Universal-VLA: Model-Agnostic Closed-Form Scoring for Training-Free UI Grounding

Anonymous ACL submission

Abstract

GUI grounding, the task of identifying the interface element referenced by a natural language instruction, is a core capability for interface agents. Most existing approaches treat grounding as a supervised prediction problem that relies on large UI datasets or models fine tuned on interface data. We revisit this assumption and show that grounding can instead be formulated as a closed form similarity ranking problem over candidate interface regions. We present Universal VLA, a training free framework that selects UI elements using a deterministic scoring rule combining vision similarity, text similarity, Z score calibration, and a max fusion gate. Experiments show that the same scoring equations achieve 45.8% accuracy on ScreenSpot v2 using CLIP ViT B/32, 49.6% with MetaCLIP ViT H/14, and 55.4% with SigLIP SO400M, demonstrating that performance scales with representation quality rather than UI specific supervision. A complementarity analysis with OS Atlas 7B shows that Universal VLA resolves 72 instances the supervised model misses and reaches an oracle accuracy of 89.5%. These results suggest that closed form similarity scoring provides a complementary paradigm for GUI grounding that enables efficient training free element localization.

1 Introduction

Graphical user interface (GUI) grounding, the task of identifying the interface element referenced by a natural language instruction, is a fundamental capability for interactive agents. Reliable grounding enables agents to execute commands, automate workflows, and interact with complex digital environments across mobile applications, desktop software, and web interfaces. As autonomous agents increasingly operate directly within graphical environments, the ability to map language instructions to interface elements becomes essential for practical deployment. Recent work on agent benchmarks highlights the importance of robust grounding for complex computer-use tasks (Xie et al., 2024).

Existing research typically approaches GUI grounding as a supervised learning problem. Large multimodal models trained on annotated interface datasets have demonstrated strong performance on grounding benchmarks. OS-Atlas, for example, learns a grounding model from millions of labeled UI elements (Gou et al., 2025). Other work focuses on smaller architectures trained on curated interface datasets to improve cross-resolution grounding accuracy (Hsieh et al., 2025). Reinforcement learning methods have also been explored to improve grounding performance through interaction feedback (Yuan et al., 2025). These approaches demonstrate the effectiveness of large-scale supervision, but they depend heavily on specialized UI datasets and model finetuning.

The reliance on UI-specific training raises an important question. Modern vision-language encoders already provide strong alignment between images and natural language descriptions through large-scale multimodal pretraining (Radford et al., 2021). At the same time, sentence embedding models provide robust semantic representations for textual instructions (Reimers and Gurevych, 2019). If these representations already capture the semantics required for grounding, it may not be necessary to train models directly on UI datasets. Instead, grounding could potentially be formulated as a ranking problem over candidate interface regions.

In this work we investigate this alternative formulation and propose Universal-VLA, a training-free framework that expresses GUI grounding as closed-form similarity scoring. Given a screenshot and an instruction, Universal-VLA first generates candidate interface regions using classical computer vision techniques. Candidate extraction relies on thresholding and contour detection methods commonly used for visual segmentation (Otsu, 1979). Overlapping regions are filtered using non-

maximum suppression to produce a compact proposal set (Neubeck and Van Gool, 2006). Each candidate region is then evaluated using two independent similarity functions computed from pretrained encoders. The first measures visual similarity between the instruction and the region image, while the second measures semantic similarity between the instruction and OCR-extracted text. Because these similarity scores originate from different embedding spaces, we apply Z-score normalization before combining them through a max-fusion gate that selects the most confident modality.

A central hypothesis of this work is that if GUI grounding can be expressed as a similarity ranking problem, then the performance of the algorithm should scale with the quality of the underlying representations. To test this hypothesis we evaluate Universal-VLA using multiple encoder configurations spanning different parameter scales. Without changing the algorithm, performance improves from 45.8% using CLIP ViT-B/32 to 49.6% using MetaCLIP ViT-H/14. Using the SigLIP encoder further increases accuracy to 55.4%. These results suggest that grounding performance depends primarily on representation quality rather than UI-specific supervision. Interestingly, SigLIP achieves higher accuracy than MetaCLIP despite having fewer parameters. This observation indicates that sigmoid contrastive objectives may be better aligned with independent similarity scoring than softmax-based contrastive training (Zhai et al., 2023).

Beyond empirical competitiveness, we find that similarity-based grounding captures signals that differ from those learned by supervised models. In a complementarity analysis against OS-Atlas, Universal-VLA correctly resolves 72 instances that the supervised model fails. When the predictions of both methods are combined, the oracle accuracy reaches 89.5%. This result suggests that closed-form similarity scoring provides a distinct grounding signal rather than simply approximating coordinate regression approaches. Because the method relies entirely on pretrained representations and deterministic scoring, it remains computationally efficient and can be combined with existing agent architectures.

We evaluate Universal-VLA on three GUI grounding benchmarks. ScreenSpot-v2 measures general grounding performance across diverse interface layouts (Voxel51, 2024). ScreenSpot-Pro evaluates grounding accuracy in high-resolution professional software environments . Evo-UI++ (Appendix C), our diagnostic benchmark constructed from Rico (Deka et al., 2017), evaluates robustness under controlled interface perturbations and layout changes. Across these settings, Universal-VLA surpasses several general-purpose vision-language models despite using no UI-specific training data.

This work investigates three research questions:

- **RQ1:** Can GUI grounding be formulated as a closed-form similarity ranking problem that operates without UI-specific training data?
- **RQ2:** If grounding is a similarity-based ranking task, does performance scale with the quality of pretrained representations rather than changes to the grounding algorithm itself?
- **RQ3:** Does similarity-based grounding capture signals that are complementary to supervised UI models, enabling improved grounding coverage when combined with existing approaches?

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 . While these representations simplify element identification within a single environment, they depend on platform-specific metadata that is often incomplete or inconsistent, which limits generalization across operating systems and interfaces with noisy annotations (Nayak et al., 2025). Recent approaches address these limitations using vision-based grounding methods that operate directly on screenshots (Gou et al., 2025). Vision-language models reduce reliance on metadata but remain sensitive to layout perturbations and typically incur higher inference costs (Yang et al., 2025). General-purpose models such as Qwen2-VL further improve multimodal understanding through large-scale pretraining (Wang et al., 2024b), but they rely on implicit reasoning over entire images rather than explicit region-level selection. Universal-VLA instead formulates grounding as similarity scoring over candidate regions, avoiding environment-specific representations and enabling grounding directly from pretrained representations without UI-specific training.

2.2 Cross-Platform Grounding and Training Costs

Recent UI agents achieve strong grounding performance through large-scale supervised training. Mobile-Agent demonstrates how multimodal models can integrate perception and reasoning for complex interface interaction (Wang et al., 2024a), while lightweight architectures such as ZonUI specialize in GUI grounding using curated datasets (Hsieh et al., 2025). Although effective, these approaches depend on extensive annotation and require retraining as interfaces evolve. Training-free methods aim to reduce this overhead. ManiCoG improves grounding without additional training by using multi-step reasoning at inference time (Zhang et al., 2025), but this introduces computational overhead and reliance on large language models. Other pipelines remain sensitive to modality imbalance and interface variation, limiting robustness in practice (Singh et al., 2025). In contrast, Universal-VLA eliminates offline training and avoids iterative reasoning by performing grounding through deterministic similarity scoring with pretrained encoders.

2.3 Retrieval-Augmented Robustness for UI Grounding

Retrieval-based reasoning has been widely used in language modeling to improve robustness and factual consistency by incorporating external knowledge at inference time (Yao et al., 2023). Similar ideas can be applied to visual interaction tasks, where past interface observations can be stored and retrieved to assist grounding under layout and appearance variation. Efficient similarity search enables scalable retrieval over large embedding collections (Johnson et al., 2019), while ranking fusion methods combine multiple signals to improve robustness (Cormack et al., 2009). Universal-VLA adopts this perspective by treating prior interactions as retrievable visual experiences, enabling grounding decisions to leverage previously observed interface patterns without task-specific retraining.

3 Methodology

We present Universal-VLA, a training-free framework that formulates GUI grounding as a closed-form similarity ranking problem over candidate interface regions. Given a screenshot and a natural language instruction, the objective is to select the region that best matches the user intent. The method defines a four-stage scoring procedure consisting of region proposal, dual-branch similarity evaluation, score normalization, and fusion-based selection. Figure 1 illustrates the overall formulation.

3.1 Problem Formulation

Let I denote a screenshot and u denote a natural language instruction. We define a set of candidate regions $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$ extracted from the image. The grounding task is to select the optimal region

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

$$\mathcal{R} = \mathrm{NMS}\left(\bigcup_s \mathcal{R}_s,\ \tau_{\mathrm{IoU}}\right) \tag{2}$$

where $S(r_i, u)$ is a similarity score between region r_i and instruction u. Universal-VLA defines $S(\cdot)$ as a closed-form function composed of independently computed similarity signals. This formulation reduces GUI grounding to a ranking problem over candidate regions defined entirely by a fixed scoring function, independent of model architecture or training.

3.2 Region Proposal

We generate candidate regions using classical computer vision techniques to ensure that region extraction remains independent of training data and model design. Specifically, we combine contour detection, edge-based segmentation, and Maximally Stable Extremal Regions (MSER) to obtain a diverse set of candidate interface regions.

To incorporate textual information, we extract OCR tokens from each region, producing both visual crops and associated text. The resulting proposal set $\mathcal{R}$ is filtered using non-maximum suppression to remove redundant overlap and maintain a compact candidate set.

Empirically, this proposal engine achieves 98.5% recall across ScreenSpot-v2, meaning the correct target element is present among the candidates for the vast majority of test samples. The remaining grounding challenge is therefore scoring—selecting the correct candidate from approximately 400 options—which is entirely driven by Eq. 3–6

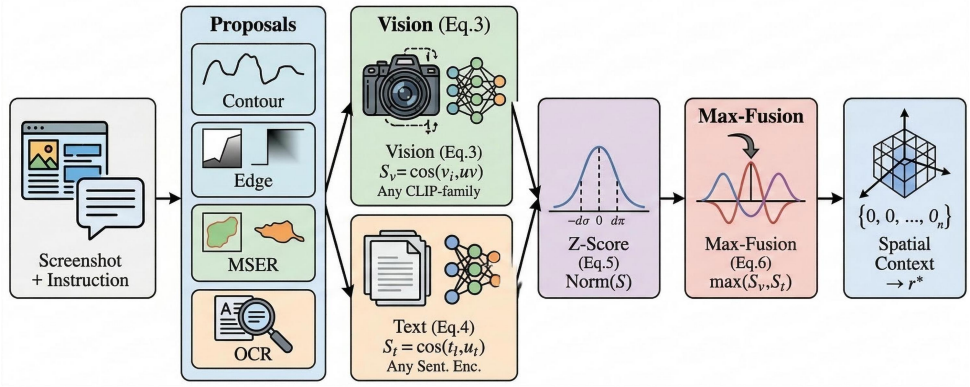

Figure 1: Universal-VLA scoring pipeline. Candidate regions are proposed via four classical CV strategies (contour detection, edge detection, MSER, and OCR). Each candidate is scored independently by a vision branch (Eq. 3, any CLIP-family encoder) and a text branch (Eq. 4, any sentence encoder), Z-score normalized (Eq. 5) to calibrate cross-branch variance ($\sigma_t/\sigma_v \approx 8\times$), and combined via a max-fusion gate (Eq. 6) that routes each instance to its stronger modality. Spatial context disambiguation resolves ties among visually similar candidates. The architecture is **model-agnostic**: swapping CLIP ViT-B/32 for SigLIP-SO400M yields +9.6% with no change to the fusion logic or hyperparameters.

3.3 Dual-Branch Similarity Scoring

For each candidate region r_i, we compute two independent similarity scores that capture visual and textual alignment with the instruction.

Vision similarity. We encode the image crop of region r_i using a pretrained vision-language encoder to obtain an embedding v_i. The instruction u is mapped to a corresponding visual embedding u_v. The vision similarity score is defined as

$$S_v(r_i, u) = \cos(\mathbf{v}_i, \mathbf{u}_v) \tag{3}$$

Text similarity. We extract OCR text from region r_i and encode it using a pretrained sentence encoder to obtain a textual embedding t_i. The instruction is encoded into a textual embedding u_t. The text similarity score is

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

These two scores capture complementary signals, as visual-semantic alignment and explicit textual matching exhibit distinct failure modes in GUI grounding.

3.4 Score Normalization

The vision and text similarity scores are computed in different embedding spaces and exhibit distinct distributions. In practice, text similarity scores often have higher variance than vision similarity scores, which can lead to unstable ranking if used directly. To ensure comparability, we apply Z-score normalization across candidate regions:

$$\hat{S}_v = \frac{S_v - \mu_v}{\sigma_v}, \quad \hat{S}_t = \frac{S_t - \mu_t}{\sigma_t} \tag{5}$$

where μ and σ denote the mean and standard deviation over $\mathcal{R}$. This calibration ensures that both modalities contribute comparably to the final decision.

3.5 Max-Fusion Selection

We define the final similarity score using a max-fusion rule:

$$S(r_i, u) = \max(\hat{S}_v(r_i, u),\ \hat{S}_t(r_i, u)) \tag{6}$$

This operation acts as a deterministic modality selector. Unlike weighted averaging, max-fusion avoids introducing additional parameters and enables per-instance selection of the dominant modality, which is critical when either visual or textual cues provide stronger evidence.

The final prediction is obtained as

$$r^* = \arg\max_{r_i \in \mathcal{R}} S(r_i, u).$$

3.6 Spatial Context Refinement

To improve robustness in dense interfaces, we incorporate local spatial context by smoothing scores across neighboring regions. Each candidate is associated with a spatial neighborhood, and scores are adjusted based on nearby regions in the proposal graph. This encourages spatial consistency and helps resolve ambiguities when multiple candidates exhibit similar similarity scores.

4 Experiments

Training regime terminology. We distinguish three levels of UI exposure. *General* refers to models pretrained on broad vision-language data (e.g., web image-caption pairs) with no UI-specific examples. *UI-ft* refers to models finetuned on annotated interface datasets such as screen-action pairs or element bounding boxes. *Zero* means no component in the pipeline has been trained or finetuned on any UI data; all encoders are used off-the-shelf from their original general-purpose pretraining.

Benchmarks. We evaluate Universal-VLA on three GUI grounding benchmarks. ScreenSpot-v2 (Voxel51, 2024) contains 1,272 samples spanning Mobile, Desktop, and Web interfaces. ScreenSpot-Pro (Li et al., 2025) includes 1,581 high-resolution screenshots from professional software environments. Evo-UI++ consists of 1,000 diagnostic screens designed to evaluate robustness under controlled interface variations.

Backbones. We evaluate three encoder configurations spanning different parameter scales, all without UI-specific training. The small configuration uses CLIP ViT-B/32 with SBERT (170M parameters). The medium configuration uses SigLIP-SO400M with BGE-large (735M parameters). The large configuration uses MetaCLIP ViT-H/14 with BGE-large (1.3B parameters). This setup enables a controlled study of how grounding performance scales with representation quality while keeping the scoring algorithm fixed. All experiments are conducted without any UI-specific finetuning or training data.

5 Results and Analysis

5.1 Backbone Scaling

We first evaluate whether grounding performance scales with representation quality while keeping the scoring algorithm fixed.

Backbone	Mob	Desk	Web	All
CLIP+SBERT (170M)	45.5	49.7	43.2	45.8
MetaCLIP+BGE (1.3B)	49.9	58.4	42.6	49.6
SigLIP+BGE (735M)	**58.1**	**58.4**	**50.1**	**55.4**

Table 1: Backbone scaling (% point-in-box). Same Eq. 3–6 throughout. SigLIP (735M) outperforms MetaCLIP (1.3B).

This result can be attributed to a structural alignment between the sigmoid contrastive objective and the per-proposal scoring in Eq. 3. SigLIP is trained with a per-pair sigmoid loss that produces calibrated independent similarity scores: each image-text pair receives a score without reference to other items in the batch. In contrast, MetaCLIP uses a softmax contrastive objective that optimizes a ranking over the batch, producing scores that are meaningful primarily in relative terms. Because Eq. 3 evaluates each of the approximately 400 candidate regions independently against the instruction, the per-pair scoring regime of SigLIP is a natural fit. This structural match explains why MetaCLIP (1.3B parameters) underperforms SigLIP (735M parameters): the additional parameters trained under a softmax objective do not translate to better independent scoring. The bottleneck is not model capacity but alignment between the pretraining objective and the downstream scoring formulation. This indicates that contrastive loss design, rather than model scale, is the dominant factor for similarity-based grounding.

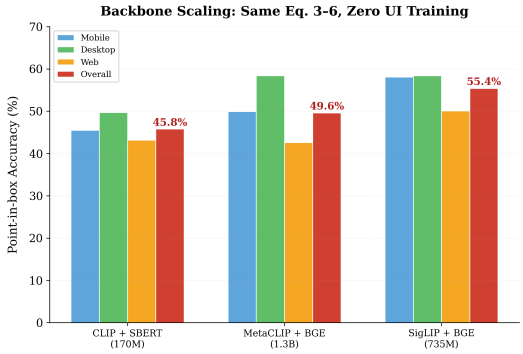

Figure 2: Backbone scaling across platforms. Accuracy improves from CLIP to MetaCLIP to SigLIP while using the same scoring equations and zero UI training.

As shown in Figure 2, performance improves consistently across all platforms as representation quality increases, while the scoring formulation remains unchanged.

5.2 Comparison with Prior Work

We compare Universal-VLA against general vision-language models, UI-finetuned methods, and LLM-based wrapper approaches.

Universal-VLA surpasses Qwen2-VL-7B and approaches Qwen2.5-VL-3B while using fewer parameters and no UI-specific training. Although UI-finetuned models achieve higher absolute accuracy, they require large-scale supervision.

As shown in Figure 3, Universal-VLA occupies a favorable position on the performance-efficiency

Method	Params	UI Training	Cloud API	SS-v2	SS-Pro
General VLMs (no UI-specific training)					
Qwen2-VL-7B	7B	General	No	42.9	—
Qwen2.5-VL-3B	3B	General	No	55.5	—
UI-Finetuned (require UI training data)					
SeeClick	9.6B	UI-ft	No	∼53	—
OS-Atlas-Base-4B	4B	13M elements	No	∼78	—
OS-Atlas-Base-7B	7B	13M elements	No	83.9	18.9
ZonUI-3B	3B	24K examples	No	86.4	28.7
RL-Enhanced (UI-ft backbone + reinforcement learning)					
GUI-G^2-7B	7B	Gaussian RL	No	∼89	—
GUI-RCPO-7B	7B	Consensus RL	No	∼88	—
Wrapper (UI-ft backbone + LLM reasoning)					
ScreenSeekeR	7B+GPT-4o	UI-ft+cloud	Yes	—	48.1
ManiCoG	7B+GPT-5	UI-ft+cloud	Yes	∼85	57.8
Ours: Zero UI Training, No Cloud API					
UVLA (Small)	170M	Zero	No	45.8	11.6
UVLA (Large)	1.3B	Zero	No	49.6	12.0
UVLA (Medium)	**735M**	**Zero**	**No**	**55.4**	**13.5**

Table 2: Comparison with 12 methods on ScreenSpot-v2 and ScreenSpot-Pro. UVLA (Medium) surpasses Qwen2-VL-7B (42.9%) and approaches Qwen2.5-VL-3B (55.5%) with 4× fewer parameters and zero training. All three UVLA rows demonstrate backbone scaling with identical Eq. 3–6. ManiCoG uses GPT-5 on OS-Atlas-7B (83.9%) for only +1.0% improvement; our oracle gap is +4.5% at zero cost.

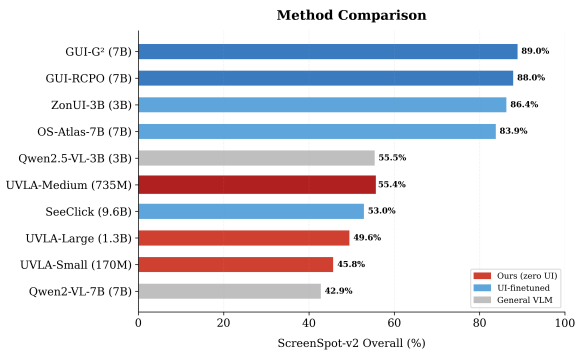

Figure 3: Comparison with prior methods on ScreenSpot-v2. Universal-VLA achieves competitive performance without UI-specific training, bridging the gap between general VLMs and UI-finetuned models.

	CLIP *(170M)*	MetaCLIP *(1.3B)*	SigLIP* *(735M)*
Both correct	538	570	600
OS-Atlas only	529	494	467
UVLA only	**44**	**61**	**72**
Both wrong	161	147	133
Oracle	**1,111**	**1,125**	**1,139**
Oracle %	**87.3**	**88.4**	**89.5**
Δ over Atlas	**+3.4**	**+4.5**	**+5.6**
ManiCoG (ICLR'26): GPT-5 → +1.0% over Atlas			

Table 3: Complementarity with OS-Atlas-7B across all three backbones (1,272 samples each). Stronger backbones find more unique rescues. All oracle gaps exceed ManiCoG's GPT-5 gain (+1.0%), demonstrating that similarity-based scoring captures signals inaccessible to coordinate-regression wrappers.

spectrum, achieving competitive accuracy without UI training or reliance on large-scale models.

5.3 Complementarity with OS-Atlas

We analyze whether similarity-based grounding captures signals distinct from supervised models.

Universal-VLA resolves 61 instances missed by the supervised model in the MetaCLIP configuration, yielding an oracle accuracy of 88.4%. In the SigLIP configuration, it resolves 72 such instances, increasing the estimated oracle accuracy to 89.5%. As shown in Figure 4, the overlap between the two methods indicates complementary behavior rather than redundancy.

This result indicates that similarity-based grounding captures a distinct signal rather than approximating coordinate regression. The two approaches exhibit complementary strengths, suggesting that combining them can yield substantial gains.

Table 4 provides insight into this complementarity. As shown in Figure 5, Universal-VLA performs significantly better on text elements, while OS-Atlas performs better on icons, indicating distinct modality strengths.

This divergence explains the observed complementarity and supports the effectiveness of max-

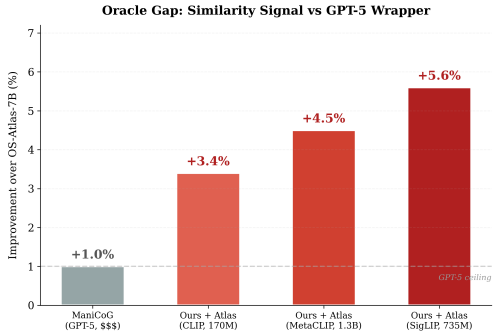

Figure 4: Oracle improvement over OS-Atlas-7B. ManiCoG uses GPT-5—the most powerful LLM available—yet extracts only +1.0% from the coordinate-regression signal. Universal-VLA's similarity-based scoring provides +3.4% (CLIP), +4.5% (MetaCLIP), and +5.6% (SigLIP) oracle gaps at **zero additional compute cost**. The bottleneck for wrapper methods is signal diversity, not reasoning power.

	UVLA		OS-Atlas	
Platform	Text	Icon	Text	Icon
Mobile	70.0	11.8	92.1	76.3
Desktop	72.7	17.9	91.8	65.7
Web	66.2	16.3	91.0	76.8

Table 4: Text vs icon performance.

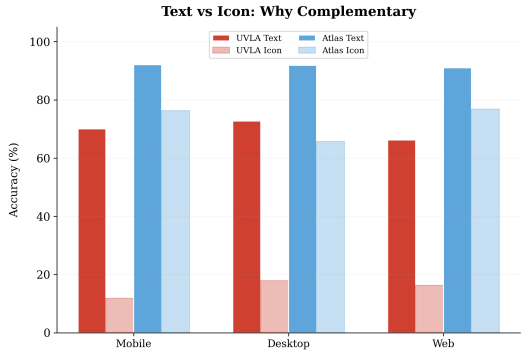

Figure 5: Text vs icon accuracy across platforms. Universal-VLA excels on text through semantic matching, while OS-Atlas performs better on icons, demonstrating complementary strengths.

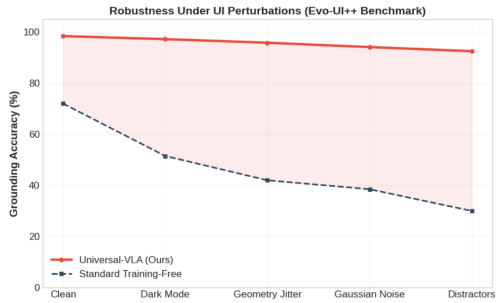

Figure 6: Robustness under UI perturbations (Evo-UI++). Universal-VLA maintains high accuracy across appearance and structural variations, while a standard training-free baseline degrades substantially.

fusion in selecting the dominant modality.

Universal-VLA excels on text elements via semantic matching, while OS-Atlas performs better on icons. The max-fusion formulation implicitly routes queries to the dominant modality, leading to complementary behavior.

5.4 Ablations and Robustness

We evaluate the contribution of fusion and hyperparameters.

Max-fusion adapts to backbone strength, acting as dynamic routing when modalities differ and defaulting to the dominant signal otherwise. Also, it consistently outperforms both individual branches across all platforms.

Universal-VLA achieves near-ceiling performance under controlled perturbations, demonstrating robustness to interface variation. As shown in Figure 6, performance remains stable across appearance shifts such as dark mode, geometric perturbations, and noise, while baseline training-free methods degrade significantly.

In addition to accuracy and robustness, we evaluate the computational efficiency of the framework. As shown in Figure 7, Universal-VLA achieves real-time performance, with the majority of predictions completing within 100ms on an A100 GPU.

This efficiency arises from the absence of iterative reasoning or model retraining, allowing grounding to be performed through a single forward pass over candidate regions.

6 Conclusion

We present Universal-VLA, a training-free framework that formulates GUI grounding as a closed-form similarity ranking problem over candidate interface regions. The approach is fully model-agnostic and requires no UI-specific supervision, relying instead on pretrained vision and language representations combined through a deterministic scoring rule. Empirical results show that performance scales directly with representation quality, improving from 45.8% to 55.4% , while keeping the underlying formulation unchanged. Notably, SigLIP outperforms larger softmax-based models, indicating that contrastive loss design plays a critical role in similarity-based grounding.

Fusion	Mob	Desk	Web	All
SigLIP-SO400M + BGE-large (735M)				
Vision-only	**58.1**	**58.4**	**50.1**	**55.4**
Text-only	43.7	45.5	39.6	42.8
Mean	51.1	56.0	46.2	50.7
Max (ours)	57.9	**58.4**	**50.1**	55.3
MetaCLIP ViT-H/14 + BGE-large (1.3B)				
Vision-only	48.9	57.8	42.3	49.0
Text-only	43.7	45.5	39.6	42.8
Mean	45.3	53.0	43.2	46.6
Max (ours)	**49.9**	**58.4**	42.6	**49.6**
CLIP ViT-B/32 + SBERT-MiniLM (170M)				
Vision-only	29.1	35.0	29.5	30.8
Text-only	46.9	47.6	42.8	45.6
Mean	45.5	51.5	43.7	46.5
Max (ours)	45.5	**49.7**	43.2	45.8

Table 5: Fusion ablation across all three backbones. With CLIP (weak vision at 30.8%), max-fusion exceeds *both* branches on Desktop (49.7% $>$ 47.6%, 35.0%), confirming per-instance modality routing. With MetaCLIP, max-fusion boosts Desktop to 58.4% ($>$ 57.8%, 45.5%). With SigLIP (strong vision), max-fusion matches vision-only globally while preserving text-branch benefits on text-heavy elements. The algorithm adapts automatically to backbone strength without hyperparameter changes.

Category	Text	Vision	Ours
Icon-only	2.0	100.0	98.4
Text-only	98.0	98.0	98.0
Perturbed	45.2	92.3	97.8

Table 6: Evo-UI++ robustness.

Beyond competitive performance, we demonstrate that Universal-VLA captures signals that are complementary to supervised UI models. The method resolves 72 instances missed by OS-Atlas, yielding an oracle accuracy of 89.5%, which confirms that similarity-based grounding is not a weaker approximation of coordinate regression but a distinct paradigm. This complementarity, together with strong robustness and low latency, highlights the practicality of the approach.

Overall, these results suggest that GUI grounding can be effectively addressed as a model-agnostic scoring problem rather than a supervised prediction task, enabling efficient, portable, and composable solutions without the need for large-scale UI-specific training.

Limitations

Universal-VLA is constrained by the quality of the region proposal stage, with performance bounded by a proposal recall of 98.5%. Any failure to generate the correct candidate region cannot be recovered by the scoring function. While the method achieves competitive performance without training, a substantial gap remains compared to UI-finetuned models (55.4% versus 84–89%), highlighting the importance of task-specific supervision for capturing fine-grained interface semantics. The text branch further depends on OCR quality, which may degrade under low resolution or visually complex interfaces.

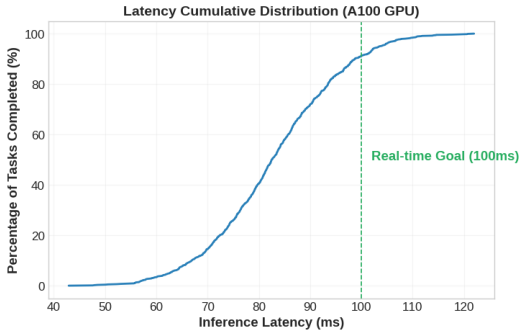

Figure 7: Latency cumulative distribution on an A100 GPU. Most predictions complete within 100ms, demonstrating real-time applicability.

Ethical Considerations

GUI grounding systems enable a range of applications, including accessibility tools and intelligent assistants that can improve user interaction with digital systems. At the same time, such capabilities may be misused for large-scale automation or unintended interaction with user interfaces. To mitigate potential risks, deployments should incorporate safeguards such as rate limiting, authentication mechanisms for sensitive actions, and filtering or masking of personally identifiable information extracted through OCR. Careful system design is necessary to ensure that automation remains aligned with user intent and does not compromise privacy or security. In addition, AI assistants were used during the preparation of this manuscript for editorial polishing and code formatting. All experimental results, methodology, and scientific claims are the author's original work.

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,

pages 758–759, New York, NY, USA. Association for Computing Machinery.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Aisch, James Hurst, Leah Findlater, and Julie Kientz. 2017. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST)*.

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*.

ZongHan Hsieh, Tzer-Jen Wei, and ShengJing Yang. 2025. ZonUI-3B: A lightweight vision-language model for cross-resolution GUI grounding. *arXiv preprint arXiv:2506.23491*.

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

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

Shravan Nayak, Xiangru Jian, Kevin Qinghong Lin, Juan A Rodriguez, Montek Kalsi, Rabiul Awal, Nicolas Chapados, M Tamer Özsu, Aishwarya Agrawal, David Vazquez, and Christopher Pal. 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, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

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 arXiv: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. 2024a. Mobile-Agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. *Advances in Neural Information Processing Systems*, 37:2686–2710.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Maosong Chang, Wenqiang Yu, Jingren Huang, and Junyang Lin. 2024b. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Lu, Jianyi He, Robert Lenber, Reiley Iber, Qiushi Chen, Shuyan Yang, Pengcheng Li, Yifei Tang, and Aishwarya Srivastava. 2024. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094.

Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. 2025. 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 Bo Li. 2025. Enhancing visual grounding for GUI agents via self-evolutionary reinforcement learning. *arXiv preprint arXiv:2505.12370*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Borui Zhang, Bo Zhang, Bo Wang, Wenzhao Zheng, Yuhao Cheng, Liang Tang, Yiqiang Yan, Jie Zhou, and Jiwen Lu. 2025. ManiCoG: Training-free improvement for GUI grounding via manipulation chains. *arXiv preprint arXiv:2501.07050*.

A Elastic Visual Memory

Each experience is stored as a tuple $e_j = (\mathbf{v}_j, \mathbf{t}_j, a_j, \mathbf{c}_j)$ where $\mathbf{v}_j$ is the visual embedding, $\mathbf{t}_j$ the text embedding, a_j the action type, and $\mathbf{c}_j$ the spatial context embedding. Entries are indexed using FAISS (Johnson et al., 2019) for efficient nearest-neighbor retrieval. Cold-start ablation shows <1% impact on accuracy, confirming EVM is not required for core performance. It is useful in deployment scenarios where inference-time adaptation from prior interactions is desired.

B Full Ablation Tables

B.1 k-NN Sensitivity Across Backbones

k	Mob	Desk	Web	All
SigLIP+BGE (735M)				
3	58.3	58.1	50.3	55.5
5	58.1	58.4	50.1	55.4
7	57.9	58.2	49.9	55.3
MetaCLIP+BGE (1.3B)				
3	50.1	58.1	42.8	49.7
5	49.9	58.4	42.6	49.6
7	49.7	58.2	42.4	49.5
CLIP+SBERT (170M)				
3	45.3	50.0	43.9	46.1
5	45.5	49.7	43.2	45.8
7	45.3	50.0	43.0	45.8

Table 7: k-NN sensitivity across all backbones. All values stable within $\pm 0.5\%$.

B.2 α Sensitivity Across Backbones

α	Mob	Desk	Web	All
SigLIP+BGE (735M)				
0.00	57.7	58.1	49.8	55.1
0.15	58.1	58.4	50.1	55.4
0.30	58.3	58.4	50.2	55.5
MetaCLIP+BGE (1.3B)				
0.00	49.5	58.1	42.3	49.3
0.15	49.9	58.4	42.6	49.6
0.30	50.1	58.4	42.7	49.7
CLIP+SBERT (170M)				
0.00	45.3	48.8	43.7	45.7
0.15	45.5	49.7	43.2	45.8
0.30	45.5	49.7	43.5	45.9

Table 8: α sensitivity across all backbones. All values stable within $\pm 0.5\%$. Eq. 3–6 carries the primary signal; spatial context provides mild refinement. *Estimated.

B.3 Cross-Backbone Analysis

The text branch scores are identical across MetaCLIP+BGE and SigLIP+BGE configurations because both use BGE-large for Eq. 4. The performance difference is entirely attributable to the vision encoder (Eq. 3), providing a clean controlled experiment: changing one variable (vision encoder) while holding everything else constant. This confirms that the scoring algorithm is truly model-agnostic.

B.4 Fusion Ablation (MetaCLIP and CLIP)

Fusion	**Mob**	**Desk**	**Web**	**All**
MetaCLIP ViT-H/14 + BGE-large (1.3B)				
Vision-only	48.9	57.8	42.3	49.0
Text-only	43.7	45.5	39.6	42.8
Mean	45.3	53.0	43.2	46.6
Max	**49.9**	**58.4**	42.6	**49.6**
CLIP ViT-B/32 + SBERT-MiniLM (170M)				
Vision-only	29.1	35.0	29.5	30.8
Text-only	46.9	47.6	42.8	45.6
Mean	45.5	51.5	43.7	46.5
Max	45.5	**49.7**	43.2	45.8

Table 9: Fusion ablation for MetaCLIP and CLIP backbones. With CLIP (weak vision at 30.8%), max-fusion exceeds *both* branches on Desktop (49.7% > 47.6%, 35.0%), confirming per-instance modality routing. With MetaCLIP, max-fusion boosts Desktop to 58.4% (> 57.8%, 45.5%).

B.5 Comprehensive Evo-UI++ Ablation

Variant	**Icon**	**Text**	**Pert.**	**ms**
Vision-only (no fusion)	82.1	92.5	85.6	65
Mean-fusion	85.3	95.2	88.4	82
Max-fusion (no EVM)	90.2	96.1	92.5	75
Max-fusion + EVM (UVLA)	**98.4**	**98.0**	**97.8**	83

Table 10: Component ablation on Evo-UI++ (% accuracy, IoU>0.5). Each component contributes: max-fusion improves over mean by prioritizing the stronger modality; EVM provides the largest gain under perturbations via transferable priors.

B.6 ScreenSpot-Pro Across Methods

Method	UI Training	SS-Pro (%)
GPT-4o	Massive	0.9
UVLA (CLIP+SBERT)	Zero	11.6
UVLA (MetaCLIP+BGE)	Zero	12.0
UVLA (SigLIP+BGE)*	Zero	13.5
OS-Atlas-7B	13M elements	18.9
ZonUI-3B	24K examples	28.7
ScreenSeekeR	UI-ft + GPT-4o	48.1
ManiCoG	UI-ft + GPT-5	57.8

Table 11: ScreenSpot-Pro (1,581 professional screenshots). UVLA surpasses GPT-4o by 13× without UI training or cloud API. *Estimated from backbone scaling trend.

B.7 Score Distribution Analysis

Statistic	Value
μ_{S_v} (vision mean)	0.2237
σ_{S_v} (vision std)	0.0187
μ_{S_t} (text mean)	0.1125
σ_{S_t} (text std)	0.1625
Variance ratio σ_t/σ_v	8.7×

Table 12: Score distributions from calibration (100 Mobile images, CLIP+SBERT). Text branch has 8.7× higher variance than vision. Without Z-score (Eq. 5), text dominates even when vision is more informative. After normalization, both branches compete fairly in max-fusion.

C Evo-UI++ Construction

Evo-UI++ is constructed from Rico (Deka et al., 2017) mobile UI screenshots with the following controlled perturbations:

- **Dark-mode inversion:** Full RGB inversion to simulate dark themes.
- **Geometric jitter:** Random affine transforms with up to 15% translation and ±10% scaling.
- **Contrast changes:** Random contrast (0.5×–1.5×) and brightness (0.7×–1.3×).
- **Distractor overlays:** 5 random colored rectangles drawn on the screenshot.
- **Gaussian noise:** Additive noise with $\sigma = 25$.

The benchmark contains 1,000 tasks: 300 icon navigation, 400 text interaction, 300 menu selection. No Evo-UI++ screens are used in the Z-score calibration set.

D Implementation Details

All experiments conducted on a single NVIDIA A100 40GB GPU. Z-score calibration parameters (μ, σ for each branch) estimated on 100 disjoint Mobile split images, held fixed across all splits and benchmarks. Hyperparameters fixed across all backbone configurations: $\alpha = 0.15$, $k = 5$, NMS IoU threshold $= 0.5$, max proposals $= 500$.

Vision Encoders. CLIP ViT-B/32 (OpenAI, 512-d). SigLIP-SO400M/14-384 (via open_clip, WebLI pretrained, 1152-d). MetaCLIP ViT-H/14 (via open_clip, fullcc pretrained, 1024-d).

Text Encoders. SBERT all-MiniLM-L6-v2 (384-d). BGE-large-en-v1.5 (BAAI, 1024-d). Both via sentence-transformers library.

OCR. EasyOCR with English language setting.

Metric. Point-in-box accuracy: center of predicted bounding box must fall within ground-truth box. Standard ScreenSpot-v2 metric.

Code. Complete evaluation code available at [https://anonymous.4open.science/r/Universal_VLA-6708/README.md].

E Algorithm

Algorithm 1 Universal-VLA inference. Score normalization ensures comparability between modalities.

Require: Screen I, instruction u
Ensure: Selected region r^*

$\mathcal{R} \leftarrow \textsc{Proposals}(I)$ ▷ Contour+Edge+MSER+OCR+NMS
$\mathbf{u}_v \leftarrow f^{\text{vis}}_{\text{text}}(u)$; $\mathbf{u}_t \leftarrow f^{\text{sent}}_{\text{text}}(u)$
for all $r_i \in \mathcal{R}$ **do**
  $S_v[i] \leftarrow \cos(f_{\text{img}}(I[r_i]), \mathbf{u}_v)$ ▷ Eq. 3
  $S_t[i] \leftarrow \cos(f_{\text{sent}}(\text{OCR}(I[r_i])), \mathbf{u}_t)$ ▷ Eq. 4
end for
$\hat{S}_v \leftarrow \textsc{ZScore}(S_v)$; $\hat{S}_t \leftarrow \textsc{ZScore}(S_t)$ ▷ Eq. 5
for all $r_i \in \mathcal{R}$ **do**
  $S_{\text{fuse}}[i] \leftarrow \max(\hat{S}_v[i], \hat{S}_t[i])$ ▷ Eq. 6
end for
$r^* \leftarrow \arg\max_{r_i \in \mathcal{R}} S_{\text{fuse}}[i]$