, costing 6.9 EM points and 6.5 GP points against the default. From 64 upward the method is stable, with Overall TM spanning 1.6 points and Exact Match highest at the default. FocusGUI requires no per-backbone or per-benchmark search over either parameter, and we fix $K = 4$ and $d_z = 64$ throughout. For more comprehensive evaluation results, including performance across interaction depths on other backbones and capacity sensitivity analyses, please refer to the Appendix C.

Table 3: Sensitivity to the boundary window K and the compact focus dimension d_z on Qwen2.5-Omni. The default ($K = 4$, $d_z = 64$, shaded) is fixed across all backbones and never tuned per benchmark.

	Overall TM	Overall EM	Overall GP	Temporal GP
Boundary window K ($d_z = 64$)				
$K = 1$	72.83	39.76	15.13	17.66
$K = 4$	71.65	40.16	17.49	26.35
$K = 8$	70.28	37.80	15.72	16.04
$K = 16$	71.65	39.76	18.09	25.10
Action-sufficient dimension d_z ($K = 4$)				
$d_z = 32$	70.47	33.27	10.95	12.45
$d_z = 64$	71.65	40.16	17.49	26.35
$d_z = 128$	72.83	37.99	15.05	15.73
$d_z = 256$	71.23	38.19	18.28	15.83

4 Related Work

4.1 GUI Agents in Omni-Modal Environments

GUI agents interact with digital environments by repeatedly translating user intentions and interface observations into executable actions. Research across mobile, web, and desktop platforms has established this perception–action loop and progressively extended it from isolated grounding decisions

to long-horizon interaction (Rawles et al., 2023; Deng et al., 2023; Xie et al., 2024; Cheng et al., 2024; Hong et al., 2024; Zhang et al., 2025; Chen et al., 2026b). Most agents nevertheless organize their observation around the current screenshot, with previous actions or screens retained as auxiliary context. Temporal GUI benchmarks broaden this view by requiring agents to understand dynamic visual content (Jang et al., 2025; Zhang et al., 2026b; Chen et al., 2025), while OmniGUI further places each action within an interleaved observation of screen, video, audio, and interaction history (Henry et al., 2026). These developments shift GUI interaction from processing a sequence of interfaces to acting over an evolving omni-modal state.

In parallel, omni-modal foundation models increasingly support unified or streaming perception of text, vision, and audio (Xu et al., 2025; Fu et al., 2025; Cui et al., 2026). Broader evaluations of real-time assistants similarly emphasize online processing and continuity as multimodal events unfold (Lu et al., 2026). Such capabilities determine what a model can receive; how an agent should organize that evidence for the next GUI action remains open. The closest observation-side approach augments computer-use agents with gated keyframes, audio transcription, and persistent visual narration (Li & Shi, 2026). FocusGUI addresses the complementary problem inside an omni-modal agent: given the complete observation, it learns which evidence should shape the current action and how to deliver that evidence to a frozen decoder.

4.2 Perception and Grounding for GUI Agents

Perception is central to GUI agents because an instruction must ultimately be connected to an actionable interface element. One line of work strengthens this connection through GUI-specific representation learning and large-scale grounding data (Cheng et al., 2024; Hong et al., 2024; Gou et al., 2025; Wu et al., 2025b); another exposes explicit interface structure to general-purpose models through screen parsing (Lu et al., 2024; Liu et al., 2026a; Fu et al., 2026). Recent visual agents also reduce redundant screen tokens while jointly learning grounding and action generation (Lin et al., 2025; Liu et al., 2026c). Together, these methods improve how an agent understands the current interface and determines where an action should be executed.

A complementary direction makes perception selective. Rather than processing the full screenshot uniformly, visual search and test-time refinement narrow the observation to promising regions (Li et al., 2025a; Luo et al., 2025; Liu et al., 2026d); active perception methods further decide when and how to crop, zoom, shift, or expand the visual field (Chen et al., 2026a; Liu et al., 2026b). These approaches demonstrate the value of suppressing irrelevant interface content, but their primary control axis remains spatial: they determine which part of the current screen deserves closer inspection (Zhang et al., 2026a). Omni-modal interaction introduces an additional axis, since relevance also changes across information sources and action steps. FocusGUI extends focused perception from spatial localization to step-dependent representation construction over the complete observation, complementing existing GUI grounding mechanisms.

5 Conclusion

We identify Sequential Distraction as a primary challenge in multi-step omni-modal GUI interaction, where unfiltered observations can lead to compounding errors due to transient and sparse relevance. To address this, we formulate the concept of Focused Perception and implement it through FocusGUI, a lightweight adapter that dynamically prioritizes perceptual channels, compresses them into an action-sufficient latent, and delivers the result via a zero-initialized residual at the action boundary. This method relies solely on action annotations and leaves the underlying backbone completely frozen. Evaluations across four backbones demonstrate that FocusGUI substantially improves Type Match over existing enhancement methods and maintains high accuracy at deeper interaction steps where historical context typically degrades. While our current prioritization depends on the present observation, future work could extend this mechanism to anticipate channel relevance, which may help mitigate trajectory drift earlier in the interaction sequence.

REPRODUCIBILITY STATEMENT

We provide the details necessary to reproduce our experiments throughout the main paper and appendices. Specifically, we describe the dataset and evaluation metrics, the train/validation/test splits, model backbones and baselines, training objectives, optimization hyperparameters, decoding settings, and hardware configuration. Appendix D further specifies the system prompt, action space, per-step input template, controlled inputs, and the complete default configuration of FocusGUI. We use the same episode-disjoint data split and evaluation protocol across all compared backbones and methods, and keep the underlying omni-modal backbones frozen when training FocusGUI. These implementation details are intended to facilitate faithful reproduction of the reported results.

AI USE STATEMENT

Generative AI tools were used solely for language polishing and literature retrieval during the preparation of this manuscript. All retrieved references were independently checked by the authors against the original sources, and the authors take full responsibility for the content of the manuscript.

REFERENCES

ByteDance Seed. Seedrealtime audio-visual full-duplex llm released: Toward omni-modal natural interaction. ByteDance Seed Blog, August 2026. URL https://seed.bytedance.com/en/blog/seedrealtime-audio-visual-full-duplex-llm-released-toward-omni-modal-natural-interaction. Accessed: 2026-08-30.

Chen Chen, Jiawei Shao, Dakuan Lu, Haoyi Hu, Xiangcheng Liu, Hantao Yao, and Wu Liu. Gui-eyes: Tool-augmented perception for visual grounding in gui agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29350–29358, 2026a.

Dongping Chen, Yue Huang, Siyuan Wu, Jingyu Tang, Huichi Zhou, Qihui Zhang, Zhigang He, Yilin Bai, Chujie Gao, Liuyi Chen, et al. Gui-world: A video benchmark and dataset for multimodal gui-oriented understanding. In *International Conference on Learning Representations*, volume 2025, pp. 22812–22864, 2025.

Yurun Chen, Biao Yi, Keting Yin, and Shengyu Zhang. Skillharness: Harnessing safe skills for computer-use agents. *arXiv preprint arXiv:2606.20636*, 2026b.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9313–9332, 2024.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, Jiancheng Gui, Luoyuan Zhang, Xian Sun, Fuwei Huang, Moye Chen, Zhuo Lin, Hanyu Liu, Qingxin Gui, Qingzhe Han, Yuyang Wen, Huiping Liu, Rongkang Wang, Yaqi Zhang, Hongliang Wei, Chi Chen, You Li, Kechen Fang, Jie Zhou, Yuxuan Li, Guoyang Zeng, Chaojun Xiao, Yankai Lin, Xu Han, Maosong Sun, Zhiyuan Liu, and Yuan Yao. Minicpm-o 4.5: Towards real-time full-duplex omni-modal interaction, 2026. URL https://arxiv.org/abs/2604.27393.

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. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

Chaoyou Fu, Haojia Lin, Xiong Wang, Yi-Fan Zhang, Yunhang Shen, Xiaoyu Liu, Haoyu Cao, Zuwei Long, Heting Gao, Ke Li, Long Ma, Xiawu Zheng, Rongrong Ji, Xing Sun, Caifeng Shan, and Ran He. Vita-1.5: Towards gpt-4o level real-time vision and speech interaction, 2025. URL https://arxiv.org/abs/2501.01957.

Zichuan Fu, Shirong Wang, Wenlin Zhang, Guojing Li, Yimin Deng, Jingtong Gao, Junjia Qi, Hanyu Yan, Yefeng Zheng, Xiaopeng Li, Wanyu Wang, Xian Wu, and Xiangyu Zhao. Gui-lens: Coarse-to-fine cropping for gui grounding with general-purpose vlms, 2026. URL https://arxiv.org/abs/2608.03270.

Boyu Gou, Demi Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for gui agents. In *International Conference on Learning Representations*, volume 2025, pp. 30851–30883, 2025.

Felix Henry, Xiaochen Lin, Jiangyou Zhu, Yangfan, Bingqian Zhang, Min Chen, and Shiyu Huang. Omnigui: Benchmarking gui agents in omni-modal smartphone environments, 2026. URL https://arxiv.org/abs/2605.18758.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14281–14290, 2024.

Lawrence Jang, Yinheng Li, Dan Zhao, Charles Ding, Justin Lin, Paul Pu Liang, Rogerio Bonatti, and Kazuhito Koishida. Videowebarena: Evaluating long context multimodal agents with video understanding web tasks. In *International Conference on Learning Representations*, volume 2025, pp. 36934–36958, 2025.

Bojie Li and Noah Shi. Agent-computer observation interfaces enable dynamic computer use. *arXiv preprint arXiv:2606.29472*, 2026.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: Gui grounding for professional high-resolution computer use. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 8778–8786, 2025a.

Wanli Li, Bowen Zhou, Yunyao Yu, Zhou Xu, Yifan Yang, Dongsheng Li, and Caihua Shan. Weavebench: A long-horizon, real-world benchmark for computer-use agents with hybrid interfaces, 2026. URL https://arxiv.org/abs/2606.09426.

Yadong Li, Jun Liu, Tao Zhang, Song Chen, Tianpeng Li, Zehuan Li, Lijun Liu, Lingfeng Ming, Guosheng Dong, Da Pan, et al. Baichuan-omni-1.5 technical report. *arXiv preprint arXiv:2501.15368*, 2025b.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Stan Weixian Lei, Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for gui visual agent. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19498–19508. IEEE, 2025.

Chen Liu, Ling Chen, Hanzhang Zhou, Liangyu Chen, Chenglin Cai, Xin Yu, Steven Hoi, and Yue Wang. One forward beats two: Innerzoom for accurate and efficient gui grounding, 2026a. URL https://arxiv.org/abs/2606.30084.

Yichao Liu, Huawen Shen, Liu Yu, Shiyu Liu, Zeyu Chen, and Yu Zhou. Drs-gui: Dynamic region search for training-free gui grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 34606–34616, June 2026b.

Ziwei Liu, Tao Feng, Borui Kang, Yanbing Yang, and Jun Luo. Zoom to essence: Trainless gui grounding by inferring upon interface elements, 2026c. URL https://arxiv.org/abs/2603.14448.

Ziwei Liu, Borui Kang, Hangjie Yuan, Zixiang Zhao, Wei Li, Yifan Zhu, and Tao Feng. Continual GUI agents. In *Forty-third International Conference on Machine Learning*, 2026d. URL https://openreview.net/forum?id=7WeA5TAjHK.

Xudong Lu, Xueying Li, Annan Wang, Yang Bo, Jinpeng Chen, Zengliang Li, Nianzu Yang, Rui Liu, Xue Yang, Jingwen Hou, et al. Omniinteract: Benchmarking real-world streaming interaction for real-time omnimodal assistants. *arXiv preprint arXiv:2605.26485*, 2026.

Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. Omniparser for pure vision based gui agent, 2024. URL https://arxiv.org/abs/2408.00203.

Tiange Luo, Lajanugen Logeswaran, Justin Johnson, and Honglak Lee. Visual test-time scaling for gui agent grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19989–19998, October 2025.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems*, 36:59708–59728, 2023.

Qianhui Wu, Kanzhi Cheng, Rui Yang, Chaoyun Zhang, Jianwei Yang, Huiqiang Jiang, Jian Mu, Baolin Peng, Bo Qiao, Reuben Tan, Si Qin, Lars Liden, Qingwei Lin, Huan Zhang, Tong Zhang, Jianbing Zhang, Dongmei Zhang, and Jianfeng Gao. Gui-actor: Coordinate-free visual grounding for gui agents, 2025a. URL https://arxiv.org/abs/2506.03143.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. Os-atlas: Foundation action model for generalist gui agents. In *International Conference on Learning Representations*, volume 2025, pp. 5090–5108, 2025b.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report, 2025. URL https://arxiv.org/abs/2503.20215.

Jiashu Yao, Heyan Huang, Daiqing Wu, Wangke Chen, Huaxi Ai, Haoyu Wen, Zeming Liu, and Yuhang Guo. Benchmarking living-screen-native gui agents on short-video platforms, 2026. URL https://arxiv.org/abs/2606.04701.

Mengqi Yuan, Zilong Zhou, Xinzhuang Xiong, Weiming Wu, Jiayang Sun, Jiamin Song, Kaiqian Cui, Bowen Wang, Haoyuan Wu, Yitong Li, Dunjie Lu, Haikong Lu, Qi Zhen, Xinyuan Wang, Jiaqi Deng, Yuhao Yang, Cheng Chen, Boyuan Zheng, Alex Su, Xiao Yu, Hao Zou, Saaket Agashe, Xing Han Lu, Manpreet Kaur, Zhengyang Qi, Vincent Sunn Chen, Frederic Sala, Dayiheng Liu, Junyang Lin, Zhou Yu, Yu Su, Siva Reddy, Xin Eric Wang, Peng Qi, Tianbao Xie, and Tao Yu. Osworld 2.0: Benchmarking computer use agents on long-horizon real-world tasks, 2026. URL https://arxiv.org/abs/2606.29537.

Borui Zhang, Bo Zhang, Bo Wang, Wenzhao Zheng, Yuhao Cheng, Liang Tang, Yiqiang Yan, Jie Zhou, and Jiwen Lu. Bami: Training-free bias mitigation in gui grounding, 2026a. URL https://arxiv.org/abs/2605.06664.

Fuyao Zhang, Jiaming Zhang, Che Wang, Xiongtao Sun, Yurong Hao, Guowei Guan, Wenjie Li, Longtao Huang, and Wei Yang Bryan Lim. Dualtap: A dual-task adversarial protector for mobile mllm agents, 2025. URL https://arxiv.org/abs/2511.13248.

Fuyao Zhang, Jiaming Zhang, Che Wang, Boyang Chen, Yurong Hao, Xiongtao Sun, Guowei Guan, Blaise Delattre, Yang Cao, and Wei Yang Bryan Lim. Synchain: Inducing computer-use agent systems to construct their own attack chains, 2026b. URL https://arxiv.org/abs/2608.06862.

APPENDIX

A Notation **14**

B Discussion and Limitations **14**

C More Experimental Results **14**

C.1 Capacity surface. 14

C.2 Interaction depth across backbones. 14

C.3 Prioritization, not capacity. 15

C.4 Paired Error Correction 15

D More Implementation Details **16**

D.1 Prompt and action space. 16

D.2 Training target and inference. 18

D.3 Step template. 18

D.4 Experiment Details 18

D.5 Controlled inputs. 18

E Algorithm **20**

F Case Study **20**

G Theoretical Analysis **22**

A NOTATION

Table 4 lists the main symbols. Subscript t denotes the interaction step and superscript (m) a perceptual channel.

Table 4: Notation.

Symbol	Meaning	Symbol	Meaning
t	Interaction step index	P_m	Channel projector
$\mathcal{O}_t$	Omni-modal observation	$e_t^{(m)} \in \mathbb{R}^d$	Channel embedding
g	User goal	d	Shared channel dim. (128)
I_t	Execution screenshot	$e_t^{(\text{all})}$	Integrated observation state
V_t	Video observation	q_t	Decision reference
A_t	Audio observation	r_m	Channel-identity embedding
H_t	Interaction history	$s_t^{(m)}$	Channel score
S_t	Device state	τ	Softmax temperature (1.0)
y_t	Ground-truth action tokens	$\alpha_t^{(m)}$	Step-specific priority
L_t	Number of action tokens	c_t	Prioritized representation
$\mathcal{F}_{\Theta_0}$	Frozen omni-modal backbone	$z_t \in \mathbb{R}^{d_z}$	Compact focus latent
Θ_0	Frozen backbone parameters	d_z	Action-sufficient dim. (64)
ϕ	Trainable FocusGUI parameters	R_t	Residual, $\mathbb{R}^{K \times d_{\text{model}}}$
E_t	Prompt embeddings	K	Residual slots (4)
$\widetilde{E}_t$	Embeddings after injection	J_t	$\{b_t - K, \ldots, b_t - 1\}$
d_{model}	Decoder hidden size	$\mathcal{S}_{J_t}(\cdot)$	Placement at J_t, else zero
b_t	Action boundary position	γ	Residual scale (1.0)
$\mathcal{M}$	Channel index set	$\mathcal{L}_{\text{act}}$	Action-token NLL, Eq. (4)
$x_t^{(m)}$	Raw channel feature	TM / EM / GP	Evaluation metrics

B DISCUSSION AND LIMITATIONS

FocusGUI is designed as a lightweight mechanism for improving how frozen omni-modal GUI agents organize step-dependent evidence, and it is complementary to advances in perception, grounding, and planning. Our current formulation computes channel priorities from the observation available at each interaction step, which keeps the method simple and model-agnostic; extending this mechanism to incorporate anticipatory signals about which evidence may become important in subsequent steps is a natural direction for future work. Our experiments span four omni-modal backbones and five interaction categories; extending focused perception to broader computer-use settings is a natural next step.

C MORE EXPERIMENTAL RESULTS

C.1 CAPACITY SURFACE.

Figure 6 shows the full grid, with markers at measured configurations and a thin-plate interpolation between them. For Overall TM, EM, and GP the surface is flat over most of its extent: no ridge appears that a joint search over K and d_z would locate, and the default sits inside the flat region rather than on a peak that depends on either parameter. Reading the markers rather than the interpolated surface, the only configuration that separates from the rest is $d_z = 32$, where compression removes information the action needs.

C.2 INTERACTION DEPTH ACROSS BACKBONES.

Figures 7 to 9 repeat the analysis on MiniCPM-o 4.5, VITA-1.5, and Baichuan-Omni-1.5, which span 13.98 to 36.02 Overall TM under frozen inference and differ in architecture and multimodal representation. Each frozen backbone declines as the episode advances, and each FocusGUI counterpart holds its accuracy over the same range, so the margin widens with depth in every case. On

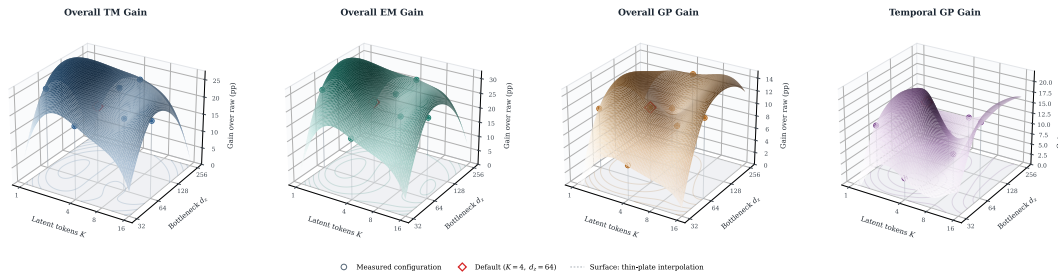

Figure 6: **The capacity grid is flat around the default.** Overall TM, EM, GP, and Temporal GP as a function of the boundary window K and the action-sufficient dimension d_z on Qwen2.5-Omni. Circles mark measured configurations, and the surface is a thin-plate interpolation between them; the diamond marks the default $K = 4$, $d_z = 64$ used throughout the paper. The interpolated regions carry no measurement and are shown only to convey the shape of the grid.

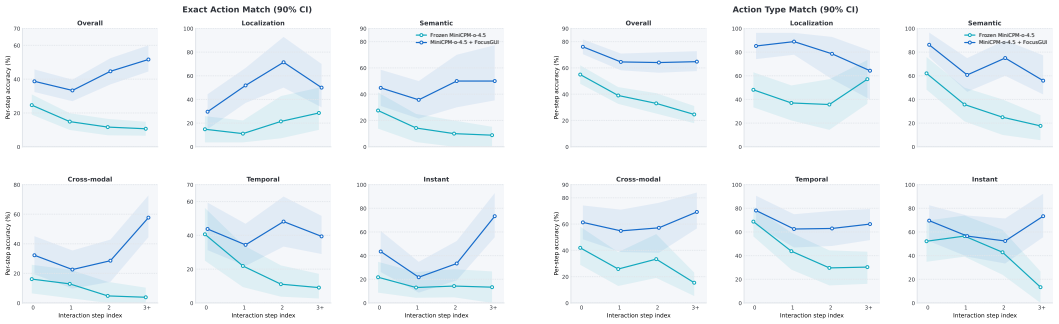

Figure 7: **MiniCPM-o 4.5 across interaction depth.** Per-step Exact Match (left) and Type Match (right) with steps binned by their index within the episode and 3+ collecting the fourth step onward. Shaded regions are 90% episode-clustered bootstrap intervals. As on Qwen2.5-Omni, the frozen backbone declines with depth while FocusGUI holds, so the margin widens rather than staying constant.

exact match the FocusGUI curve is flat or rising at the depths where the frozen curve falls. The decay is therefore a property of the interaction regime rather than of a particular model, and the mechanism that addresses it transfers across backbone families with no change to the architecture or to the training recipe.

C.3 Prioritization, not capacity.

FocusGUI adds trainable parameters to a frozen backbone, so part of its gain could in principle come from that budget alone. Table 5 holds the budget fixed to within 79 parameters and replaces the decision reference with a fixed concatenation layout, leaving the channels available but no longer scored per step. This control differs from the uniform-priority variant in Table 2: uniform priorities average the channel embeddings with equal weights, so channel identity is lost in the pooled representation, whereas concatenation keeps each channel in its own slot and lets the downstream projection learn a fixed, step-independent weighting. It is therefore the stronger static baseline, and isolates per-step scoring as the only difference. Dynamic prioritization improves all three metrics, with the largest gains on Exact Match (+3.54) and Goal Progress (+3.44) and a smaller change in Type Match (+0.98). The gain therefore appears where the mechanism is meant to act, on executable grounding rather than on primitive selection, and it is not explained by latent capacity.

C.4 Paired Error Correction

Figure 10 examines how FocusGUI changes action correctness relative to the backbone across task categories. Corrections consistently outnumber regressions in all five categories, with 153 corrected predictions versus 13 regressions overall. The largest net improvement occurs in temporal reason-

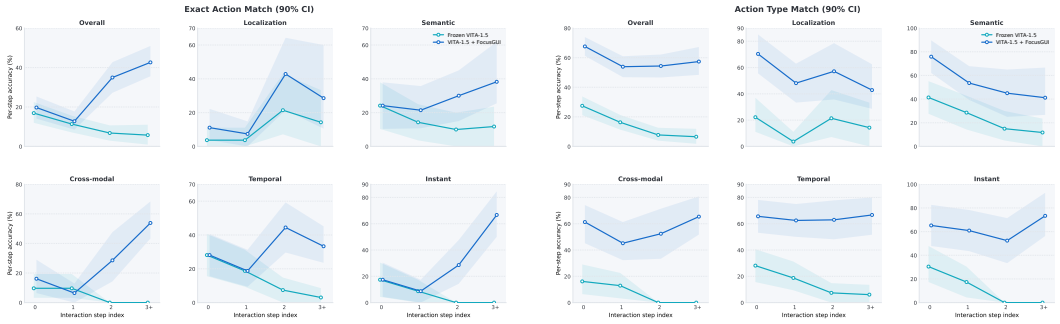

Figure 8: **VITA-1.5 across interaction depth.** Per-step Exact Match (left) and Type Match (right). VITA-1.5 is the weakest of the four backbones at 13.98 Overall TM, and the depth-dependent separation appears at that level as well.

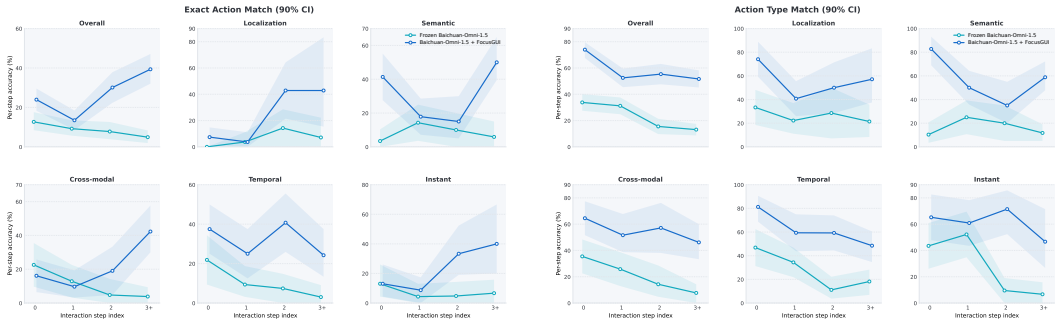

Figure 9: **Baichuan-Omni-1.5 across interaction depth.** Per-step Exact Match (left) and Type Match (right). The same widening margin holds on a third backbone family, indicating that the decay is a property of the interaction regime rather than of a particular model.

ing, indicating that FocusGUI is particularly effective on tasks requiring decisions over evolving observations.

D More Implementation Details

D.1 Prompt and action space.

All four backbones and all five methods share the system prompt below, so the task description, the action vocabulary, and the output format stay fixed while only the organization of the observation differs. The action space follows OmniGUI and contains thirteen primitives: three pointer actions carrying one coordinate, four swipes carrying a start and an end coordinate, text entry carrying a string, four navigation and termination actions without arguments, and `NONE` for steps at which the agent should observe rather than act. Coordinates are normalized to $[0, 1000]$ and emitted as text through the frozen language-model head, which is what lets FocusGUI retain the base model's action vocabulary instead of discretizing the screen into a classification grid. The model returns one JSON object per step, from which Type Match reads the primitive and Exact Match reads the arguments. The residual window $\mathcal{J}_t$ occupies the four embedding positions immediately preceding that object, so the injected latent is available to every generated action token while the prompt itself is left unchanged.

System Prompt — Mobile GUI Agent

Role. You are an Android GUI agent. Given a task, previous action history, and the available screenshots, audio, and video, predict the next executable action required to complete the task.

Table 5: Parameter-matched control on Qwen2.5-Omni. The concatenation variant joins the perceptual channels with a fixed layout instead of scoring them against a decision reference, and is sized to the same trainable budget. Both configurations were trained under identical settings and evaluated on the same 508 test steps.

		Metric (%)		
Configuration	**Params**	TM	EM	GP
Parameter-matched concatenation	1,444,913	71.85	33.46	10.14
Dynamic prioritization	1,444,992	**72.83**	**37.01**	**13.58**
Δ (dyn. $-$ concat.)	+79	+0.98	+3.54	+3.44

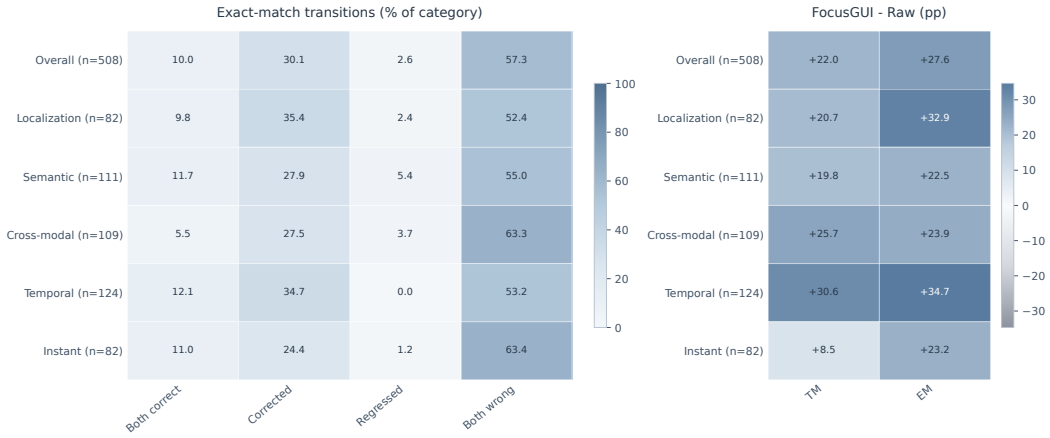

Figure 10: Paired prediction outcomes across task categories. Left: proportions of steps that both models predict correctly, FocusGUI corrects, FocusGUI regresses, or both predict incorrectly. Right: improvements in type match (TM) and exact match (EM), measured in percentage points.

Output. Return *exactly one* valid JSON object without explanations, markdown, or additional text.

Action space.

ID	Action	Arguments
−1	NONE	— observation or waiting
0	TAP	`coordinate: [x, y]`
1	DOUBLE_TAP	`coordinate: [x, y]`
2	LONG_PRESS	`coordinate: [x, y]`
3	SWIPE_UP	`start_coordinate, end_coordinate`
4	SWIPE_DOWN	`start_coordinate, end_coordinate`
5	SWIPE_LEFT	`start_coordinate, end_coordinate`
6	SWIPE_RIGHT	`start_coordinate, end_coordinate`
7	INPUT	`text: string`
8	BACK	—
9	HOME	—
10	TASK_COMPLETE	—
11	TASK_IMPOSSIBLE	—

Conventions.

- Coordinates use $[x, y]$ order, normalized to $[0, 1000]$; the current screenshot is the execution surface.
- Return `NONE` when observation or waiting is required; `TASK_COMPLETE` when the goal has already been achieved; `TASK_IMPOSSIBLE` only when the task cannot be completed.

D.2 Training target and inference.

During training, the ground-truth action is serialized as a compact JSON string and appended only as the assistant target, e.g., {"action_type":0,"coordinate":[421,736]}. Loss is computed exclusively over assistant action tokens; all system and user tokens are masked. During inference, the target is absent and the original frozen LM head autoregressively generates the action using greedy decoding with beam size one.

The raw backbone and FocusGUI receive exactly the same semantic prompt, multimodal observations, action history, action schema, and decoding configuration. FocusGUI does not append textual instructions or modality hints. Its selected evidence is projected into continuous latent residuals at the assistant action boundary before generation by the original decoder and LM head.

D.3 Step template.

Each step is presented with the template below. The multimodal blocks appear in a fixed order, with the history screenshot first when one exists, then the temporal and acoustic observations, and the execution screen last, so the screen the action applies to is always the most recent visual input. The goal and the step index follow, then the actions already executed in this episode, rendered as text rather than as structured state. The observation elements therefore reach the backbone exactly as they would without FocusGUI, and the channels of Sec. 2 are read from this same input rather than replacing any part of it. The closing line ends the prompt and fixes the action boundary b_t at which the residual is applied.

User Prompt — Step Template

Optional history <HistoryScreenshot>
Temporal observation <Video_t>
Acoustic observation <Audio_t>
Execution screen <Screen_t>

Goal. {task_instruction} **Step.** {step_index}

Action history.

Step 0: {previous_action_0}
...
Step $t-1$: {previous_action_t-1}

Analyze the current observation and output the next action as JSON.

D.4 Experiment Details

We construct episode-disjoint training, validation, and test sets containing 1,807, 257, and 508 interaction steps from 495, 71, and 142 trajectories, respectively. All feature transformations and normalization statistics are fitted using the training split only. Validation data are used exclusively for checkpoint selection, and the test split is evaluated once after model selection. At inference time, the original LM head generates one structured action using greedy decoding with beam size one. Raw backbones and their FocusGUI variants use identical observations, prompts, action histories, action spaces, and decoding configurations. On Qwen2.5-Omni, FocusGUI contains 1.445M trainable parameters compared with 8.932B frozen Thinker parameters, corresponding to only 0.0162% trainable parameters. We train a separate FocusGUI adapter for each of the four backbones, with no adapter parameters shared across backbone families. Across the evaluated backbones, the adapter size ranges from 1.15M to 1.45M parameters. All experiments are conducted using two NVIDIA L20 GPUs.

D.5 Controlled inputs.

Table 7 lists what the frozen backbone and FocusGUI receive at inference. The prompt, the observations, the action history, the schema, and the decoding procedure are identical, and neither

Table 6: Default FocusGUI implementation configuration.

Configuration	Value
Shared channel dimension d	128
Action-sufficient dimension d_z	64
Residual slots K	4
Residual initialization	Zero
Residual scale γ	1.0
Channel temperature τ	1.0
Dropout	0.05
Optimizer	AdamW
Learning rate	2×10^{-4}
Weight decay	10^{-3}
Maximum epochs / patience	3 / 2
Gradient clipping	1.0
Decoding	Greedy, beam size 1
Qwen FocusGUI parameters	1.445M
Frozen Qwen Thinker parameters	8.932B
Trainable proportion	0.0162%

Table 7: What the two configurations receive at inference. Every row above the rule is shared, so the columns are indistinguishable until the last two entries. $\mathcal{J}_t$ denotes the K embedding positions preceding the action boundary.

	Frozen backbone	+ FocusGUI
System prompt	identical	
Multimodal observations	identical	
Action history	identical	
Action schema	identical	
Decoding	greedy, beam size 1	
Textual demonstrations	none	
Modality-relevance hints	none	
Backbone weights	frozen	
Trainable parameters	0	1.15M–1.45M
Input at $\mathcal{J}_t$	$E_{t,\mathcal{J}_t}$	$E_{t,\mathcal{J}_t} + \gamma R_t$

configuration is given textual demonstrations or any hint about which modality matters at a step. The two differ in the parameters that build z_t and in the residual added at $\mathcal{J}_t$, and Table 5 separates those by holding the parameter budget fixed while removing the per-step scoring. A gain over the frozen backbone therefore reflects the content of R_t rather than a longer prompt, an in-context example, or a different decoding rule.

Algorithm 1 FocusGUI forward pass at interaction step t

Require: Observation $\mathcal{O}_t$; frozen backbone $\mathcal{F}_{\Theta_0}$; trainable $\phi = \{P_m, P_q, \{r_m\}, W_e, W_q, v, P_s, W_p, b_p\}$; hyperparameters K, d_z, τ, γ

Ensure: Modified embeddings $\widetilde{E}_t$; channel priorities $\{\alpha_t^{(m)}\}$

$E_t,\ \{x_t^{(m)}\}_{m\in\mathcal{M}},\ e_t^{(\text{all})} \leftarrow \mathcal{F}_{\Theta_0}.\textsc{Encode}(\mathcal{O}_t)$ ▷ prompt embeddings, channels, integrated state

for all $m \in \mathcal{M}$ **do** ▷ channel decomposition (CD)

  $e_t^{(m)} \leftarrow P_m\left(x_t^{(m)}\right) \in \mathbb{R}^d$

end for

$q_t \leftarrow P_q\left(e_t^{(\text{all})}\right)$ ▷ decision reference, recomputed every step

for all $m \in \mathcal{M}$ **do** ▷ step-adaptive prioritization (SP)

  $s_t^{(m)} \leftarrow v^\top \tanh\left(W_e\left(e_t^{(m)} + r_m\right) + W_q q_t\right)$

end for

$\alpha_t^{(m)} \leftarrow \dfrac{\exp\left(s_t^{(m)}/\tau\right)}{\sum_{j\in\mathcal{M}} \exp\left(s_t^{(j)}/\tau\right)}, \quad c_t \leftarrow \sum_{m\in\mathcal{M}} \alpha_t^{(m)} e_t^{(m)}$

$z_t \leftarrow P_s(c_t) \in \mathbb{R}^{d_z}$ ▷ action-sufficient projection (AP), $d_z < d$

$R_t \leftarrow \textsc{Reshape}(W_p z_t + b_p) \in \mathbb{R}^{K\times d_{\text{model}}}$ ▷ W_p, b_p zero-initialized $\Rightarrow R_t = 0$ at the start of training

$J_t \leftarrow \{b_t - K, \ldots, b_t - 1\}$ ▷ K prompt positions before the action boundary

$\widetilde{E}_t \leftarrow E_t + \mathcal{S}_{J_t}(\gamma R_t)$ ▷ additive residual alignment (RA); $\mathcal{O}_t$ is neither pruned nor extended

return $\widetilde{E}_t,\ \{\alpha_t^{(m)}\}_{m\in\mathcal{M}}$

E Algorithm

Algorithm 1 details the per-step FocusGUI forward pass, which is shared by training and inference: it exposes the perceptual channels, scores them against the decision reference q_t, compresses the result into z_t, and adds the zero-initialized residual at the K positions preceding the action boundary. Algorithm 2 gives the training and inference procedures built on it. Blue marks everything FocusGUI contributes; all remaining lines are the frozen backbone's original pathway. Gradients flow through the frozen decoder to ϕ, but no backbone parameter is updated and the prompt itself is never modified.

F Case Study

To demonstrate the comprehensiveness and robustness of FocusGUI across diverse interaction paradigms, we present two distinct case studies encompassing both cross-modal temporal reasoning and multi-step semantic localization. Real-world GUI operation requires agents to seamlessly transition between different cognitive modes: some tasks demand patient monitoring of streaming sensory inputs (e.g., waiting for an auditory cue), while others require precise, sequential spatial navigation through dense interfaces.

The following examples illustrate how FocusGUI successfully navigates these contrasting extremes by dynamically prioritizing relevant perceptual channels. Figure 11 highlights a scenario where the decisive signal is entirely auditory and temporal, requiring the agent to ignore static visual saliency until a specific semantic condition is met. Conversely, Figure 12 showcases a highly spatial and semantic task, where the agent must execute a precise sequence of UI interactions (opening filters, selecting specific nested options) based on a composite text instruction. Together, these cases confirm that FocusGUI's step-adaptive prioritization effectively generalizes across fundamentally different types of GUI challenges, preventing Sequential Distraction in both streaming and structurally complex environments.

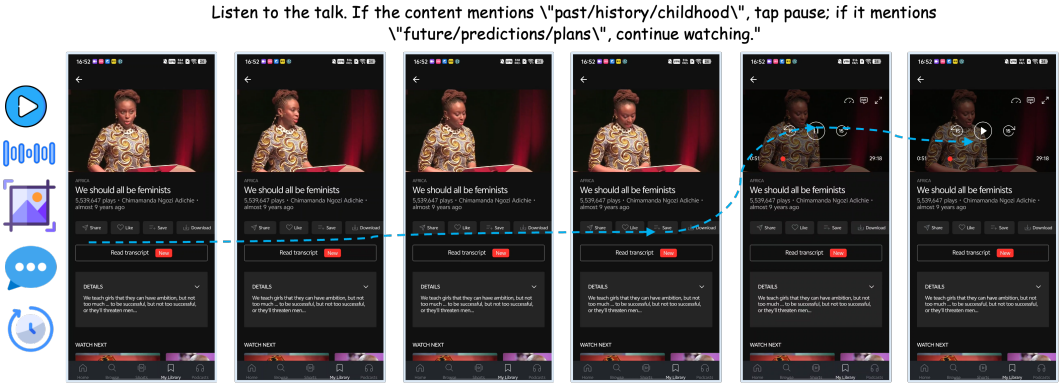

Figure 11: Case study on cross-modal temporal reasoning. The agent is instructed to pause a video based on specific auditory cues (”past/history/childhood”). The visual interface remains largely static, forcing the model to rely on streaming audio comprehension over multiple steps. FocusGUI successfully avoids premature visual actions, dynamically shifting priority to the audio channel and executing the spatial action (tapping pause) precisely when the semantic condition is met.

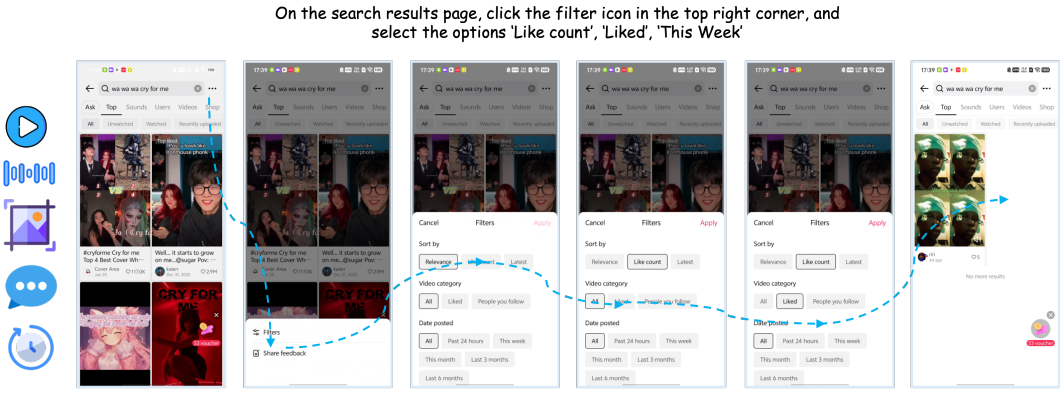

Figure 12: Case study on multi-step semantic planning and spatial localization. The agent executes a complex sequence of interface interactions based on a detailed text instruction. It successfully navigates through dynamic UI changes—locating a small filter icon, understanding the popup menu, and sequentially selecting ”Like count”, ”Liked”, and ”This Week”. This demonstrates FocusGUI’s capacity for precise spatial grounding and robust state tracking in visually dense environments.

Algorithm 2 Training from action supervision, and inference

procedure TRAIN($\mathcal{D}_{\text{tr}}, \mathcal{D}_{\text{val}}$)
  Initialize ϕ; set $W_p \leftarrow 0$, $b_p \leftarrow 0$; freeze encoder, decoder, and LM head (Θ_0)
  for epoch $= 1$ **to** $E_{\max}$ **do**
    for mini-batch $\mathcal{B} \subset \mathcal{D}_{\text{tr}}$ **do**
      for step $t \in \mathcal{B}$ **do**
        $\widetilde{E}_t, \cdot \leftarrow$ FOCUSGUIFORWARD($\mathcal{O}_t$) ▷ Algorithm 1
        $\ell_t \leftarrow -\frac{1}{L_t}\sum_{k=1}^{L_t} \log p_{\Theta_0}\big(y_{t,k} \mid y_{t,<k}, \widetilde{E}_t\big)$ ▷ teacher forcing; loss on action tokens only
      end for
      $\mathcal{L}_{\text{act}}(\phi) \leftarrow \frac{1}{|\mathcal{B}|}\sum_{t\in\mathcal{B}} \ell_t$
      $\phi \leftarrow$ ADAMW$\big(\phi,$ CLIPGRAD$\big(\nabla_\phi \mathcal{L}_{\text{act}}\big)\big)$ ▷ ∇_{Θ_0} discarded; Θ_0 unchanged
    end for
    Evaluate on $\mathcal{D}_{\text{val}}$; keep the best checkpoint; stop early if patience is exhausted
  end for
  return $\phi^\star$
end procedure

procedure INFER(g)
  $H_0 \leftarrow \emptyset$
  for $t = 0, 1, 2, \ldots$ until termination or step budget **do**
    Observe $\mathcal{O}_t = \{g, I_t, V_t, A_t, H_t, S_t\}$
    $\widetilde{E}_t, \{\alpha_t^{(m)}\} \leftarrow$ FOCUSGUIFORWARD($\mathcal{O}_t$)
    $\hat{y}_t \leftarrow \mathcal{F}_{\Theta_0}$.DECODE$\big(\widetilde{E}_t\big)$ ▷ original LM head, greedy decoding, beam size 1
    Execute $\hat{y}_t$; $H_{t+1} \leftarrow H_t \cup \{\hat{y}_t\}$ ▷ the executed action re-enters the next observation
  end for
end procedure

G THEORETICAL ANALYSIS

We trace how the latent z_t reaches the action logits through the frozen decoder, and show that it acts through two pathways with distinct roles: one changes *what* the decoder reads from a position and suffices to select the action primitive; the other changes *where* it reads and is the only route by which arguments can be corrected.

The decoder is L frozen pre-LN blocks on a residual stream $h_i^{(\ell)} \in \mathbb{R}^d$, with per-head attention $q_k = W_Q \mathrm{LN}(h_k)$, $\kappa_i = W_K \mathrm{LN}(h_i)$, $v_i = W_V \mathrm{LN}(h_i)$, $a_{k\cdot} = \mathrm{softmax}(\langle q_k, \kappa_\cdot\rangle/\sqrt{d_h})$, $o_k = W_O \sum_i a_{ki} v_i$, and logits $f_k = W_U \mathrm{LN}(h_k^{(L)})$. FocusGUI sets $h_i^{(0)} = e_i + \gamma R^{(i)} \mathbb{1}[i \in J_t]$ with $R = \mathrm{Reshape}(W_p z_t + b_p)$ and $J_t = \{b_t - K, \ldots, b_t - 1\}$. Write $\delta_i^{(\ell)} := \partial h_i^{(\ell)}/\partial\gamma$ and Λ_i for the LayerNorm Jacobian. Since the finite change $\Delta f_k = \gamma \int_0^1 \partial_\gamma f_k|_{\gamma u}\, du$, the decomposition below holds without linearization. Because $\delta_{J_t}^{(0)}$ ranges over an affine set with direction space $\mathrm{col}(W_p)$ and the frozen Jacobians act linearly, every $\delta_i^{(\ell)}$ lies in a family of dimension at most d_z: the adapter steers the frozen computation through a d_z-dimensional control, and what the network does with it is fixed by Θ_0.

For an action position $k \geq b_t$, differentiating o_k gives

$$\frac{\partial o_k}{\partial \gamma} = W_O \underbrace{\sum_i a_{ki} W_V \Lambda_i \delta_i}_{\text{(V) value}} + W_O \underbrace{\sum_i a_{ki}\, \dot{s}_{ki} \big(v_i - \bar{v}_k\big)}_{\text{(A) attention}}, \qquad \bar{v}_k = \sum_j a_{kj} v_j, \tag{5}$$

where the second form follows from the softmax Jacobian and $\sum_i \partial_\gamma a_{ki} = 0$, and

$$\dot{s}_{ki} = \tfrac{1}{\sqrt{d_h}} \big(\underbrace{\langle \delta_k,\, \Lambda_k^\top W_Q^\top W_K\, \mathrm{LN}(h_i)\rangle}_{\text{query steering}} + \underbrace{\langle q_k,\, W_K \Lambda_i \delta_i\rangle}_{\text{key displacement}} \big). \tag{6}$$

Pathway (A) conserves attention mass and moves the read along the contrasts $v_i - \bar{v}_k$: it adds no content and only reallocates which existing context the action token consumes. Content enters through (V) alone, and only at positions where $\delta \neq 0$.

Proposition G.1 (Depth-staged re-routing)**.** *At layer* 1*,* $\delta_k = 0$ *for all* $k \geq b_t$*, so only key displacement is active and equation 5 reweights attention within* J_t*. From layer* 2 *on, the first layer's attention has written* $\delta_k \neq 0$*, and query steering makes* $\dot{s}_{ki} \neq 0$ *for every unmasked* i*: a residual confined to* K *positions redistributes attention over the entire observation. Moreover, by induction under the causal mask,* $\delta_i^{(\ell)} = 0$ *for all* $i < b_t - K$ *at every layer, so the content* v_i *at every evidence position in the observation is exactly independent of* γ*.*

Consequences for action generation. The two output families of a GUI action consume different pathways. The *primitive* is a categorical choice: flipping it from b to a only requires $W_U \Lambda_k^{(L)} \delta_k^{(L)}$ to have a component along $w_a - w_b$ exceeding the current logit gap, which pathway (V) supplies directly. *Arguments* (coordinates, typed text) are produced by copy-type readout from evidence positions, which satisfy $i < b_t - K$; by Proposition G.1 their content is untouched, so argument corrections arise only through pathway (A) re-routing mass onto the correct evidence. This is why the injection must be *additive*: replacing $e_{J_t} + \gamma R$ by R leaves query steering intact but overwrites the K most recent, most heavily attended context positions with z_t-derived content of dimension at most d_z, destroying their value as evidence.