Mitigating Coordinate Prediction Bias from Positional Encoding Failures

Xingjian Tao[1], Yiwei Wang[3], Yujun Cai[4], Yihong Luo[2], Kai Han[5], and Jing Tang[1,2]*

[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]The Hong Kong University of Science and Technology [3]University of California, Los Angeles
[4]The University of Queensland [5]Shanghai University of Finance and Economics
taoxj2001@outlook.com

Abstract

While Multimodal Large Language Models (MLLMs) excel at general vision-language tasks, precise coordinate prediction remains a significant challenge, particularly as high-resolution inputs cause visual positional encodings (VPEs) to degrade. We demonstrate that these encoding failures do not result in random noise but instead trigger predictable, directional biases, suggesting that models default to internal spatial priors when grounding signals are weak. To counteract this, we introduce Vision-PE Shuffle Guidance (VPSG), a training-free, inference-time correction method. VPSG isolates position-unconditioned tendencies by shuffling VPEs and utilizes this negative evidence to steer digit decoding through a lightweight finite-state machine. Evaluation on the ScreenSpot-Pro benchmark confirms that VPSG effectively rectifies coordinate drift, yielding consistent improvements in localization accuracy across various model scales without any retraining. Our code is available at https://github.com/taoxj2001/VPSG

1 Introduction

Recent advances in large language models (LLMs; Touvron et al. 2023a; Chiang et al. 2023; Almazrouei et al. 2023; MosaicML 2023; Touvron et al. 2023b; OpenAI 2022; Google 2023) have improved language understanding and generation, but their text-only I/O limits perceptual and interactive use. Multi-modal LLMs (MLLMs) combine vision and text, e.g., Flamingo (Alayrac et al., 2022), Gemini (Team et al., 2023), and Qwen-VL (Bai et al., 2023; Wang et al., 2024; Bai et al., 2025) to enable tasks such as visual QA, captioning, and document understanding. Coordinate prediction supports applications like object manipulation and GUI automation, where an MLLM outputs a 2-D point or bounding box (e.g., "[1000,500]"). High-resolution inputs make this harder: token and compute costs rise, and larger spatial extents increase pixel–patch misalignment, leading to coordinate drift (Li et al., 2025; Gao et al., 2024; Hsieh et al., 2024; Yen et al., 2024).

Coordinate drift originates from the failure of conventional positional encodings (PEs) to scale reliably at high resolutions (Zhang et al., 2024). Although essential for anchoring visual tokens to image geometry , standard schemes—typically 2-D encodings in the vision encoder (Dosovitskiy et al., 2020) and sequence-level RoPE (Su et al., 2024) in the LLM —suffer as high-resolution inputs induce long-context regimes that diffuse attention and weaken spatial cues. Existing remedies are insufficient for precise localization: cropping-based methods (Tao et al., 2025; Wu et al., 2025) often compromise global semantics , while PE enhancements (Ge et al., 2024; Chen et al., 2025b; Heo et al., 2024) address coarse tasks like VQA but fail to mitigate the subtle biases that drive numerical coordinate errors.

Coordinate prediction requires precise spatial awareness; however, unlike humans who resort to random guessing when positional signals are shuffled, MLLMs exhibit predictable *directional biases*. These patterns closely mirror error distributions in natural high-resolution datasets, suggesting that the weakening of positional information at scale amplifies the model's inherent priors and degrades coordinate accuracy as shown in Figure 1. In this work, we seek to mitigate these biases and reinforce positional encodings to improve MLLM robustness on position-sensitive tasks.

We propose Vision-PE Shuffle Guidance (VPSG), a training-free, test-time method that rectifies coordinate biases online using only the base model. VPSG contrasts a position-conditioned main route with auxiliary routes that approximate a position-unconditioned reference by shuffling visual positional encodings. This fusion amplifies

*Corresponding Author: Jing Tang.

Findings of the Association for Computational Linguistics: ACL 2026, pages 20635–20650
July 2-7, 2026 ©2026 Association for Computational Linguistics

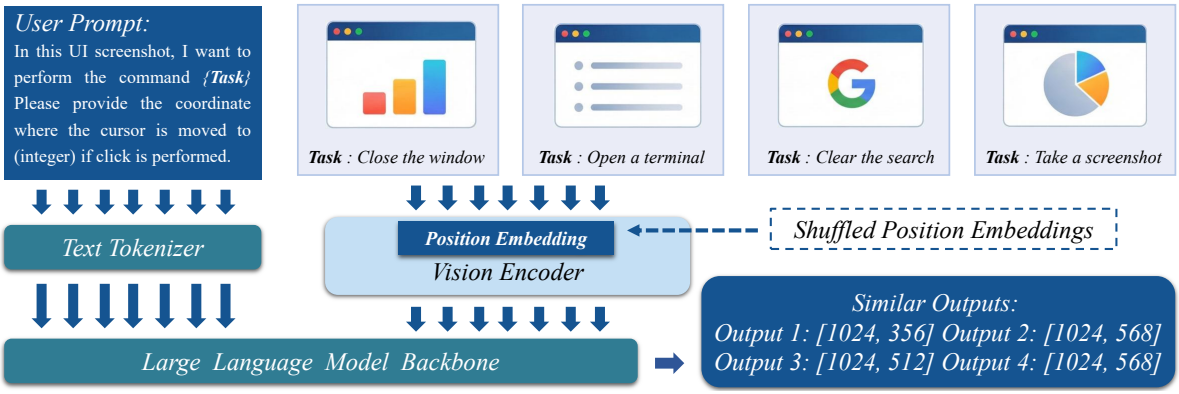

Figure 1: Effect of shuffling visual positional encodings: removing spatial conditioning causes the model to collapse to similar coordinate predictions across independent runs, indicating a position-unconditioned, directional bias rather than random variation. We also observe a similar clustered error pattern on high-resolution images (without shuffling), consistent with position-encoding failures.

spatially grounded information while suppressing spurious content that persists even when positional cues are removed. Practically, a lightweight FSM ensures that only digit tokens are adjusted—down-weighting position-free tendencies while leaving formatting characters like commas and brackets untouched. This mechanism functions as a scaled "negative score" during decoding, effectively reinforcing positional influence and stabilizing $[x, y]$ outputs without requiring any training or architectural modifications.

Two key design choices make VPSG precise and stable. Rather than relying on a single PE-shuffled auxiliary route, VPSG aggregates multiple shuffled routes in log space (geometric mean), yielding a robust estimate of the position-unconditioned bias and stabilizing the negative-evidence signal across inputs. In addition, VPSG applies a position-aware coefficient schedule: the guidance weight starts high for the first digit of x, decays geometrically for subsequent digits, resets at the first digit of y, and then decays again. This concentrates correction on the most influential digits while avoiding over-regularization of later positions and preserving natural numeric formatting. Our contributions can be summarized as follows:

- **Positional fragility analysis.** We show that perturbing visual positional encodings (VPEs) induces directional, repeatable biases in MLLM coordinate prediction, with similar effects at high resolution—revealing a resolution-dependent failure mode.
- **Training-free guidance.** We propose VPSG, a model-agnostic test-time method that shuffles VPEs to form counterfactual routes and guides final-layer digit logits via a lightweight finite-state machine.

2 Related Work

Coordinate prediction with MLLMs With the advancement of MLLMs (Bai et al., 2023; Wang et al., 2024; Bai et al., 2025; Alayrac et al., 2022; Team et al., 2023; Ma et al., 2023; Yang et al., 2023; Liu et al., 2023a; Li et al., 2023; Liu et al., 2023b; Xu et al., 2025), coordinate prediction has shifted to language-driven grounding, where coordinates are generated as discrete token sequences. This paradigm is vital for GUI interaction, supported by specialized datasets designed for screenshot-based grounding (Cheng et al., 2024; Wu et al., 2024; Li et al., 2025). To enhance UI navigation and agent autonomy, data-centric methods such as ShowUI (Lin et al., 2024) and UGround (Gou et al., 2025) utilize large-scale synthetic datasets. Furthermore, specialized frameworks like OmniParser (Lu et al., 2024) leverage auxiliary visual models to improve element localization , while WebGUM (Furuta et al., 2023) integrates hierarchical planning with perceptual grounding to enable structured web-based decision-making.

Visual position encoding Visual position encoding is essential for scaling multimodal transformers (Wang et al., 2025). Variable schemes like V2PE (Ge et al., 2024) and PyPE (Chen et al., 2025b) improve robustness in long context perception. Lightweight 2D markers enhance layout understanding in structured document tasks (Liao et al., 2025). RoPE provides extrapolation benefits

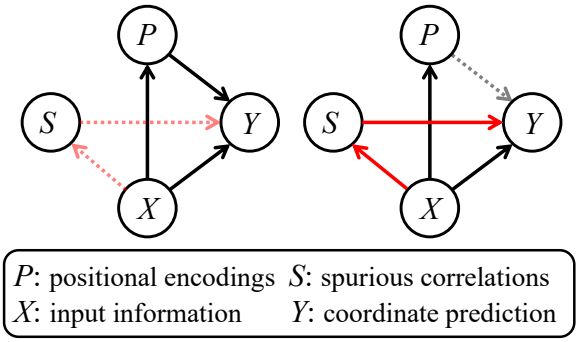

Figure 2: Causal view of coordinate prediction. Image content and prompt provide the intended causal effect on output digits (left). When VPEs are missing (e.g., at high resolution), the model relies on spurious correlations, leading to directional digit biases (right).

and semantic aware encodings adapt to perceptual similarity (Heo et al., 2024; Chen et al., 2025a). Recent studies generalize RoPE through Fourier analysis (Hua et al., 2024) or trainable matrices (Yu et al., 2025) and explore pooling interactions (Lee et al., 2025). Large vision embedding norms can suppress positional information and cause spatial reasoning failures Qi et al. (Qi et al., 2025). These results show that robust encodings are central to model advancement. However research on tasks requiring precise location information like coordinate prediction remains scarce.

3 Methodology

In this section, we introduce Vision-PE Shuffle Guidance (VPSG), a training-free, test-time guidance that stabilizes coordinate outputs in multimodal LLMs. VPSG runs the base model once on the normal input and in parallel creates shuffle-guided auxiliary views by perturbing only the visual positional encodings; discrepancies across these views reveal spurious numeric tendencies. Using this as negative evidence, VPSG gently steers the main decoding on digit tokens (leaving non-digits untouched) and improves $[x, y]$ reliability without fine-tuning or architectural changes.

3.1 Causal view of coordinate prediction

At the core of our study lies the observation that coordinate prediction in multimodal LLMs is governed by a causal mechanism: the output depends simultaneously on position-conditioned signals (e.g., visual positional encodings) and position-unconditioned signals (e.g., default digit tendencies). Prior analyses largely overlook the influence of these position-independent inputs, treating prediction errors as random noise. In contrast, we hypothesize that the non-positional pathway can introduce directional bias when positional information is weak or missing. Motivated by this insight, we adopt a causal graph (Pearl et al., 2016; Pearl, 2018; Tang et al., 2020; Wang et al., 2022) as the analytical framework to explicitly model how position-related and position-free factors jointly shape the output and to identify spurious routes that degrade coordinate accuracy. Ideally, the output digits $[x, y]$ are determined jointly by the image content and the textual prompt, with visual positional encodings (VPEs) supplying spatial grounding. However, when VPEs are missing or unreliable—such as in high-resolution inputs, the causal pathway is disrupted. Lacking accurate positional cues, the model tends to rely on spurious correlations, for example overpredicting certain digits or repeating biased numeric patterns that are not grounded in the image.

As shown in Figure 2, we model coordinate prediction with a simple structural view over four nodes: input information X (image content and prompt), positional encodings P (visual spatial cues), spurious correlations S (default numeric tendencies that emerge independently of the input, such as frequently repeated digits or preferred coordinate patterns), and the predicted coordinates Y. Here, S captures position-unconditioned regularities in the model's training data or internal priors that can influence outputs even when visual evidence is weak or missing. A minimal Structural Causal Model is $Y = g(X, P, S)$, where P is provided by the vision encoder (and varies with resolution), and S summarizes non-causal numeric regularities the model can fall back on.

When P is available and reliable, it supplies spatial grounding. The dominant pathways are $X \to Y$ and $P \to Y$. The influence of S is negligible (dotted edges): although spurious patterns exist in the model's priors, they are largely blocked in practice because the model can rely on informative X and P. When P is absent or unreliable, the informative pathway weakens ($P \dashrightarrow Y$), and the spurious path without positional condition $S \to Y$ becomes comparatively strong. The model then defaults to biased numeric templates (e.g., over-predicting certain digits or repeating patterns) that are not supported by the input. In potential-outcome terms, the discrepancy $S(x) = Y(x, p_{\text{bad}}) - Y(x, p_{\text{good}})$ captures the shift in predictions attributable to the

loss of positional grounding, with p_{good} denoting a reliable PE setting and p_{bad} an out-of-range or missing one.

This causal view clarifies the failure mode: positional degradation amplifies the non-causal route from S to Y, yielding directional digit errors even when X is unchanged. It also motivates our methodology: design a test-time procedure that (i) exposes the spurious route when P is weak and (ii) suppresses its influence on the numeric tokens while preserving the informative flow from X (and any usable P).

3.2 Bias analysis

Given the causal graph in Figure 2, weakening the positional encodings P increases the relative influence of spurious correlations S on the output Y, distorting the digit distribution and pulling predictions away from the evidence in X. We therefore propose a two-part intervention: (i) expose the spurious route when P is weak by constructing counterfactual views that differ only in positional cues, and (ii) suppress its impact on numeric tokens while preserving the informative flow from X (and any usable P). Before presenting the intervention, we first establish empirically that the resulting errors are directional rather than random, confirming that the $S \to Y$ pathway is measurable.

Let x be an input of size (W_x, H_x) with diagonal $d_x = \sqrt{W_x^2 + H_x^2}$ to normalize scale across images. Under shuffled PEs, we run the model on S cases and compute pairwise distances

$$d^{(i,j)}(x) = \left\| \hat{\mathbf{y}}^{(i)}(x) - \hat{\mathbf{y}}^{(j)}(x) \right\|_2, \qquad 1 \le i < j \le S, \tag{1}$$

then normalize as $\tilde{d}^{(i,j)}(x) = \frac{d^{(i,j)}(x)}{d_x}$. Pooling $\tilde{d}^{(i,j)}(x)$ over all inputs yields the shuffled-PE distance distribution $\mathcal{P}_{\text{shuffle}}$, while normal PEs give $\mathcal{P}_{\text{normal}}$. We compare their empirical means, against the scale-aware baseline $\mu_0 \approx 0.5214$ (Appendix E) to assess whether shuffled predictions collapse toward a small coordinate subset. Evidence of systematic, non-random bias is a clear left shift of $\mathcal{P}_{\text{shuffle}}$ toward zero relative to both μ_0 (i.e., $\mathbb{E}[\tilde{d}] \ll \mu_0$) and $\mathcal{P}_{\text{normal}}$, indicating that predictions under shuffled PEs collapse to a few favored coordinates rather than dispersing as random fluctuations would.

Figure 3 summarizes the distance statistics. Across both Qwen2.5-VL-3B and Qwen2.5-VL-7B, the diagonal-normalized average pairwise distance under shuffled positional encodings is consistently small ($\tilde{d} \approx 0.16$), whereas the normal-PE condition exhibits substantially larger dispersion ($\tilde{d} \approx 0.40$–0.44). This substantial gap, far exceeding the baseline dispersion of random uniform points, confirms that when positional encodings are disrupted the model outputs collapse to a small set of preferred coordinates rather than spreading randomly. The consistent pattern across model scales demonstrates that the observed systematic directional bias is an inherent property of the architecture rather than a size-related artifact.

Based on the empirical and causal analysis of systematic bias caused by missing positional encodings presented above, we propose a bias-reduction strategy that remains unchanged during training but is inserted during inference: Vision-PE Shuffle Guidance (VPSG). Inspired by classifier-free guidance (CFG) (Ho and Salimans, 2022), we mitigate the impact of directional bias from the perspective of probability distribution.

Overall algorithm. VPSG runs one *main route* with normal visual positional encodings (PEs) and several *auxiliary routes* with randomly shuffled PEs. We let c_t denote the position-conditioned context at step t, which primarily captures the positional encoding information from the visual encoder that provides spatial grounding for decoding. At each decoding step, we contrast the **position-conditioned** prediction $p_A(v \mid c_t)$ from the main route with an aggregated, **position-unconditioned** reference $p_B(v)$ formed by combining multiple shuffled routes. This contrast acts as *negative evidence* on digit tokens, while non-digit tokens (commas, spaces, brackets) remain untouched as shown in Figure 4. A finite-state machine (FSM) tracks whether the model is decoding the x or y coordinate and indicates when digit-specific guidance should be applied. This mechanism preserves the required $[x, y]$ format throughout decoding and prevents structural errors that could arise from spurious tokens or misaligned guidance.

Proposition 1 (VPSG token guidance)**.** *Let $\mathcal{D} \subset \mathcal{V}$ denote the digit subset of the vocabulary and α_t the step-wise guidance coefficient determined by the FSM. The VPSG-adjusted distribution satisfies*

$$p_{\text{VPSG}}(v \mid c_t) \propto \begin{cases} \exp\big(\log p_A(v \mid c_t) - \alpha_t \tilde{\ell}_B(v)\big), & v \in \mathcal{D}, \\ p_A(v \mid c_t), & v \notin \mathcal{D}, \end{cases} \tag{2}$$

where $\tilde{\ell}_B(v) = \log p_B(v)$ is the log-probability of

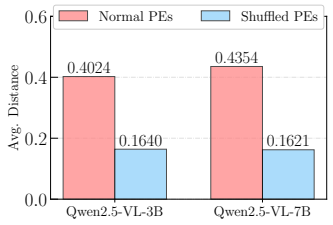

Figure 3: Diagonal-normalized average pairwise distance $\tilde{d}$ between coordinate predictions under *Normal PEs* and *Shuffled PEs*.

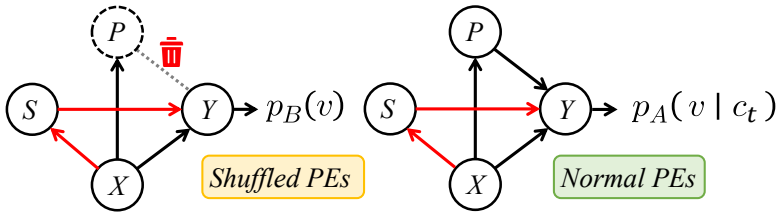

Figure 4: We obtain a position-*unconditioned* reference by shuffling visual positional encodings (left) and fuse it with the position-*conditioned* prediction (right). This conditional–unconditional contrast serves as negative evidence for digits, strengthening positional cues and suppressing spurious numeric patterns during decoding.

the position-unconditioned reference.

This compact formula shows that VPSG subtracts a scaled log-probability from the digit logits of the main route while leaving non-digit tokens unchanged. The aggregation of $p_B(v)$ and the scheduling of α_t are described below; the proof that this form is equivalent to the classifier-free guidance view is provided in Appendix B.

Seeds aggregation. The auxiliary reference $p_B(v)$ is estimated by running the model on S independent PE-shuffled seeds and aggregating their log-probabilities,

$$\tilde{\ell}_B(v) = \mathbb{E}_{\sigma\sim\nu}\Big[\log p_B^{(\sigma)}(v)\Big] = \int_{\sigma\in\mathcal{S}} \log p_B^{(\sigma)}(v)\, d\nu(\sigma) \approx \frac{1}{S}\sum_{s=1}^{S} \log p_B^{(s)}(v), \tag{3}$$

where σ indexes shuffle transformations and ν is the (uniform) seed measure. This log-space (geometric mean) aggregation provides a robust Monte-Carlo estimate of the position-unconditioned bias prior.

Coefficient decay. To focus correction on the most influential digits, VPSG uses a geometric decay on α_t along the decoding sequence. Let k_x (resp. k_y) be the index of the current digit within x (resp. y). The guidance coefficient is scheduled as

$$\alpha_t = \begin{cases} \alpha\,\mathrm{decay}^{k_x-1}, & \text{the } k_x\text{-th digit of } x, \\ \alpha, & \text{first digit of } y, \\ \alpha\,\mathrm{decay}^{k_y-1}, & \text{the } k_y\text{-th digit of } y, \end{cases} \tag{4}$$

where $0 < \mathrm{decay} < 1$. This schedule emphasizes the most significant digits, resets at the first y digit, and then tapers off, preventing over-regularization on later positions.

Summary. Overall, VPSG addresses this by effectively "asking twice": once with the normal input and again with a version whose visual positions are shuffled. Disagreements from the shuffled view act as negative evidence, gently steering the model's digit choices back toward what the image supports, while leaving non-digit text untouched. The result is a simple, plug-in procedure at inference time that stabilizes $[x, y]$ predictions without changing training or model architecture. It is model-agnostic, requires no retraining, and exactly recovers the baseline behavior when $\alpha_t \to 0$. The complete VPSG algorithm is shown in Section D. As shown in Figure 5, we provide a qualitative example of VPSG on a Screenspot-Pro case.

4 Experiments

In this section, we evaluate the performance of VPSG when applied to VLLMs. As a plug-and-play, training-free approach, VPSG enhances existing models on coordinate prediction tasks, providing consistent improvements without requiring additional fine-tuning or architectural modifications.

4.1 Experimental settings

Datasets. We adopt the widely used ScreenSpot-Pro dataset to evaluate the performance of our method. ScreenSpot-Pro is a recently released benchmark for GUI grounding, consisting of real high-resolution desktop screenshots spanning 23 applications (e.g., VSCode, Photoshop, AutoCAD), five industry categories, and three operating systems, with precise annotations provided by professional users as shown in Section C. The dataset is particularly challenging because target UI elements are often extremely small, occupying on average only 0.07% of the screen area. We evaluate VPSG

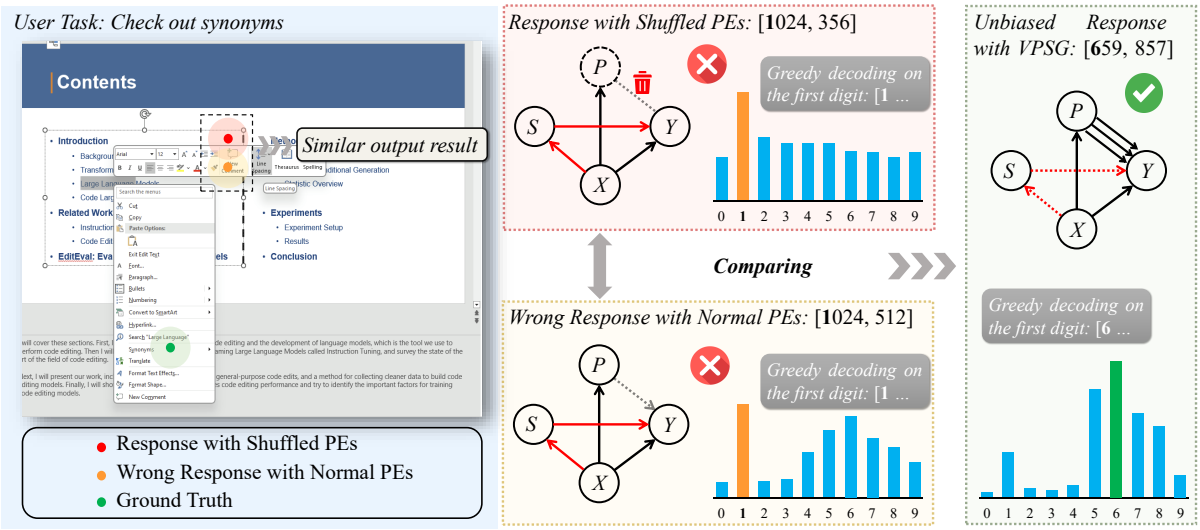

Figure 5: Qualitative example of VPSG on a Screenspot-Pro case. The base model with normal positional encodings produces a biased coordinate prediction ([1024, 512]), while the same model with shuffled positional encodings collapses to a similar but consistently biased point ([1024, 356]), revealing a directional position-unconditioned tendency. Applying VPSG corrects this bias and outputs the accurate ground truth ([659, 857]) by integrating negative evidence from multiple shuffled runs and reweighting digit logits, demonstrating how VPSG suppresses spurious patterns and restores faithful spatial grounding.

under this realistic, high-resolution, and difficult setting to validate its effectiveness in improving localization accuracy.

Models. We adopt Qwen2.5-VL (Bai et al., 2025) as our test model, including configurations with 3B and 7B parameters. Unlike previous multimodal models, Qwen2.5-VL can directly output absolute coordinates for grounding tasks without requiring additional post-hoc alignment. The Qwen2.5-VL series has demonstrated strong performance on standard coordinate prediction benchmarks, even surpassing some specialized GUI models. However, its performance still degrades considerably in high-resolution scenarios, highlighting the inherent difficulty of precise localization under long-context inputs.

Method configurations. For all evaluated models, we adopt greedy decoding to eliminate randomness and ensure reproducibility. For our proposed method VPSG, we identify the optimal hyperparameter configuration through grid search, with $\alpha = 0.55$ and decay = 0.4.

Compared models We evaluate a broad spectrum of multimodal models with an emphasis on general-purpose and training-free baselines, which are particularly important for assessing the effectiveness of our method without additional fine-tuning. This group includes Qwen-VL-7B (Bai et al., 2023), GPT-4o (Achiam et al., 2023), Qwen2-VL-7B (Wang et al., 2024), and MiniCPM-V, representing strong generalist vision–language models that can directly perform coordinate prediction. We further include the recent Qwen2.5-VL family (3B and 7B), which serves as our primary base model for applying VPSG and can output absolute coordinates without post-hoc alignment. For completeness, we also report results of specialized GUI action models such as SeeClick (Cheng et al., 2024), OS-Atlas-4B/7B (Wu et al., 2024), ShowUI-2B (Lin et al., 2024), CogAgent (Hong et al., 2024), Aria-GUI (Yang et al., 2024), and UGround-7B (Gou et al., 2025), which are trained or instruction-tuned specifically for interface grounding tasks. This diverse set of baselines enables a comprehensive evaluation of VPSG across both generic and domain-specific settings. The experimental results of the above model are cited from (Li et al., 2025).

4.2 Overall performance

As shown in Table 1, VPSG consistently improves both base models on the Screenspot-Pro benchmark when measured by percentage correct. On Qwen2.5-VL-3B, the overall percentage correct increases from 11.6 to 13.3, a gain of 1.7 percentage points. Clear improvements appear in multiple text-

Model	Development		Creative		CAD		Scientific		Office		OS		Avg
	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	
Trained GUI Action Models 🔥													
SeeClick	0.6	0.0	1.0	0.0	2.5	0.0	3.5	0.0	1.1	0.0	2.8	0.0	1.1
OS-Atlas-4B	7.1	0.0	3.0	1.4	2.0	0.0	9.0	5.5	5.1	3.8	5.6	0.0	3.7
ShowUI-2B	16.9	1.4	9.1	0.0	2.5	0.0	13.2	7.3	15.3	7.5	10.3	2.2	7.7
CogAgent-18B	14.9	0.7	9.6	0.0	7.1	3.1	22.2	1.8	13.0	0.0	5.6	0.0	7.7
Aria-GUI	16.2	0.0	23.7	2.1	7.6	1.6	27.1	6.4	20.3	1.9	4.7	0.0	11.3
UGround-7B	26.6	2.1	27.3	2.8	14.2	1.6	31.9	2.7	31.6	11.3	17.8	0.0	16.5
OS-Atlas-7B	33.1	1.4	28.8	2.8	12.2	4.7	37.5	7.3	33.9	5.7	27.1	4.5	18.9
Generalist Models or Training-free Methods ❄️													
Qwen-VL-7B	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.1
GPT-4o	1.3	0.0	1.0	0.0	2.0	0.0	2.1	0.0	1.1	0.0	0.0	0.0	0.8
Qwen2-VL-7B	2.6	0.0	1.5	0.0	0.5	0.0	6.3	0.0	3.4	1.9	0.9	0.0	1.6
MiniCPM-V	7.1	0.0	2.0	0.0	4.1	**1.6**	8.3	0.0	2.8	3.8	3.7	1.1	3.0
Qwen2.5-VL-3B	18.8	1.4	16.7	1.4	8.1	**1.6**	20.8	5.5	24.3	1.9	16.8	3.4	11.6
Qwen2.5-VL-3B + VPSG	24.7	2.1	**20.2**	2.1	**10.2**	**1.6**	21.5	5.5	26.6	5.7	15.9	1.1	13.3
Qwen2.5-VL-7B	37.7	**2.8**	19.7	2.1	7.6	**1.6**	**31.3**	5.5	41.8	11.3	29.9	**10.1**	18.5
Qwen2.5-VL-7B + VPSG	**40.9**	2.1	19.8	**2.8**	8.1	**1.6**	30.6	**5.6**	**43.5**	**13.2**	**29.9**	**10.1**	**19.1**

Table 1: Screen-based grounding results on SCREENSPOT-PRO. Each column reports the evaluation score (higher is better) for a category (*Development*, *Creative*, *CAD*, *Scientific*, *Office*, *OS*) split by UI type (*Text*/*Icon*); *Avg* is the unweighted mean across all columns. Rows marked "+ VPSG" apply our test-time guidance to the same base model, isolating the effect of the method. The best result among generalist models or training-free methods are highlighted in **bold** font.

oriented categories, including Development (Text) from 18.8 to 24.7 (+5.9 points), Creative (Text) from 16.7 to 20.2 (+3.5 points), CAD (Text) from 8.1 to 10.2 (+2.1 points), Office (Text) from 24.3 to 26.6 (+2.3 points)), and Scientific (Text) from 20.8 to 21.5 (+0.7 points). Several icon-oriented settings also benefit; for example, Development (Icon) rises from 1.4 to 2.1 (+0.7 points), Office (Icon) rises from 1.9 to 5.7 (+3.8 points) indicating that mitigating position-induced bias can stabilize landmark selection even when the target is an icon rather than text.

For Qwen2.5-VL-7B, the overall percentage correct increases from 18.5 to 19.1 (+0.6 points). Notable gains include Development (Text) from 37.7 to 40.9 (+3.2 points) and Office (Text) from 41.8 to 43.5 (+1.7 points), along with improvements in icon-oriented cases such as Office (Icon) from 11.3 to 13.2 (+1.9 points). Taken together, these results show that test-time negative-evidence guidance yields reliable lifts across model scales and interaction modes, enhancing both text-oriented and icon-oriented behaviors by suppressing spurious effects that emerge when positional signals are unreliable.

Our results underscore that a causal analysis of error pathways is essential for effectively mitigating coordinate prediction bias. By explicitly contrasting a position-conditioned distribution—obtained from normal positional encodings—with a position-unconditioned reference—derived from shuffled positional encodings—VPSG highlights and strengthens the influence of positional information in the final token distribution, while suppressing the position-agnostic tendencies that drive systematic, directional errors.

This comparison clarifies how positional cues causally affect output coordinates and ensures that guidance is grounded in a measurable contrast rather than heuristic adjustment. Because the intervention operates solely on the final-layer logits at test time, it remains fully compatible with pretrained MLLMs and introduces no additional training cost, architectural changes, or data requirements. Consequently, VPSG serves as a model-agnostic, plug-in method applicable to a broad range of coordinate-prediction tasks, enabling consistent and reproducible improvements across datasets and resolutions without modifying the existing training pipeline.

Qwen2.5-VL-3B		
Setting	**Avg**	**Δ**
VPSG	13.3	–
w/o Seeds aggregation	13.0	↓ 0.3
w/o Coefficient decay	11.9	↓ 1.4
Qwen2.5-VL-7B		
Setting	**Avg**	**Δ**
VPSG	19.1	–
w/o Seeds aggregation	18.6	↓ 0.5
w/o Coefficient decay	18.2	↓ 0.9

Table 2: Ablation study results. The analyzed method components include: (i) *seeds aggregation*, the robust log-space aggregation of multiple PE-shuffled auxiliary routes (*w/o seeds aggregation*: use a single seed, no aggregation); and (ii) *coefficient decay*, the position-aware geometric decay of the digit-only guidance weight with a reset at the first y digit (*w/o coefficient decay*: use a constant guidance weight across digits).

4.3 Ablation study

We perform ablations on Screenspot-Pro (percentage correct) to quantify the contribution of each VPSG component while holding all other settings fixed (same base model and decoding strategy).

4.3.1 seeds aggregation

The first component is seeds aggregation: instead of relying on a single PE-shuffled auxiliary route, the full method aggregates multiple routes in log space (geometric mean). Removing this component (w/o seeds aggregation) leads to a drop in performance. The rationale is that, although the errors induced by missing positional encodings are directional, any single random shuffle yields only a noisy sample from the underlying bias distribution and may not be representative. Aggregating across multiple seeds provides a more faithful estimate of the expectation of this position-unconditioned bias prior. This multi-path aggregation better recovers the output distribution absent positional conditioning.

4.3.2 Coefficient decay

The second key component of VPSG is coefficient decay. Table 2 highlights the importance of this design: removing coefficient decay (*w/o coefficient decay*) reduces the average percentage correct from 13.3 → 11.9 (↓ 1.4) on Qwen2.5-VL-3B and from 19.1 → 18.2 (↓ 0.9) on Qwen2.5-VL-7B. These drops are substantially larger than those caused by removing seeds aggregation, underscoring that position-aware scheduling is a primary driver of VPSG's gains. Beyond its positional-error weighting, coefficient decay also compensates for confidence attenuation along the digit sequence. Empirically, we observe that when the model predicts a multi-digit coordinate such as $[1234, 567]$, the confidence (logit margin between the top token and the runner-up) for the first digit is typically higher than for later digits: This progressive narrowing of logit margins indicates that later tokens are intrinsically more ambiguous, making them more sensitive to over-regularization. Applying a constant guidance coefficient would over-penalize these low-confidence positions, potentially distorting fine-scale digits or even the $[x, y]$ template.

By geometrically decaying α_t and resetting at the start of the y coordinate, VPSG aligns the guidance strength with both positional importance and intrinsic confidence: it strongly constrains the high-order digits that dominate absolute error, while reducing the weight where the model's own uncertainty is higher and logit gaps are small. This targeted scheduling suppresses position-unconditioned biases without compromising the natural fine-grained structure of the output.

Taken together, these analyses confirm that coefficient decay is essential for balancing guidance strength with positional and confidence-based considerations, enabling VPSG to suppress position-unconditioned biases effectively.

4.4 Inference Efficiency

Beyond accuracy, we evaluate the practical inference overhead introduced by VPSG on ScreenSpot-Pro benchmark. Since VPSG runs S auxiliary routes in addition to the original route, the naive theoretical cost would be approximately $(S + 1)\times$. However, as shown in Table 3, the measured overhead in our setup is substantially smaller: with $S = 3$, VPSG increases the per-case latency from 3.64s to 5.27s on Qwen2.5-VL-3B and from 4.20s to 5.50s on Qwen2.5-VL-7B, corresponding to only about $1.3\times$–$1.45\times$ slowdown. These results indicate that VPSG is practically feasible as a plug-in inference-time module for high-resolution GUI grounding pipelines, providing accuracy gains with a modest latency increase.

We attribute the favorable accuracy–latency trade-off to two key design choices. First, VPSG is a lightweight, inference-time control that operates only on the final-layer logits and does not require

Setting	Total Time (s)	Time per Case (s)
Baseline (No VPSG, 3B model)	5755	3.64
Baseline (No VPSG, 7B model)	6640	4.20
VPSG (S=3, 3B model)	8332	5.27
VPSG (S=3, 7B model)	8696	5.50

Table 3: Inference efficiency comparison on ScreenSpot-Pro.

any retraining or architectural modification. Second, VPSG is strictly activated by a finite-state machine (FSM) and only intervenes when decoding digit tokens (i.e., coordinate digits). For the majority of output tokens (e.g., brackets, commas, spaces, and other non-digit text), decoding proceeds exactly as in the baseline without guidance. Together, these design choices keep the additional computation limited, yielding a moderate latency overhead while retaining the benefits of multi-route guidance.

5 Conclusion

We introduced Vision-PE Shuffle Guidance (VPSG) to rectify coordinate drift in multimodal large language models. Our causal analysis reveals that high-resolution inputs trigger directional and non-random biases. VPSG addresses this issue by using shuffled positional encodings as negative evidence to suppress spurious numeric priors. The method incorporates multi-seed aggregation and position-aware coefficient decay to stabilize the output. Evaluation on the ScreenSpot-Pro benchmark confirms consistent performance gains without any fine-tuning or architectural changes. These results highlight the critical role of robust positional encoding in fine-grained spatial reasoning. VPSG provides a practical solution for grounding and coordinate-sensitive tasks in future vision-language systems.

Limitations

This study has several limitations. We evaluated only a small number of models and did not include larger parameter sizes. The current evaluation method does not account for incorrectly formatted or invalid coordinate outputs. Furthermore the approach is restricted to English. Expansion to other languages remains in the early stages of development

Acknowledgments

This work is partially supported by National Key R&D Program of China under Grant No. 2023YFF0725100, by the National Natural Science Foundation of China (NSFC) under Grant No. 62402410, by Guangdong Provincial Project (No. 2023QN10X025), by Guangdong Basic and Applied Basic Research Foundation under Grant No. 2023A1515110131, by Guangzhou Municipal Education Bureau (No. 2024312263), by Nansha District Project (No. 2023ZD022), by HKUST(GZ) Kunpeng&Ascend Center of Cultivation, and by the Fundamental Research Funds for the Central Universities.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. Qwen2.5-vl technical report.

Xi Chen, Shiyang Zhou, Muqi Huang, Jiaxu Feng, Yun Xiong, Kun Zhou, Biao Yang, Yuhui Zhang, Huishuai Bao, Sijia Peng, et al. 2025a. A 2d semantic-aware position encoding for vision transformers. *arXiv preprint arXiv:2505.09466*.

Zhanpeng Chen, Mingxiao Li, Ziyang Chen, Nan Du, Xiaolong Li, and Yuexian Zou. 2025b. Advancing general multimodal capability of vision-language

models with pyramid-descent visual position encoding. *arXiv preprint arXiv:2501.10967*.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. 2024. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixiang Shane Gu, and Izzeddin Gur. 2023. Multimodal web navigation with instruction-finetuned foundation models. *arXiv preprint arXiv:2305.11854*.

Muhan Gao, TaiMing Lu, Kuai Yu, Adam Byerly, and Daniel Khashabi. 2024. Insights into llm long-context failures: when transformers know but don't tell. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7611–7625.

Junqi Ge, Ziyi Chen, Jintao Lin, Jinguo Zhu, Xihui Liu, Jifeng Dai, and Xizhou Zhu. 2024. V2pe: Improving multimodal long-context capability of vision-language models with variable visual position encoding. *arXiv preprint arXiv:2412.09616*.

Google. 2023. An important next step on our ai journey.

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*.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. 2024. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pages 289–305. Springer.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. 2024. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14281–14290.

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long T Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, et al. 2024. Found in the middle: Calibrating positional attention bias improves long context utilization. *arXiv preprint arXiv:2406.16008*.

Ermo Hua, Che Jiang, Xingtai Lv, Kaiyan Zhang, Youbang Sun, Yuchen Fan, Xuekai Zhu, Biqing Qi, Ning Ding, and Bowen Zhou. 2024. Fourier position embedding: Enhancing attention's periodic extension for length generalization. *arXiv preprint arXiv:2412.17739*.

Wonjun Lee, Bumsub Ham, and Suhyun Kim. 2025. Maximizing the position embedding for vision transformers with global average pooling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18154–18162.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. 2023. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36:28541–28564.

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. *arXiv preprint arXiv:2504.07981*.

Wenhui Liao, Jiapeng Wang, Hongliang Li, Chengyu Wang, Jun Huang, and Lianwen Jin. 2025. Doclayllm: An efficient multi-modal extension of large language models for text-rich document understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4038–4049.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Weixian Lei, Lijuan Wang, and Mike Zheng Shou. 2024. Showui: One vision-language-action model for gui visual agent. *arXiv preprint arXiv:2411.17465*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, Lei Zhang, Jianfeng Gao, and Chunyuan Li. 2023b. Llava-plus: Learning to use tools for creating multimodal agents.

Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. 2024. Omniparser for pure vision based gui agent. *arXiv preprint arXiv:2408.00203*.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

MosaicML. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms. Accessed: 2023-05-05.

OpenAI. 2022. OpenAI: Introducing ChatGPT.

Judea Pearl. 2018. Causal and counterfactual inference. *The Handbook of Rationality*, pages 1–41.

Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. 2016. *Causal inference in statistics: A primer*. John Wiley & Sons.

Jianing Qi, Jiawei Liu, Hao Tang, and Zhigang Zhu. 2025. Beyond semantics: Rediscovering spatial awareness in vision-language models. *arXiv preprint arXiv:2503.17349*.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. 2020. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3716–3725.

Xingjian Tao, Yiwei Wang, Yujun Cai, Zhicheng Yang, and Jing Tang. 2025. Understanding gui agent localization biases through logit sharpness. *arXiv preprint arXiv:2506.15425*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Yiwei Wang, Muhao Chen, Wenxuan Zhou, Yujun Cai, Yuxuan Liang, Dayiheng Liu, Baosong Yang, Juncheng Liu, and Bryan Hooi. 2022. Should we rely on entity mentions for relation extraction? debiasing relation extraction with counterfactual analysis. *arXiv preprint arXiv:2205.03784*.

Ziqi Wang, Hanlin Zhang, Xiner Li, Kuan-Hao Huang, Chi Han, Shuiwang Ji, Sham M. Kakade, Hao Peng, and Heng Ji. 2025. Eliminating position bias of language models: A mechanistic approach.

Hang Wu, Hongkai Chen, Yujun Cai, Chang Liu, Qingwen Ye, Ming-Hsuan Yang, and Yiwei Wang. 2025. Dimo-gui: Advancing test-time scaling in gui grounding via modality-aware visual reasoning. *arXiv preprint arXiv:2507.00008*.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. 2024. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*.

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. 2025. Qwen2.5-omni technical report.

Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. 2024. Aria-ui: Visual grounding for gui instructions. *arXiv preprint arXiv:2412.16256*.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1.

Howard Yen, Tianyu Gao, and Danqi Chen. 2024. Long-context language modeling with parallel context encoding.

Hao Yu, Tangyu Jiang, Shuning Jia, Shannan Yan, Shunning Liu, Haolong Qian, Guanghao Li, Shuting Dong, and Chun Yuan. 2025. Comrope: Scalable and robust rotary position embedding parameterized by trainable commuting angle matrices. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4508–4517.

Yi-Fan Zhang, Qingsong Wen, Chaoyou Fu, Xue Wang, Zhang Zhang, Liang Wang, and Rong Jin. 2024. Beyond llava-hd: Diving into high-resolution large multimodal models. *arXiv preprint arXiv:2406.08487*.

A Use of Large Language Models

Large language models (LLMs) were used solely for language polishing and minor editorial assistance (e.g., grammar, wording, and clarity). They were not involved in the conception of research ideas, design of experiments, data analysis, or interpretation of results. All scientific content, methods, and conclusions were developed independently by the authors.

B Proof of Proposition (VPSG token guidance)

Setup. Let $\mathcal{V}$ be the vocabulary and $\mathcal{D} \subset \mathcal{V}$ the digit subset. At decoding step t, denote by $p_A(v \mid c_t)$ the position-conditioned (main-route) distribution and by $p_B(v)$ the position-*un*conditioned reference obtained from PE-shuffled auxiliary routes. Write $\ell_A(v \mid c_t) = \log p_A(v \mid c_t)$ and $\tilde{\ell}_B(v) = \log p_B(v)$. We assume: (i) the same logits processors are applied to both routes before any guidance; (ii) guidance acts only at the final-layer logits; and (iii) decoding is greedy, i.e., depends on the $\arg\max$ over logits.

CFG form. Consider the (digit-only) classifier-free guidance (CFG) mixing:

$$p_{\mathrm{CFG}}(v \mid c_t) \propto \begin{cases} \dfrac{p_A(v \mid c_t)^{1+\lambda_t}}{p_B(v)^{\lambda_t}}, & v \in \mathcal{D}, \\ p_A(v \mid c_t), & v \notin \mathcal{D}, \end{cases} \tag{5}$$

with normalization constant $Z_{\mathrm{CFG}}(c_t)$ implicit over $v \in \mathcal{V}$. Taking logs for $v \in \mathcal{D}$,

$$\begin{aligned} &\log p_{\mathrm{CFG}}(v \mid c_t) \\ &= (1+\lambda_t)\,\ell_A(v \mid c_t) - \lambda_t\,\tilde{\ell}_B(v) - \log Z_{\mathrm{CFG}}(c_t). \end{aligned} \tag{6}$$

Digit-only positive affine rescaling. Define, for $v \in \mathcal{D}$,

$$\begin{aligned} \psi_t(v) &= \frac{1}{1+\lambda_t}\Big(\log p_{\mathrm{CFG}}(v \mid c_t) + \log Z_{\mathrm{CFG}}(c_t)\Big) \\ &= \ell_A(v \mid c_t) - \underbrace{\frac{\lambda_t}{1+\lambda_t}}_{\alpha_t}\,\tilde{\ell}_B(v). \end{aligned} \tag{7}$$

For $v \notin \mathcal{D}$, keep $\psi_t(v) = \log p_A(v \mid c_t)$ (no change).

Lemma 1 (Argmax invariance under positive affine transforms)**.** *Let $S \subseteq \mathcal{V}$ and $a > 0$, $b \in \mathbb{R}$. For any scores $\{u(v)\}_{v \in S}$, $\arg\max_{v \in S} u(v) = \arg\max_{v \in S}\{a\,u(v) + b\}$.*

Proof. For $a > 0$, $u(v_1) \geq u(v_2) \iff a\,u(v_1) + b \geq a\,u(v_2) + b$.

Applying the lemma to (6) with $a = \frac{1}{1+\lambda_t}$ and $b = \frac{\log Z_{\mathrm{CFG}}(c_t)}{1+\lambda_t}$ *restricted to* $v \in \mathcal{D}$ shows that

$$\begin{aligned} &\arg\max_{v \in \mathcal{D}} \log p_{\mathrm{CFG}}(v \mid c_t) = \arg\max_{v \in \mathcal{D}} \psi_t(v) \\ &\quad = \arg\max_{v \in \mathcal{D}} \big(\ell_A(v \mid c_t) - \alpha_t\,\tilde{\ell}_B(v)\big). \end{aligned} \tag{8}$$

Since $p_{\mathrm{CFG}}(v \mid c_t) = p_A(v \mid c_t)$ for $v \notin \mathcal{D}$ by (5), the combined $\arg\max$ under CFG equals that under the piecewise score

$$s(v) = \begin{cases} \ell_A(v \mid c_t) - \alpha_t \tilde{\ell}_B(v), & v \in \mathcal{D}, \\ \ell_A(v \mid c_t), & v \notin \mathcal{D}, \end{cases} \tag{9}$$

which is the VPSG "negative-evidence" scoring form.

Moreover, (7) yields the parameter mapping:

$$\boxed{\alpha_t = \frac{\lambda_t}{1+\lambda_t}} \iff \boxed{\lambda_t = \frac{\alpha_t}{1-\alpha_t}}. \tag{10}$$

□

Equivalence under greedy decoding. Greedy decoding selects $\hat{v}_t = \arg\max_{v \in \mathcal{V}} \log p_{\mathrm{CFG}}(v \mid c_t)$. By the lemma and the piecewise definition above, the same $\hat{v}_t$ is obtained by maximizing $s(v)$, because: (i) on digits we used a positive affine transform of $\log p_{\mathrm{CFG}}$; (ii) on non-digits the two forms coincide; and (iii) both are compared in the same joint candidate set $\mathcal{V}$. Therefore CFG (5) and VPSG scoring $s(v)$ are *decision-equivalent* under greedy decoding.

Normalization and distributional form. If a normalized distribution is desired, define

$$p_{\mathrm{VPSG}}(v \mid c_t) = \frac{\exp(s(v))}{\sum_{u \in \mathcal{V}} \exp(s(u))}. \tag{11}$$

This coincides with (5) up to the digit-only affine rescaling leading to the same $\arg\max$, and yields the proposition's statement:

$$\begin{aligned} &p_{\mathrm{VPSG}}(v \mid c_t) \propto \\ &\begin{cases} \exp\big(\ell_A(v \mid c_t) - \alpha_t \tilde{\ell}_B(v)\big), & v \in \mathcal{D}, \\ p_A(v \mid c_t), & v \notin \mathcal{D}. \end{cases} \end{aligned} \tag{12}$$

Remarks on assumptions. (1) *Same logits processors:* ensures that when $\alpha_t \to 0$ (or $\lambda_t \to 0$) the VPSG rule recovers the baseline exactly. (2) *Final-layer intervention:* guarantees that the affine

Group	Text	Icon	Total
CAD	197	64	261
Creative	198	143	341
Dev	154	145	299
OS	107	89	196
Office	177	53	230
Scientific	144	110	254
All	**977**	**604**	**1581**

Table 4: Category and UI-type counts in Screenspot-Pro.

transformation does not change any upstream normalization. (3) *Greedy decoding:* makes decision-equivalence depend only on $\arg\max$; for sampling or beam search, the same mapping holds at the score level, but selection statistics may also depend on temperature/length penalties (which can still be shared across routes).

C Details about dataset ScreenSpot-Pro

We show categories and UI-type counts in Screenspot-Pro in Table 4.

D Algorithm of VPSG

The complete algorithm is shown in Algorithm 1.

E Expected distance in the unit square

Claim 1. *If $X = (X_1, X_2)$ and $Y = (Y_1, Y_2)$ are independent and uniformly distributed on $[0,1]^2$, then*

$$\mathbb{E}\big[\|X - Y\|_2\big] = \frac{2 + \sqrt{2} + 5\ln(1+\sqrt{2})}{15} \approx 0.5214. \tag{13}$$

Proof. Let $U = |X_1 - Y_1|$ and $V = |X_2 - Y_2|$. For $u, v \in [0,1]$, the joint density of (U,V) is $f_{U,V}(u,v) = 4(1-u)(1-v)$, as U, V are i.i.d. triangular variables with $f_U(u) = 2(1-u)$. The Euclidean distance is $R = \sqrt{U^2 + V^2}$. Hence:

$$\mathbb{E}[R] = \int_0^1 \int_0^1 4\sqrt{u^2+v^2}(1-u)(1-v)\,du\,dv. \tag{14}$$

Switching to polar coordinates ($u = r\cos\theta, v = r\sin\theta$), the boundary $0 \le r \le r_{\max}(\theta)$ where $r_{\max}(\theta) = \min\{\sec\theta, \csc\theta\}$ gives:

$$\begin{aligned}\mathbb{E}[R] =& 4\int_0^{\pi/2}\int_0^{r_{\max}(\theta)} r^2(1 - r\cos\theta) \\ &\times (1 - r\sin\theta)\,dr\,d\theta.\end{aligned} \tag{15}$$

Algorithm 1 Vision-PE Shuffle Guidance (VPSG)

Require: Image I, prompt q, base model $\mathcal{M}$, seeds $\{s_1, \ldots, s_S\}$, base coefficient α, decay factor $0 < \text{decay} < 1$

Ensure: Coordinate prediction $\hat{y} = [x, y]$

Main route (position-conditioned):
Run $\mathcal{M}$ on (I, q) with normal positional encodings (PEs) to obtain token distribution $p_A(v \mid c_t)$.
Auxiliary routes (position-unconditioned):
for each seed s **do**
    Shuffle PEs and run $\mathcal{M}$ to get $p_B^{(s)}(v)$.
end for
Aggregate in log-space:

$$\tilde{\ell}_B(v) \leftarrow \frac{1}{S}\sum_{s=1}^{S} \log p_B^{(s)}(v).$$

FSM state tracking:
Use a finite-state machine aligned to the $[x, y]$ template to determine whether the model is decoding a digit in x or y.
Coefficient scheduling:
if decoding the k_x-th digit of x **then**
    $\alpha_t \leftarrow \alpha \cdot \text{decay}^{\,k_x - 1}$
else if decoding the first digit of y **then**
    $\alpha_t \leftarrow \alpha$
else if decoding the k_y-th digit of y **then**
    $\alpha_t \leftarrow \alpha \cdot \text{decay}^{\,k_y - 1}$
end if
Negative-evidence scoring:
for each token v **do**

$$s(v) \leftarrow \begin{cases} \log p_A(v \mid c_t) - \alpha_t \tilde{\ell}_B(v), & v \in \mathcal{D}, \\ \log p_A(v \mid c_t), & v \notin \mathcal{D}. \end{cases}$$

end for
Token selection:
Choose $\hat{v}_t \leftarrow \arg\max_v s(v)$, append to output, and advance FSM.
Termination:
Repeat Steps 3–6 until EOS. Decode tokens into coordinates $\hat{y}$.

Splitting the integral at $\theta = \pi/4$ where the boundary constraint changes:

$$\begin{aligned}\mathbb{E}[R] =& 4\int_0^{\pi/4}\int_0^{\sec\theta} g(r,\theta)\,dr\,d\theta \\ &+ 4\int_{\pi/4}^{\pi/2}\int_0^{\csc\theta} g(r,\theta)\,dr\,d\theta,\end{aligned} \tag{16}$$

where $g(r,\theta) = r^2 - r^3(\cos\theta + \sin\theta) + r^4 \sin\theta\cos\theta$. Integrating in r for a general upper limit a:

$$\int_0^a g(r,\theta)dr = \frac{a^3}{3} - \frac{\cos\theta + \sin\theta}{4}a^4 + \frac{\sin\theta\cos\theta}{5}a^5. \tag{17}$$

Substituting $a = \sec\theta$ for the first range $[0, \pi/4]$ yields:

$$\begin{aligned} J_1(\theta) = &\frac{\sec^3\theta}{3} \\ &- \frac{\cos\theta + \sin\theta}{4}\sec^4\theta \\ &+ \frac{\sin\theta\cos\theta}{5}\sec^5\theta \\ &= \sec^3\theta\left(\frac{1}{12} - \frac{1}{20}\tan\theta\right). \end{aligned} \tag{18}$$

Symmetrically, for $a = \csc\theta$ on $[\pi/4, \pi/2]$, we have $J_2(\theta) = \csc^3\theta(\frac{1}{12} - \frac{1}{20}\cot\theta)$. Thus:

$$\begin{aligned} \mathbb{E}[R] = &4\int_0^{\pi/4}\left(\frac{1}{12}\sec^3\theta - \frac{1}{20}\sec^3\theta\tan\theta\right)d\theta \\ &+ 4\int_{\pi/4}^{\pi/2}\left(\frac{1}{12}\csc^3\theta - \frac{1}{20}\csc^3\theta\cot\theta\right)d\theta. \end{aligned} \tag{19}$$

Using the antiderivatives $\int \sec^3\theta\, d\theta = \frac{1}{2}(\sec\theta\tan\theta + \ln(\sec\theta + \tan\theta))$ and $\int \sec^3\theta\tan\theta\, d\theta = \frac{1}{3}\sec^3\theta$, and noting the symmetry between the two θ-ranges, the integral evaluates to:

$$\begin{aligned} \mathbb{E}[R] &= \frac{1}{3}(\sqrt{2} + \ln(1+\sqrt{2})) - \frac{2}{15}(2\sqrt{2} - 1) \\ &= \frac{2 + \sqrt{2} + 5\ln(1+\sqrt{2})}{15}. \end{aligned} \tag{20}$$

This completes the proof. □

Remark (Application to images) The constant $\mu_\square$ is the dispersion benchmark for a unit square. For arbitrary image sizes (W, H), one can: (i) anisotropically rescale coordinates to $[0,1]^2$ before computing distances, or (ii) form a per-image Monte-Carlo null by sampling i.i.d. uniform points from $[0, W] \times [0, H]$ to estimate the baseline for that specific aspect ratio.

F More Experiment Result

F.1 distance analysis

More details of distance analysis results are shown in Table 5

Image size	Qwen2.5-VL-3B		Qwen2.5-VL-7B	
	Normal	Shuffled	Normal	Shuffled
1920×1080	0.123	0.202	0.211	0.128
2160×1440	0.391	0.146	0.437	0.159
2560×1440	0.392	0.171	0.436	0.180
2560×1600	0.357	0.072	0.348	0.048
2560×1664	0.428	0.139	0.471	0.211
2880×1800	0.379	0.119	0.462	0.111
2992×1870	0.463	0.175	0.347	0.061
3456×2160	0.471	0.122	0.348	0.081
3456×2234	0.457	0.169	0.466	0.106
3840×1080	0.298	0.227	0.287	0.140
3840×2160	0.468	0.117	0.504	0.146
5120×1440	0.435	0.133	0.404	0.058
5120×2880	0.399	0.168	0.408	0.052
6016×3384	0.341	0.117	0.335	0.057
Overall mean	**0.402**	**0.164**	**0.435**	**0.162**

Table 5: Comparison of mean pairwise Euclidean distances (diagonal-normalized) across image resolutions on SCREENSPOT-PRO. Normal PEs exhibit significantly larger within-input distances, while shuffled PEs lead to much tighter clustering and systematic collapse of coordinate predictions, demonstrating the impact of positional information on spatial diversity.

F.2 Case study with per-step logits.

To better illustrate how VPSG corrects coordinate prediction bias, we analyze a single case from SCREENSPOT-PRO (`ppt_windows/screenshot_2024-10-27_21-07-29.png`) Figure 6. The ground-truth bounding box center is $[659, 857]$. The base model without guidance predicts $[1024, 856]$, while VPSG successfully outputs the correct $[659, 857]$. Table 6 lists the top-10 logits probabilities at each decoding step. At the earliest x-digit steps, the uncorrected model shows a strong bias toward larger numbers (e.g., tokens "1" and "0" dominate), reflecting spurious numeric priors induced by missing or unreliable positional encodings. VPSG integrates negative evidence from multiple shuffled PE runs and systematically down-weights these spurious peaks, allowing the true digit sequence to emerge and stabilizing the final $[x, y]$ prediction.

F.3 Ablation: removing seeds aggregation.

To evaluate the contribution of seeds aggregation in VPSG, we remove the multi-seed log-space aggregation and instead rely on a single randomly shuffled positional encoding as the auxiliary route. Table 7 reports the detailed category- and type-level accuracies (percentage of correct predictions) for both Qwen2.5-VL-3B and Qwen2.5-VL-7B models. Without aggregation, accuracy drops across almost all groups and UI-types, confirming that a single random shuffle provides only a noisy sample of the underlying position-unconditioned bias

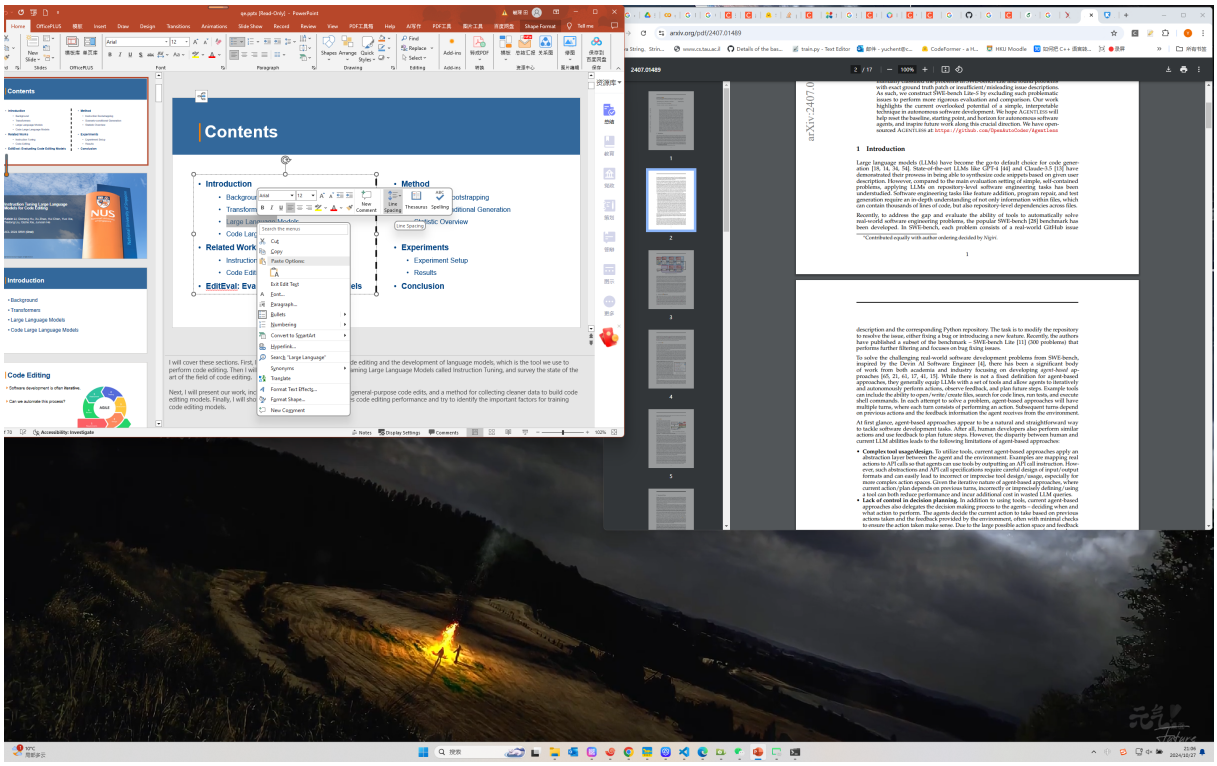

Figure 6: An image case from dataset ScreenSpot-Pro (ppt_windows/screenshot_2024-10-27_21-07-29.png)

Step	Token	VPSG Prob.	Base Prob.
1	[	0.805	0.805
2	6	0.243	0.210
3	5	0.172	0.264 (0 highest)
4	9	0.221	0.140 (2 highest)
5	,	0.999	0.987
6	space	0.999	0.976
7	8	0.740	0.355
8	5	0.589	0.642 (but 6/3 confused)
9	7	0.272	0.421 (6 highest)
10	]	0.999	0.999
11	<eos>	0.993	0.996

Table 6: Per-step top-1 logits probabilities for each decoded token in a representative example. The base model without guidance drifts toward spurious large-x digits (e.g., "1", "0" early), yielding an incorrect coordinate $[1024, 856]$. VPSG, by integrating negative evidence from multiple shuffled PE runs, suppresses these biased peaks and converges to the correct coordinate $[659, 857]$. This example highlights how VPSG stabilizes numeric decoding and restores faithful spatial grounding.

distribution and cannot capture its full expectation. This validates the theoretical claim that multi-seed aggregation approximates the expected bias prior and yields more stable and accurate guidance.

F.4 More details about bias analysis

Table 8 reports the ten most frequent individual numbers across all $[x, y]$ coordinate predictions.

Category / UI type	3B		7B	
	Text (%)	Icon (%)	Text (%)	Icon (%)
CAD	10.66	1.56	6.60	1.56
Creative	18.69	2.10	19.70	2.80
Dev	22.73	2.07	41.56	3.45
OS	15.89	2.25	28.97	12.36
Office	29.38	5.66	41.81	13.21
Scientific	20.14	2.73	27.78	4.55
Overall	19.55	2.48	26.71	5.46

Table 7: Accuracy (%) of VPSG without seeds aggregation across categories and UI types. Both model sizes show clear drops compared with the full VPSG (see main text Table 2): for instance, the 3B model drops from 13.3% overall to 13.0%, and the 7B model from 19.1% to 18.6%. Losses are consistent across text and icon settings, supporting the view that multi-seed aggregation provides a faithful estimate of the expected position-unconditioned bias and stabilizes the guidance effect.

Normal PE = standard positional encodings; shuffled PE = visual positional encodings randomly shuffled at inference time. All counts reflect total occurrences of a number as either the x or y component of predicted coordinates.

Qwen2.5-VL-7B (normal PE)			Qwen2.5-VL-3B (normal PE)			Qwen2.5-VL-3B (shuffled PE)			Qwen2.5-VL-7B (shuffled PE)		
Rank	**Number**	**Freq**	**Rank**	**Number**	**Freq**	**Rank**	**Number**	**Freq**	**Rank**	**Number**	**Freq**
1	1024	296	1	1024	397	1	1024	591	1	1024	902
2	105	54	2	1056	82	2	1056	426	2	1234	582
3	10	35	3	356	73	3	568	184	3	567	580
4	2048	26	4	35	44	4	1000	182	4	672	270
5	2058	26	5	36	39	5	238	159	5	368	266
6	2016	25	6	10	36	6	512	141	6	384	162
7	1940	23	7	1234	28	7	560	141	7	200	41
8	100	23	8	105	27	8	1052	128	8	36	33
9	200	21	9	102	25	9	200	121	9	38	27
10	1056	20	10	1048	23	10	248	119	10	896	26

Table 8: Top-10 most frequent numbers appearing in prediction results for Qwen2.5-VL-3B and Qwen2.5-VL-7B under normal and shuffled positional encodings.