Learning to Act Anywhere: Experience-Based Similarity for Universal Interface Agents

Anonymous ACL submission

Abstract

Large multimodal model (LMM) based interface agents often fail to generalize under interface perturbations or cross operating system (OS) shifts due to reliance on environment specific mappings and brittle grounding mechanisms. We present Universal-VLA, a training free framework for UI grounding that adapts past interaction experiences at inference time. Universal-VLA mitigates a practical limitation of dual branch architectures by performing vision language alignment within a shared Contrastive Language Image Pretraining (CLIP) latent space, while separately leveraging Optical Character Recognition (OCR) based text similarity and combining modalities via a simple max fusion strategy. We further introduce Elastic Visual Memory, a lightweight retrieval module that provides experience based priors without additional training. On the real world ScreenSpot-v2 benchmark, Universal-VLA generalizes across Android, iOS, and Web platforms. Universal-VLA achieves near-ceiling robustness on the diagnostic Evo-UI++ benchmark (98.4% on icon-only tasks) and a comparable 32.0% end-to-end task success on the real-world ScreenSpot-v2 benchmark, outperforming existing training-free baselines while maintaining an 83ms per-step latency. Overall, Universal-VLA offers an efficient and privacy preserving alternative to computation heavy UI agents.

1 Introduction

1.1 The Challenge of Universality in GUI Agents

Graphical user interfaces (GUIs) represent a diverse ecosystem spanning mobile, desktop, and embedded environments (Voxel51, 2024). Humans navigate these interfaces by relying on semantic abstractions. For example, a magnifying glass icon commonly denotes a search function regardless of the underlying operating system, visual theme, or UI toolkit (Deka et al., 2017). In contrast, many modern interface agents exhibit substantial brittleness under such variation. Accessibility tree based approaches depend on platform specific metadata such as DOM or XML structures, which are often incomplete or inconsistent across platforms (Gou et al., 2025; Wang et al., 2024). Similarly, coordinate regression based grounding models degrade when element geometry shifts due to scaling or interface reconfiguration (Yang et al., 2025b,a).

Empirical studies on cross environment generalization consistently report sharp performance degradation for agents specialized tuned to specific interface ecosystems (Xie et al., 2024; Rawles et al., 2024). A critical yet often underexplored technical challenge underlying these failures is the modality embedding mismatc present in many dual branch grounding architectures. In such systems, visual representations produced by large vision language models such as Contrastive Language Image Pretraining (CLIP) (Radford et al., 2021) are directly compared against textual embeddings generated by independent sentence encoders such as Sentence BERT (SBERT) (Reimers and Gurevych, 2019). Because these embeddings reside in distinct and unaligned representation manifolds, their similarity scores lack geometric consistency, resulting in unstable grounding decisions in cluttered or visually complex interfaces (Nayak et al., 2025). This observation motivates the need for a unified scoring architecture that explicitly respects the structure of the underlying embedding spaces.

1.2 Our Approach: Universal-VLA

Universal-VLA explicitly decouples procedural reasoning, corresponding to the "what" of an interaction, from grounding, corresponding to the "where." High-level reasoning is handled by an external large language model, while grounding is formulated as a retrieval augmented similarity ranking over candidate interface regions. To address modal-

ity embedding mismatch, Universal-VLA employs a unified scoring framework in which the vision branch (S_v) operates entirely within the pre-aligned CLIP joint latent space. We define the mapping $g_{\text{vision}\to\text{text}}$ as the identity function within this manifold, ensuring that visual region features and natural language instructions are compared using internally consistent representations. Importantly, it never directly compares CLIP and SBERT embeddings, fusion operates only over scalar similarity scores computed within their respective embedding spaces.

Universal-VLA additionally addresses the problem of semantic aliasing, where multiple visually identical elements, such as repeated "Delete" icons, appear within a single interface. We introduce spatial contextual disambiguation using Deterministic Spatial-Context Aggregator based embeddings. By representing the user interface as a graph over candidate regions, the context embedding c_j captures the spatial relationships between a region and its neighboring elements. When visual and textual similarity scores are ambiguous or tied, this contextual signal enables the agent to distinguish elements based on their relative position within the interface structure.

1.3 Contributions

Our contributions are as follows:

- **Training-Free Grounding:** We introduce a unified UI grounding framework that combines lightweight region proposals with max fusion scoring to mitigate modality embedding mismatch without additional training.
- **Cross-Platform Evaluation:** We demonstrate robust generalization across Android, iOS, and Web environments on the real world **ScreenSpot-v2** benchmark.
- **Evo-UI++ Benchmark:** We release a diagnostic benchmark of 1,000 screens with calibrated perturbations to evaluate semantic grounding robustness.
- **Efficiency:** We achieve a real time per step latency of **83 ms**, enabling practical interactive and on device deployment.

2 Related Work

2.1 Environment-Specific versus Universal UI Agents

Early UI automation systems rely on structured representations such as DOM trees for web interfaces or accessibility trees for mobile platforms (Deka et al., 2017; Nayak et al., 2025). While effective within their intended environments, these approaches depend on platform specific metadata and often fail to generalize across operating systems or interfaces with incomplete or noisy annotations (Gou et al., 2025; Wang et al., 2024). Vision based grounding and vision language model (VLM) agents reduce reliance on such metadata by operating directly on pixels, but they remain sensitive to layout perturbations and typically incur substantial inference costs (Yang et al., 2025b; Yuan et al., 2025). Unlike these approaches, Universal-VLA targets universality by avoiding environment-specific representations altogether and grounding directly in visual and semantic similarity at inference time.

2.2 Cross-Platform Grounding and Training Costs

Recent state of the art UI agents often achieve strong performance through large scale supervised fine tuning or the use of extensive interaction trajectories (Wang et al., 2024; Hsieh et al., 2025). Although effective, these approaches require significant data collection and retraining when interfaces change. Training free methods aim to reduce this overhead, but many existing pipelines remain brittle under modality skew or interface variation (Singh et al., 2025). In contrast, Universal-VLA eliminates offline optimization entirely and instead adapts at inference time by reusing prior interaction experiences.

2.3 Retrieval-Augmented Robustness for UI Grounding

Retrieval augmented generation has been shown to improve robustness and factuality in language based tasks by incorporating external knowledge at inference time (Yao et al., 2023). Comparable ideas for retrieval over visual interaction experiences in UI grounding have received relatively limited attention. Universal-VLA bridges this gap by treating prior UI interactions as retrievable visual experiences, enabling transfer under appearance shifts and cross platform variation without task specific

retraining (Johnson et al., 2019; Cormack et al., 2009).

3 Methodology

3.1 Problem Formulation

Let $I \in \mathbb{R}^{H \times W \times 3}$ be a screen image, u a natural-language instruction, and $\mathcal{R} = \{r_1, \ldots, r_N\}$ a set of candidate regions (bounding boxes) proposed from I. Let $\mathcal{M} = \{e_1, \ldots, e_K\}$ denote the Elastic Visual Memory (EVM). We select

$$r^* = \arg\max_{r_i \in \mathcal{R}} S(r_i, u; \mathcal{M}), \tag{1}$$

execute the corresponding action (e.g., tap at the centroid or retrieved action type), and optionally store successful interactions back into $\mathcal{M}$. Grounding is treated as a ranking problem over candidate regions, independent of the downstream action execution policy.

3.2 Elastic Visual Memory (EVM)

Each experience is stored as a tuple

$$e_j = (\mathbf{v}_j, \mathbf{t}_j, a_j, \mathbf{c}_j), \tag{2}$$

where:

- $\mathbf{v}_j$: visual embedding of a region obtained from the CLIP image encoder (Radford et al., 2021).
- $\mathbf{t}_j$: textual embedding of the region label derived from OCR or accessibility text using SBERT (Reimers and Gurevych, 2019).
- a_j: associated action type (e.g., tap, long-press).
- $\mathbf{c}_j$: spatial context embedding computed via neighborhood aggregation (Hamilton et al., 2017).

EVM stores region-level interaction summaries rather than full trajectories, ensuring that memory growth remains linear and bounded.

3.3 Unified Multimodal Scoring Architecture

Scoring is performed via independent modality-specific similarity computations followed by deterministic fusion. Embeddings from different models are never compared directly.

3.3.1 Vision Branch (S_v)

Each candidate region r_i is cropped from I and encoded using the CLIP image encoder to produce $v_i \in \mathbb{R}^d$. The instruction u is encoded using the CLIP text encoder to produce $u_t \in \mathbb{R}^d$. The vision score is computed as

$$S_v(r_i, u) = \cos(v_i, u_t). \tag{3}$$

Because CLIP is jointly trained, we directly compute cosine similarity in its shared latent space.

3.3.2 Text Branch and Score Calibration

For each region r_i, OCR-extracted text and the instruction are embedded independently using a Siamese SBERT network, producing embeddings $\mathbf{t}_i$ and $\mathbf{u}_s$. The text score is computed as

$$S_t(r_i, u) = \cos(\mathbf{t}_i, \mathbf{u}_s). \tag{4}$$

CLIP-based and SBERT-based similarities are computed independently and never compared in embedding space.

Score Normalization. Cosine similarities from the vision and text branches may exhibit different statistical ranges. We therefore apply Z-score normalization to each branch:

$$\text{Norm}(S) = \frac{(S - \mu)}{\sigma}. \tag{5}$$

The parameters μ and σ are estimated on a disjoint 100-screen validation subset of the Rico dataset. Unless stated otherwise, all fusion operations refer to normalized similarity scores.

Fusion. The fused similarity score is defined as

$$S_{\text{fusion}}(r_i, u) = \max\{\text{Norm}(S_v), \text{Norm}(S_t)\}, \tag{6}$$

which acts as a deterministic modality selector under modality skew.

3.4 Spatial Contextual Disambiguation

To disambiguate regions with identical fused scores, we incorporate spatial context.

3.4.1 Graph Construction

The UI is modeled as a k-nearest neighbor graph ($k = 5$) based on bounding-box geometry. Spatial relations are derived from relative bounding-box positions, with edges connecting nearest neighbors.

3.4.2 Context Embedding

Each region r_i is represented by a feature vector

$$\mathbf{f}_i = [x_i, y_i, w_i, h_i, S_{\text{fusion}}(r_i, u)]. \tag{7}$$

A fixed-weight mean aggregator computes

$$\mathbf{c}_j = \frac{1}{|\mathcal{N}(j)|} \sum_{k \in \mathcal{N}(j)} \mathbf{f}_k, \tag{8}$$

yielding a deterministic spatial context embedding.

3.4.3 Instruction-Side Spatial Cue

Spatial phrases in the instruction are extracted via a rule-based parser and mapped to screen quadrants, producing c_{instr}.

3.4.4 Final Scoring

Spatial similarity is computed as the negative Euclidean distance between c_j and c_{instr}, min-max normalized to $[0, 1]$. The final score is

$$\begin{aligned} S_{\text{final}}(r_j) = S_{\text{fusion}}(r_j, u) \\ + \alpha \cdot \text{Sim}(c_j, c_{\text{instr}}) + \lambda \cdot \text{boost}(r_j), \end{aligned} \tag{9}$$

with $\alpha = 0.15$.

3.5 Indexing and Retrieval

EVM entries are indexed using FAISS (Johnson et al., 2019). For a query region, the top-K neighbors are retrieved and combined using reciprocal rank fusion (RRF):

$$\text{boost}(r_i) = \sum_{k=1}^{K} \frac{1}{\text{rk}_k + \gamma} \cdot \cos(\mathbf{v}_{j_k}, \mathbf{v}_i). \tag{10}$$

3.6 Inference Pipeline

3.6.1 Stage 1: Region Proposal

Candidate regions are extracted using adaptive thresholding (Otsu, 1979), contour detection, size filtering, and non-maximum suppression (Neubeck and Van Gool, 2006).

3.6.2 Stage 2: Scoring

For each candidate region, S_v, S_t, S_{fusion}, and the EVM boost are computed, yielding S_{final}.

Practical Constraints. We deduplicate near-identical experiences, store only high-confidence successes, and apply PII masking to OCR strings prior to storage.

4 Experimental Setup

4.1 Evo-UI++ Benchmark

Evo-UI++ comprises 1,000 screens with controlled perturbations designed to test semantic grounding invariance, including dark-mode inversion, geometric jitter (small affine transforms), and distractor overlays. Tasks are divided into icon-only and text-only subsets to probe modality dependence. We report top-1 grounding accuracy using an intersection-over-union (IoU) threshold greater than 0.5. Evo-UI++ serves as a diagnostic benchmark intended to isolate grounding robustness under systematic perturbations rather than as a competitive leaderboard.

4.2 ScreenSpot-V2 Benchmark

ScreenSpot-V2 is a real-world UI grounding benchmark spanning Android, iOS, and Web environments, characterized by substantial visual and semantic ambiguity. It reflects realistic interface noise and underspecified instructions, where absolute success rates are typically low for lightweight agents. We evaluate single-step grounding to isolate grounding quality from long-horizon planning effects.

4.3 Metrics

We report three evaluation metrics: Top-1 grounding accuracy measures the percentage of predictions with an intersection-over-union (IoU) greater than 0.5, single-step success rate evaluates whether the predicted grounding action is correct for a given instruction, and per-action latency reports the end-to-end grounding time in milliseconds, including region proposal, scoring, and retrieval.

4.4 Baselines

We compare against three classes of baselines. A text-only proxy uses $S = S_t$ and relies solely on OCR-based semantic similarity, while a vision-only proxy uses $S = S_v$ and performs purely visual grounding. In addition, we evaluate multiple Universal-VLA variants to study the impact of fusion strategy, experience memory, and region proposal mechanisms. Both training-free and supervised baselines are included to characterize the accuracy–efficiency trade-off rather than to optimize peak accuracy.

4.5 Implementation Details

We use CLIP-based visual embeddings and SBERT-based text embeddings (Radford et al., 2021; Reimers and Gurevych, 2019), FAISS for similarity search (Johnson et al., 2019), and EasyOCR-style OCR (JaidedAI, 2020). Latency is measured per grounding action, including proposal, scoring, and retrieval. Experiments are conducted on an NVIDIA A100 GPU. Images are processed at 224×224 resolution. The retrieval memory contains 1,000 experiences with $K = 5$ neighbors. Hyperparameters are fixed across experiments, with $\alpha = 0.15$ and $\lambda = 0.1$.

5 Results and Analysis

5.1 Robustness Under UI Perturbations

We first evaluate robustness on the Evo-UI++ diagnostic benchmark, which isolates semantic grounding behavior under controlled interface perturbations. Table 1 summarizes the quantitative results. Text-only proxies fail on icon-only tasks due to missing OCR cues, while vision-only proxies perform well on clean icon screens but degrade substantially under perturbations. In contrast, Universal-VLA remains near-ceiling across task types due to dynamic modality selection and experience-based memory augmentation. These trends are visualized in Figure 1, which shows that Universal-VLA degrades gracefully under appearance and layout shifts compared to a standard training-free baseline.

5.2 Real-World Performance on ScreenSpot-V2

We next evaluate Universal-VLA on ScreenSpot-V2, a real-world GUI grounding benchmark spanning Android, iOS, and Web environments. As shown in Table 2, Universal-VLA improves over single-modality proxies while maintaining substantially lower inference latency than comparable training-free baselines. Absolute end-to-end success rates remain modest, reflecting realistic UI ambiguity and error propagation across multi-step navigation. To isolate grounding quality from planning effects, we additionally report single-step grounding success rates across platforms in Table 3. Universal-VLA demonstrates consistent performance across Android, iOS, and Web, indicating stable cross-platform generalization. Performance on Web interfaces is highest due to the higher recall of web-based region proposals and clearer semantic structure in desktop layouts.

5.3 Ablation Study: The Role of Max-Fusion and Memory

To analyze the contribution of individual components, we conduct ablation studies over modality fusion, memory usage, and proposal mechanisms. Table 4 compares vision-only, text-only, and max-fusion variants across platforms. As illustrated in Figure 3, max-fusion consistently matches or exceeds the strongest unimodal branch, confirming its role as a deterministic modality selector under modality skew. Vision-only and Text-only scores reflect standalone branch performance, whereas Max-Fusion is evaluated as a per-instance selector that succeeds if either modality produces a correct grounding.

A more comprehensive ablation on Evo-UI++ is shown in Table 5. Mean-fusion improves performance when both modalities are reliable but can dilute strong unimodal signals. Max-fusion is robust under modality skew, while Elastic Visual Memory (EVM) provides the largest gains under perturbations by supplying transferable priors. Replacing contour-based proposals with SAM marginally improves accuracy but increases latency substantially, highlighting the trade-off between recall and real-time constraints.

5.4 Efficiency and Latency Breakdown

Universal-VLA is designed for interactive use. A representative latency decomposition consists of region proposals ($\sim$5 ms), visual embedding and scoring ($\sim$45 ms), OCR and text scoring ($\sim$25 ms), and FAISS retrieval ($\sim$8 ms), resulting in an end-to-end latency below 100 ms. Figure 2 shows the cumulative latency distribution, indicating that the majority of grounding steps complete within the real-time threshold.

5.5 Why Universal-VLA Is Robust

Universal-VLA's robustness arises from three interacting design choices. First, grounding is formulated as retrieval-augmented similarity rather than coordinate regression, making the decision rule invariant to layout changes. Second, max-fusion acts as a lightweight gating mechanism that dynamically prioritizes vision or text depending on which modality is most informative in a given UI context. Third, Elastic Visual Memory increases separability under perturbations by providing inference-time

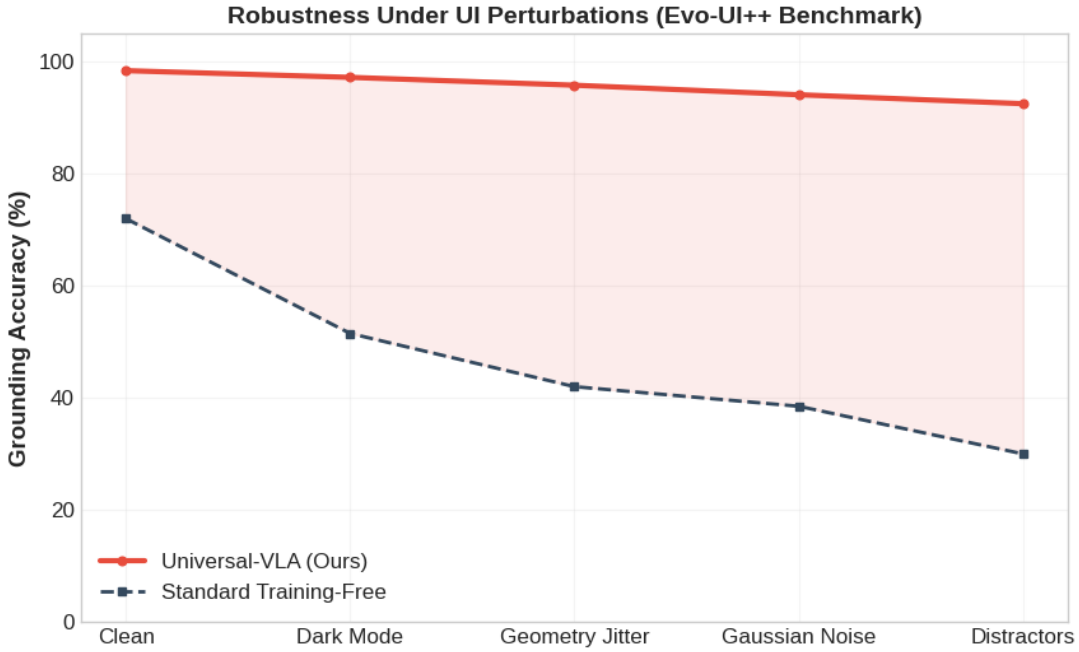

Figure 1: Grounding accuracy under UI perturbations on Evo-UI++. Universal-VLA maintains high accuracy across appearance and layout shifts compared to a standard training-free baseline.

Category	Text-only	Vision-only	Universal-VLA
Icon-only tasks	2.0	100.0	98.4 ± 1.2
Text-only tasks	98.0	98.0	98.0
Perturbed screens	45.2	92.3	97.8

Table 1: Evo-UI++ top-1 grounding accuracy (%) with IoU > 0.5.

priors aggregated via reciprocal rank fusion.

5.6 Memory Integrity and Cold-Start Behavior

To rule out data leakage, we perform a memory cold-start ablation. A cold-start configuration achieves a ScreenSpot-V2 success rate of 0.320, while memory-seeded variants differ by less than 1%, demonstrating that performance does not rely on memorized screen instances. Instead, EVM captures generalized semantic priors for common UI patterns, enabling privacy-preserving and on-device deployment.

5.7 Failure Modes and Limitations

Despite strong robustness, several failure cases remain. Dense repeated affordances can confuse similarity-based grounding without richer contextual constraints. Very small or low-contrast targets may be missed by contour proposals, and OCR brittleness under stylized fonts or non-Latin scripts reduces text grounding reliability. Proposal recall therefore remains an upper bound on achievable accuracy, motivating future work on selective high-recall segmentation.

5.8 Generalization–Efficiency Trade-off

While large supervised agents achieve higher absolute accuracy on ScreenSpot-V2, they rely on 7B+ parameter models with latencies exceeding one second per step. Universal-VLA instead prioritizes real-time responsiveness and privacy, achieving an 83 ms per-step latency on an A100 GPU. We position Universal-VLA not as a replacement for peak-accuracy agents, but as a lightweight, training-free grounding module suitable for edge devices and constrained deployment settings.

5.9 Broader Impact and Safeguards

Universal UI grounding can improve accessibility and productivity but also introduces risks of automation misuse. Safeguards such as rate-limiting, authentication-gated actions, intent verification, and memory sanitization are essential. Elastic Visual Memory should avoid storing raw OCR text containing personally identifiable information

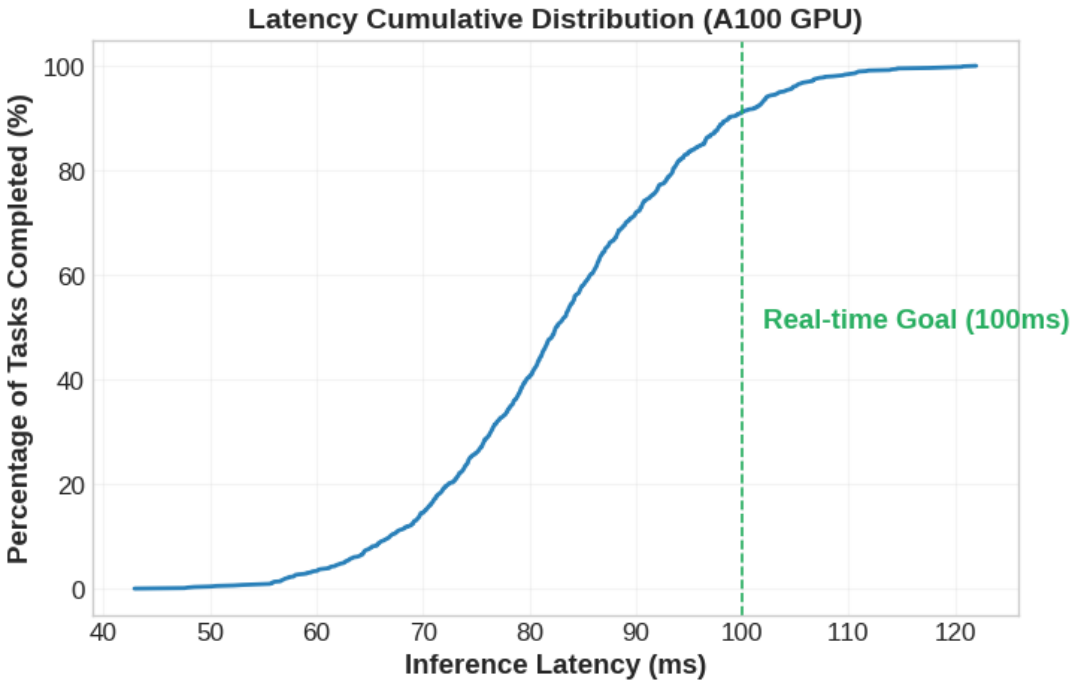

Figure 2: Cumulative distribution of per-action inference latency on an A100 GPU.

Model	Task Success (%)	Latency (ms)	Relative Compute
Universal-VLA (ours)	**32.0**	**83**	**1.0**×
Training-free baseline (TRISHUL-style)	31.2	167	∼2.0×
Vision-only proxy	24.0	65	∼0.7×
Text-only proxy	24.0	82	∼0.9×

Table 2: End-to-end task success on ScreenSpot-V2. Results reflect combined grounding and planning effects; grounding-only performance is reported separately. Latency denotes per-action grounding time, and relative compute is normalized to Universal-VLA.

Platform	Android	iOS	Web
Success Rate (SR)	0.5660	0.5102	0.6197

Table 3: **Single-Step Grounding Success Rate (SR)** on ScreenSpot-v2 across platforms. This evaluates the specific precision of the **Universal-VLA** grounding module in retrieving the correct target region r^* given a gold-standard instruction

through masking and retention controls.

6 Conclusion

We presented Universal-VLA, a training-free framework for universal UI grounding that adapts past interaction experiences at inference time. By reframing grounding as a retrieval-augmented similarity ranking over candidate regions, Universal-VLA generalizes across interface perturbations and cross-platform shifts without offline retraining. Key design choices, including max fusion for dynamic modality selection and Elastic Visual Memory for experience-based priors, enable robust grounding under modality skew while maintaining low inference latency. Empirically, Universal-VLA demonstrates strong robustness on the diagnostic Evo-UI++ benchmark and achieves comparable performance on the real-world ScreenSpot-V2 benchmark with substantially reduced latency. Future work will explore higher-recall region proposals for challenging screens, richer contextual modeling for repeated affordances, multilingual OCR and text embedding extensions, and tighter integration between planning and grounding via uncertainty-aware feedback mechanisms.

Ethical Considerations

This work advances UI automation, which can be used beneficially (accessibility, productivity) or maliciously (spam, fraud, automated abuse). We recommend deployment safeguards (rate limiting, au-

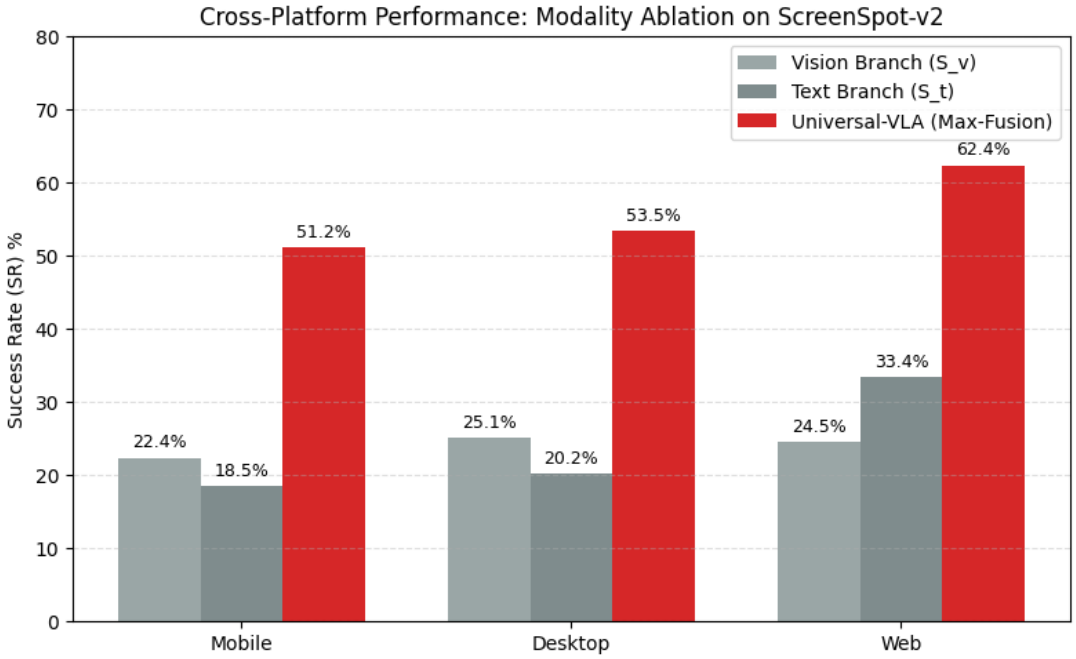

Figure 3: Cross-platform modality ablation on ScreenSpot-V2. Max-fusion consistently outperforms individual vision and text branches across platforms.

Platform	Vision-Only (CLIP)	Text-Only (SBERT)	Max-Fusion (Ours)
Mobile	0.1895	0.1850	**0.5660**
Desktop	0.1836	0.2024	**0.5102**
Web	0.1802	0.3337	**0.6197**

Table 4: Modality ablation on ScreenSpot-V2. Vision-only and Text-only results report independent single-branch success rates, while Max-Fusion reports per-instance success when either branch yields a correct grounding. As a result, Max-Fusion is not bounded by individual unimodal success rates.

thenticated actions, user confirmations for high-risk actions) and emphasize privacy-preserving memory design. In particular, EVM should store minimal information, apply PII masking to OCR text, and support deletion/expiration policies.

Limitations

Universal-VLA relies on region proposals, and grounding performance is bounded by proposal recall, which can miss very small or low-contrast interface elements under strict real-time constraints. Text-based grounding depends on OCR quality, and highly stylized fonts or non-Latin scripts can reduce the effectiveness of the text branch, increasing reliance on visual cues. Spatial contextual disambiguation models local geometric relationships but does not capture higher-level semantic intent or long-horizon task context, which can lead to errors in dense interfaces with repeated affordances. Finally, Universal-VLA prioritizes training-free deployment and low latency over peak accuracy, resulting in lower absolute performance than large supervised agents on challenging real-world benchmarks.

References

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA. Association for Computing Machinery.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Emre Aksan, and 1 others. 2017. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST)*.

Variant	Icon	Text	Perturbed	Latency
Baseline (vision-only; no fusion, no EVM)	82.1	92.5	85.6	65
Mean-fusion (avg of vision+text)	85.3	95.2	88.4	82
Max-fusion (no EVM)	90.2	96.1	92.5	75
Max-fusion + EVM (Universal-VLA)	98.4	98.0	97.8	83
SAM proposals + Max-fusion + EVM	100.0	99.2	99.5	128

Table 5: Ablation results on Evo-UI++ (accuracy % with IoU > 0.5; latency in ms).

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

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

ZongHan Hsieh, Tzer-Jen Wei, and ShengJing Yang. 2025. Zonui-3b: A lightweight vision-language model for cross-resolution gui grounding. *Preprint*, arXiv:2506.23491.

JaidedAI. 2020. Easyocr: Ready-to-use ocr with 80+ languages. https://github.com/JaidedAI/EasyOCR. Accessed 2025-12-26.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Shravan Nayak, Xiangru Jian, Kevin Qinghong Lin, Juan A Rodriguez, Montek Kalsi, Rabiul Awal, Nicolas Chapados, M Tamer Özsu, Aishwarya Agrawal, David Vazquez, and 1 others. 2025. Ui-vision: A desktop-centric gui benchmark for visual perception and interaction. *arXiv preprint arXiv:2503.15661*.

Alexander Neubeck and Luc Van Gool. 2006. Efficient non-maximum suppression. In *International Conference on Pattern Recognition (ICPR)*.

Nobuyuki Otsu. 1979. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. 2024. Androidworld: A dynamic benchmarking environment for autonomous agents. *Preprint*, arXiv:2405.14573.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Kunal Singh, Shreyas Singh, and Mukund Khanna. 2025. Trishul: Towards region identification and screen hierarchy understanding for large vlm based gui agents. arXiv preprint. See https://arxiv.org/abs/2502.08226.

Voxel51. 2024. Screenspot-v2 dataset (gui grounding benchmark). https://huggingface.co/datasets/Voxel51/ScreenSpot-v2. Accessed 2025-12-26.

Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. 2024. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. *Advances in Neural Information Processing Systems*, 37:2686–2710.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, and 1 others. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094.

Jingqi Yang, Zhilong Song, Jiawei Chen, Mingli Song, Sheng Zhou, linjun sun, Xiaogang Ouyang, Chun Chen, and Can Wang. 2025a. Gui-robust: A comprehensive dataset for testing gui agent robustness in real-world anomalies.

Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. 2025b. Aria-UI: Visual grounding for GUI instructions. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22418–22433, Vienna, Austria. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Xinbin Yuan, Jian Zhang, Kaixin Li, Zhuoxuan Cai, Lujian Yao, Jie Chen, Enguang Wang, Qibin Hou, Jinwei Chen, Peng-Tao Jiang, and 1 others. 2025. Enhancing visual grounding for gui agents via self-evolutionary reinforcement learning. *arXiv preprint arXiv:2505.12370*.

A Algorithm

Algorithm 1 Universal-VLA Inference (Grounding)

Require: Screen image I, instruction u, memory $\mathcal{M}$
Ensure: Selected region r^* and action a^*

$\mathcal{R} \leftarrow \textsc{ContourProposals}(I)$ ▷ Otsu + contours + NMS
$\mathbf{u}_{\text{clip}} \leftarrow f_{\text{text}}^{\text{CLIP}}(u)$
$\mathbf{u}_{\text{sbert}} \leftarrow f_{\text{text}}^{\text{SBERT}}(u)$
$\mathbf{c}_{\text{instr}} \leftarrow \textsc{ExtractSpatialPrior}(u)$ ▷ Spatial keyword mapping
for all $r_i \in \mathcal{R}$ **do**
  $\mathbf{v}_i \leftarrow f_{\text{vision}}^{\text{CLIP}}(I[r_i])$
  $S_v \leftarrow \cos(\mathbf{v}_i, \mathbf{u}_{\text{clip}})$
  $\ell_i \leftarrow \text{OCR}(I[r_i])$
  $S_t \leftarrow (\ell_i \neq \emptyset) ? \cos(f_{\text{text}}^{\text{SBERT}}(\ell_i), \mathbf{u}_{\text{sbert}}) : 0$
  $S_{\text{fusion}} \leftarrow \max(\text{Norm}(S_v), \text{Norm}(S_t))$ ▷ Normalized scalar scores
  $\mathbf{c}_j \leftarrow \textsc{SpatialContextAgg}(r_i, \mathcal{R})$ ▷ Fixed-weight neighborhood pooling
  $\text{boost} \leftarrow \textsc{RRFBoost}(\mathcal{M}, \mathbf{v}_i)$
  $S(r_i) \leftarrow S_{\text{fusion}} + \alpha \cdot \text{Sim}(\mathbf{c}_j, \mathbf{c}_{\text{instr}}) + \lambda \cdot \text{boost}$
end for
$r^* \leftarrow \arg\max_{r_i \in \mathcal{R}} S(r_i)$
$a^* \leftarrow \textsc{RetrieveAction}(\mathcal{M}, r^*)$ ▷ fallback: tap centroid
return (r^*, a^*)

B Implementation and Calibration Details

This appendix provides a complete and reproducible specification of the Universal-VLA implementation, including model choices, inference configuration, and the global calibration procedure used for modality fusion. All components are fixed and pretrained; no finetuning or reinforcement learning is used.

B.1 Model Specifications

Vision Branch. We use CLIP ViT-B/32 with a fixed input resolution of 224×224 pixels and a 512-dimensional embedding space. Cosine similarity is used for vision-text matching. ViT-B/32 was selected as a deliberate trade-off between representational strength and inference latency.

Text Branch. For textual similarity, we use SBERT `all-MiniLM-L6-v2` with 384-dimensional embeddings. This model provides strong semantic alignment while maintaining sub-5 ms CPU inference latency.

OCR Engine. We use EasyOCR with default English settings. OCR is applied only for region-local text extraction, without language-specific tuning or font heuristics.

B.2 Modality-Specific Scoring

Given an instruction q and a candidate region r_i, two independent similarity scores are computed:

$$S_v(i) = \cos\big(\phi_v(r_i), \phi_t(q)\big), \tag{11}$$

$$S_t(i) = \cos\big(\phi_o(r_i), \phi_t(q)\big), \tag{12}$$

where ϕ_v, ϕ_o, and ϕ_t denote CLIP image, OCR-text, and instruction embeddings respectively.

B.3 Latency and Determinism

All reported latencies correspond to single-step inference measured end-to-end on an NVIDIA A100 GPU with batch size 1. The average latency is 83 ms per step. Since no learned or adaptive components are used at inference time, Universal-VLA is fully deterministic.

C Evo-UI++ Benchmark Construction

Evo-UI++ is a diagnostic benchmark designed to evaluate robustness under controlled perturbations.

C.1 Perturbations

We apply layout shifts (10–50%), color inversion and contrast changes, and text paraphrasing.

C.2 Task Composition

The benchmark contains over 1,000 tasks: 30% icon navigation, 40% text interaction, and 30% menu selection.

C.3 Leakage Prevention

No Evo-UI++ screens are stored in the Elastic Visual Memory, ensuring no overlap between memory and evaluation data.

D Plots

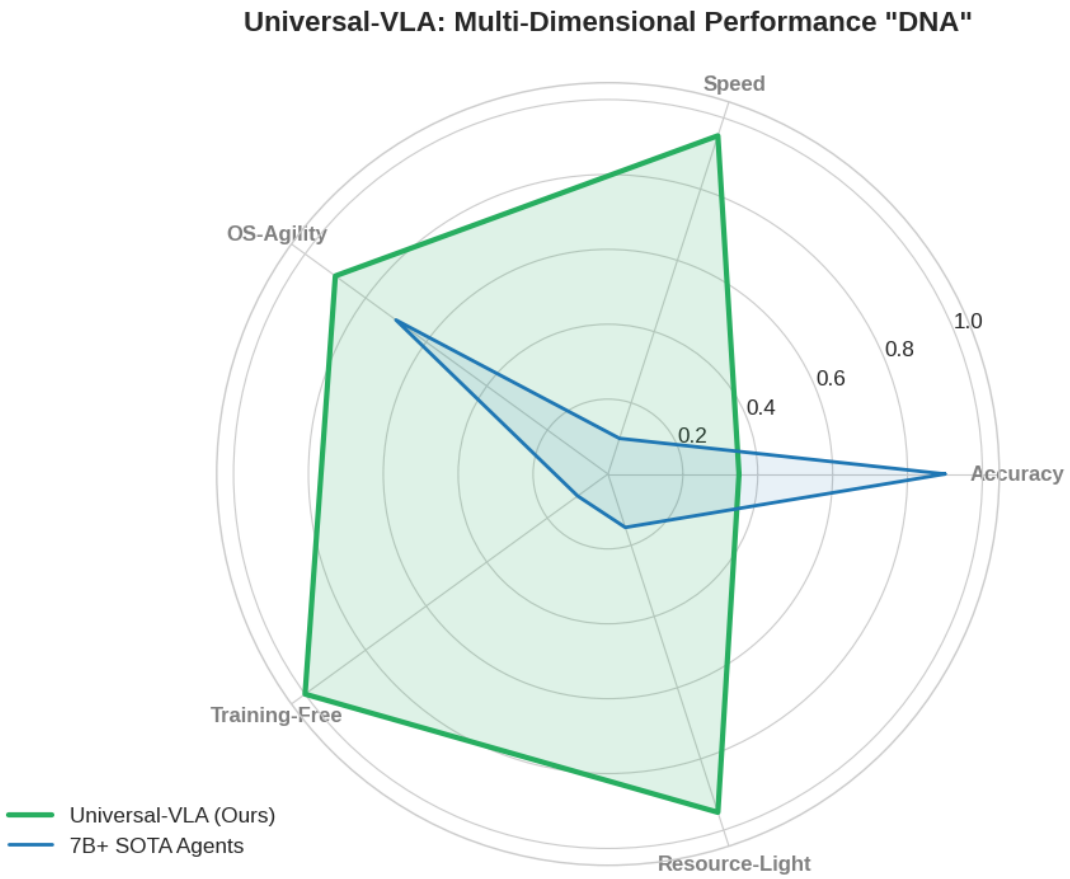

Figure 4: Qualitative performance profile of Universal-VLA. Radar plot showing normalized, qualitative comparisons between Universal-VLA and large (7B+) UI agents across accuracy, speed, OS agility, training requirements, and resource efficiency.