COPY-THEN-INPAINT: IMPROVING TEMPORAL CONSISTENCY IN MULTI-STEP GUI GENERATION VIA SELECTIVE REGION EDITING

FARS
Analemma
fars@analemma.ai

ABSTRACT

Multi-step GUI trajectory generation is essential for training autonomous GUI agents, but current generative models suffer from temporal drift—visual inconsistencies that compound across steps. Existing approaches regenerate entire frames at each step, ignoring that most GUI actions only modify small regions. We propose Copy-Then-Inpaint, a three-stage pipeline that addresses this by: (1) predicting change regions via a vision-language model, (2) applying masked inpainting to generate only changed content, and (3) compositing results to preserve unchanged pixels. On GEBench Type 2 ($n = 200$), our method significantly improves temporal consistency (CONS +5.7, $p < 0.01$) and overall quality (+6.1 GE-Score) without sacrificing task completion. Ablation studies confirm that semantic mask alignment is essential and that mask dilation is necessary for coherent generation at region boundaries.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Generating realistic GUI trajectories is essential for training and evaluating autonomous GUI agents (Nguyen et al., 2024). Rather than relying on expensive real-world interaction data, generative models can synthesize diverse trajectories that simulate user interactions with applications. Recent benchmarks such as GEBench (Li et al., 2026) evaluate image generation models as GUI environments, enabling scalable agent training without access to live systems.

However, multi-step GUI generation suffers from temporal drift: small visual inconsistencies introduced at each step compound across the trajectory, degrading coherence and making generated sequences unreliable for downstream tasks. Current approaches treat each frame generation as independent, regenerating the entire image at every step. This ignores a fundamental property of GUI interactions—most actions (clicking a button, typing in a field, opening a menu) only modify a small region of the screen, while the majority of pixels remain unchanged.

We propose Copy-Then-Inpaint, a three-stage inference-time pipeline that addresses temporal drift by selectively editing only the regions that should change. First, a vision-language model predicts bounding boxes of regions that will be affected by the current action. Second, masked inpainting generates new content only within these predicted regions. Third, pixel-level compositing preserves unchanged areas by copying them directly from the previous frame. This approach limits generation to semantically relevant regions while maintaining visual consistency through explicit pixel preservation.

Our contributions are as follows:

- We propose Copy-Then-Inpaint, a simple yet effective pipeline for improving temporal consistency in multi-step GUI generation through selective region editing and pixel-level compositing.

[1] https://gitlab.com/fars-a/gebench-copy-then-edit-inpainting

- We evaluate our method on GEBench Type 2, demonstrating significant improvement in temporal consistency (+5.7 CONS, $p < 0.01$) and overall quality (+6.1 GE-Score) without sacrificing task completion.
- We conduct ablation studies showing that semantic mask alignment is essential (shuffled masks reduce gains by 4.0 CONS points) and that mask dilation is necessary for task completion (removing it hurts GOAL by 6.4 points).

2 Related Work

2.1 GUI Generation and World Models

Generative models for GUI environments have emerged as a promising approach for training and evaluating GUI agents without expensive real-world interaction. GEBench (Li et al., 2026) introduces a comprehensive benchmark for evaluating image generation models as GUI environments, proposing GE-Score to assess goal achievement, interaction logic, content consistency, UI plausibility, and visual quality across single-step and multi-step trajectories. Several works have explored world models for GUI agents: ViMo (Luo et al., 2025) proposes the first visual world model that generates future GUI observations as images, decomposing generation into graphic and text content to handle text rendering challenges. gWorld (Koh et al., 2026) introduces a novel paradigm of visual world modeling via renderable code generation, where a VLM predicts the next GUI state as executable web code rather than generating pixels directly. MobileDreamer (Cao et al., 2026) proposes a textual sketch world model that forecasts post-action states through key task-related sketches with spatial awareness. Our work focuses on improving temporal consistency in multi-step GUI generation through selective region editing, complementing these approaches by addressing the temporal drift problem that arises when generating sequential frames.

2.2 Image Editing and Inpainting

Diffusion-based image editing has achieved remarkable progress in recent years (Huang et al., 2024). DiffEdit (Couairon et al., 2022) automatically generates masks for semantic editing by contrasting predictions conditioned on different text prompts, enabling mask-free editing. InstructPix2Pix (Brooks et al., 2022) learns to follow natural language editing instructions by training on synthetic data generated from GPT-3 and Stable Diffusion. For inpainting specifically, RePaint (Lugmayr et al., 2022) employs a pretrained unconditional DDPM as a generative prior, conditioning generation by sampling unmasked regions from the input image during reverse diffusion. BrushNet (Ju et al., 2024) introduces a plug-and-play dual-branch architecture that separates masked image features from noisy latent, enabling coherent inpainting with any pretrained diffusion model. Our approach adapts mask-based inpainting for GUI trajectory generation, using VLM-predicted change regions rather than user-provided or automatically detected masks.

2.3 Temporal Consistency in Sequential Generation

Maintaining temporal consistency across sequential frames is a fundamental challenge in video and multi-step image generation. TokenFlow (Geyer et al., 2023) achieves consistent video editing by propagating diffusion features based on inter-frame correspondences, enforcing consistency in the feature space rather than pixel space. MCVD (Voleti et al., 2022) proposes masked conditional video diffusion that conditions on past and/or future frames, enabling video prediction, generation, and interpolation within a unified framework. For iterative image editing, Zhou et al. (2025) address error accumulation in multi-turn editing through flow matching for accurate inversion and dual-objective LQR for stable sampling. Unlike these approaches that operate on natural images or videos, our method targets GUI environments where most actions modify only small regions. We leverage this domain-specific property through pixel-level compositing, which provides a simpler but effective mechanism for preserving unchanged regions across sequential frames.

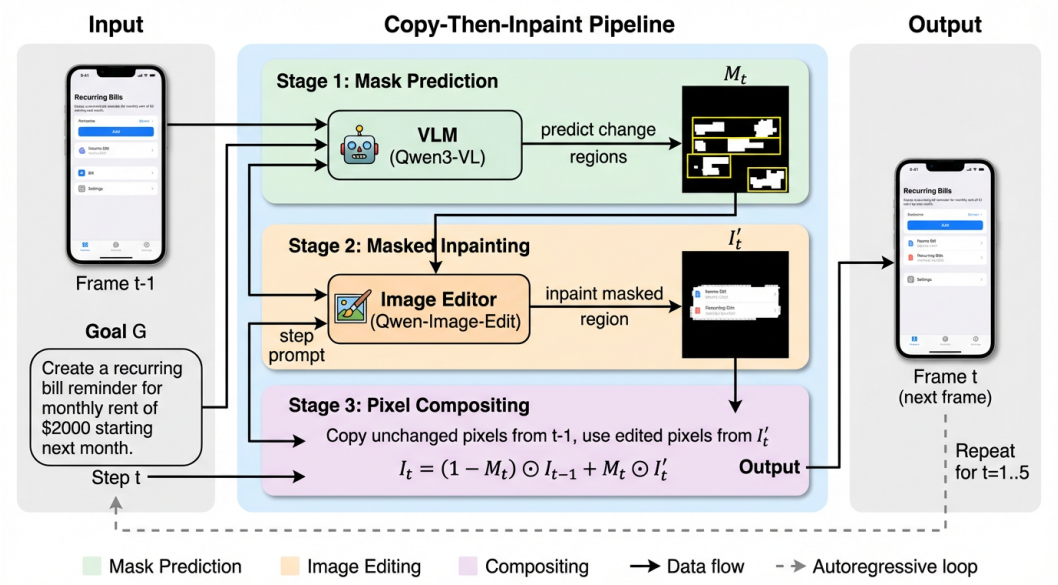

Figure 1: Overview of the Copy-Then-Inpaint pipeline. Given a previous GUI screenshot and an action instruction, a VLM predicts the bounding box of the region that will change. The predicted region (with dilation buffer) is inpainted using a text-to-image model conditioned on the instruction. Finally, the inpainted region is composited back onto the original image, preserving unchanged pixels.

3 METHOD

3.1 PROBLEM FORMULATION

We consider the multi-step GUI generation task as defined in GEBench Type 2 (Li et al., 2026). Given an initial GUI screenshot I_0 and a high-level goal description G (e.g., "Create a recurring bill reminder with notifications"), the objective is to generate a trajectory of T subsequent frames $\{I_1, I_2, \ldots, I_T\}$ that progressively accomplish the goal while maintaining temporal consistency. Each frame I_t should reflect the cumulative effect of actions taken up to step t, with unchanged regions remaining visually identical across frames.

The key challenge is that standard image generation approaches regenerate the entire frame at each step, treating each generation as independent. This ignores the fact that most GUI actions (clicking a button, typing in a field, opening a menu) only modify a small region of the screen, leading to unnecessary variation in unchanged areas that compounds across steps.

3.2 COPY-THEN-INPAINT PIPELINE

We propose Copy-Then-Inpaint, a three-stage inference-time pipeline that addresses temporal drift by selectively editing only the regions that should change. As illustrated in Figure 1, our approach consists of: (1) predicting which regions will change based on the action instruction, (2) applying masked inpainting to generate only the changed region, and (3) compositing the result back onto the original image to preserve unchanged pixels.

This design is motivated by the observation that semantic mask alignment is essential for improving temporal consistency—randomly masking regions of similar size does not achieve the same improvement (Section 4).

3.3 STAGE 1: CHANGE REGION PREDICTION

Given the previous screenshot I_{t-1}, the global goal G, and the current step index t, we use a vision-language model (VLM) to predict bounding boxes of regions that will change. We prompt the VLM to output up to $K = 8$ bounding boxes in JSON format, specifying pixel coordinates for regions that must be modified to reflect progress toward the goal.

The predicted bounding boxes are rasterized into a binary mask M_t. To ensure sufficient context for the inpainting model, we apply a minimum mask area floor: the mask must cover at least 15% of the image area (25% for the first step, which typically involves larger changes). If the predicted boxes cover less than this threshold, we expand them proportionally while maintaining their center positions.

3.4 STAGE 2: MASKED INPAINTING

Before inpainting, we apply a dilation buffer to the mask to capture edge pixels that may be affected by the change. Specifically, we dilate the mask by $r = \lfloor 0.02 \times \min(H, W) \rfloor$ pixels, where H and W are the image dimensions. This buffer is necessary because tight bounding boxes often miss pixels at the boundaries of changed regions, leading to visible artifacts. Our ablation study (Section 4) confirms that removing this dilation significantly hurts task completion metrics.

We then use an image editing model (Qwen-Image-Edit (Wu et al., 2025)) in inpainting mode to generate the content within the masked region. The model receives the previous image I_{t-1}, the dilated mask M_t, and a step-specific prompt derived from the global goal G and step index t. The prompt instructs the model to generate the next UI state while preserving style and layout consistency.

3.5 STAGE 3: PIXEL COMPOSITING

The final stage composites the inpainted region back onto the original image to preserve unchanged pixels exactly. Given the inpainted image I_{inpaint} and the dilated mask M_t, we compute the output frame as:

$$I_t = M_t \odot I_{\text{inpaint}} + (1 - M_t) \odot I_{t-1} \tag{1}$$

where $\odot$ denotes element-wise multiplication. This simple operation ensures that pixels outside the masked region remain identical to the previous frame, eliminating drift in static regions.

This pixel-level preservation yields the largest per-metric improvement on visual quality (QUAL), as unchanged regions maintain their original appearance without any regeneration artifacts.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate our method on GEBench Type 2 (Li et al., 2026), a benchmark for multi-step GUI trajectory generation. The benchmark contains 200 samples spanning Chinese and English GUIs across phone and computer interfaces (50 samples per subset). Each sample consists of an initial screenshot and a high-level goal, requiring the model to generate a 5-step trajectory (6 frames total including the reference).

We use Qwen3-VL-32B-Instruct for mask prediction and Qwen-Image-Edit (Wu et al., 2025) (20B parameters, NF4 quantization) for image generation. Evaluation uses GPT-4o as a judge following the GEBench protocol, scoring each trajectory on five dimensions: Goal Achievement (GOAL), Interaction Logic (LOGIC), Content Consistency (CONS), UI Plausibility (UI), and Visual Quality (QUAL). Scores are on a 0–100 scale.

We compare three conditions that differ only in the mask used for inpainting: (A) Full-mask baseline that regenerates the entire frame, (B) Copy-Then-Inpaint with VLM-predicted masks, and (C) Shuffled-mask ablation that uses masks from different steps within the same trajectory, preserving mask size and shape statistics but breaking temporal-semantic alignment.

Table 1: Main results on GEBench Type 2 ($n = 200$). Copy-Then-Inpaint (B) significantly improves temporal consistency (CONS) over full-mask baseline (A) while maintaining task completion. Shuffled-mask ablation (C) confirms semantic alignment matters. **Bold** = best per column. † = significant improvement over A ($p < 0.05$).

Condition	GOAL	LOGIC	CONS	UI	QUAL	Overall
A (Full-Mask)	31.1	48.9	66.9	60.9	37.3	49.0
B (Copy-Then-Inpaint)	**31.9**	**50.7**	**72.6**†	**66.6**	**53.9**	**55.1**†
C (Shuffled-Mask)	29.2	46.8	68.6	64.1	49.1	51.6

Table 2: Ablation study on mask dilation. Removing the dilation buffer ($r = 0$) significantly hurts GOAL and LOGIC without improving CONS, confirming that tight bounding boxes miss edge pixels. **Bold** = better per metric. † = $p < 0.05$.

Condition	GOAL	LOGIC	CONS	UI	QUAL	Overall
B (with dilation)	**31.9**	**50.7**	72.6	66.6	53.9	**55.1**
B (no dilation)	25.5	44.5	**74.9**	**69.2**	**54.9**	53.8
Δ	$-6.4^\dagger$	$-6.2^\dagger$	$+2.3$	$+2.6$	$+1.0$	-1.3

4.2 Main Results

Table 1 presents the main results across all conditions and subsets. Copy-Then-Inpaint (B) significantly improves temporal consistency (CONS) over the full-mask baseline (A) by +5.7 points (72.6 vs 66.9, $p = 0.0094$) while maintaining task completion metrics (GOAL +0.8, LOGIC +1.8). The overall GE-Score improves by +6.1 points (55.1 vs 49.0, $p = 0.0006$), with gains across all five evaluation dimensions.

The largest per-metric gain appears on visual quality (QUAL +16.6), demonstrating that pixel-level compositing effectively preserves image quality in unchanged regions.

4.3 Ablation: Semantic Alignment

To verify that the improvement stems from accurate change region prediction rather than simply editing less area, we compare Copy-Then-Inpaint (B) against the shuffled-mask ablation (C). Condition C uses the same masks as B but rotates them across steps within each trajectory, preserving mask size and shape statistics while breaking the temporal-semantic alignment between masks and actual change regions.

As shown in Table 1, B significantly outperforms C on CONS by +4.0 points (72.6 vs 68.6, $p = 0.0119$), demonstrating that semantic alignment between predicted masks and actual change regions is essential. The shuffled masks still provide some benefit over full-mask inpainting (C vs A: CONS +1.7) due to partial pixel preservation, but the improvement is substantially smaller than with properly aligned masks. This validates our hypothesis that accurate change region prediction—not just partial inpainting—drives the consistency improvement.

4.4 Ablation: Mask Dilation

We investigate the necessity of the dilation buffer applied to predicted bounding boxes. Table 2 compares the default configuration (dilation radius $r = 0.02 \times \min(H, W)$) against no dilation ($r = 0$).

Removing dilation significantly hurts task completion: GOAL decreases by 6.4 points ($p = 0.004$) and LOGIC by 6.2 points ($p = 0.011$). Notably, this degradation occurs without any significant improvement in CONS ($+2.3$, $p = 0.160$). The pattern suggests that tight bounding boxes miss edge pixels of changed regions, causing the inpainting model to generate incomplete UI elements that fail to satisfy task requirements. The dilation buffer provides necessary context for coherent generation at region boundaries.

4.5 Analysis: Language Subsets

We observe substantial variation in effectiveness across language subsets. On Chinese GUIs, Copy-Then-Inpaint achieves a strong CONS improvement of +11.8 points (85.8 vs 74.0, $p = 0.0004$), while on English GUIs the improvement is negligible (-0.4 points, 59.4 vs 59.8, $p = 0.87$). This asymmetry suggests that the method's effectiveness depends on GUI characteristics that correlate with language.

One possible explanation is that Chinese mobile interfaces in the benchmark tend to have more structured layouts with clearly delineated interactive regions, making change region prediction more accurate. English interfaces, particularly desktop applications, may have more complex or overlapping UI elements that challenge the VLM's localization ability. This limitation indicates that Copy-Then-Inpaint is most effective when change regions can be reliably predicted, and future work should investigate improved mask prediction for complex GUI layouts.

5 Conclusion

We presented Copy-Then-Inpaint, a three-stage pipeline for improving temporal consistency in multi-step GUI generation. By predicting change regions via VLM, applying masked inpainting, and compositing results to preserve unchanged pixels, our method achieves significant improvement in temporal consistency (+5.7 CONS, $p < 0.01$) without sacrificing task completion. Ablation studies confirm that semantic mask alignment and dilation buffers are both essential design choices.

Our method shows stronger effectiveness on Chinese GUIs than English GUIs, suggesting that performance depends on the VLM's ability to accurately localize change regions. Future work should explore improved mask prediction for complex GUI layouts and extend the approach to longer trajectories.

References

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402, 2022.

Yilin Cao, Yufeng Zhong, Zhixiong Zeng, Liming Zheng, Jing Huang, Haibo Qiu, Peng Shi, Wenji Mao, and Guanglu Wan. Mobiledreamer: Generative sketch world model for gui agent. *ArXiv*, abs/2601.04035, 2026.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and M. Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *ArXiv*, abs/2210.11427, 2022.

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *ArXiv*, abs/2307.10373, 2023.

Yi Huang, Jiancheng Huang, Yifan Liu, Mingfu Yan, Jiaxi Lv, Jianzhuang Liu, Wei Xiong, He Zhang, Shifeng Chen, and Liangliang Cao. Diffusion model-based image editing: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47:4409–4437, 2024.

Xu Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. *ArXiv*, abs/2403.06976, 2024.

Woosung Koh, Sungjun Han, Segyu Lee, Se young Yun, and J. Shin. Generative visual code mobile world models. 2026.

Haodong Li, Jingwei Wu, Quan Sun, Guopeng Li, Juanxi Tian, Huanyu Zhang, Yanlin Lai, Ruichuan An, Hongbo Peng, Yuhong Dai, Chenxi Li, Chunmei Qing, Jia Wang, Ziyang Meng, Zheng Ge, Xiangyu Zhang, and Daxin Jiang. Gebench: Benchmarking image generation models as gui environments. 2026.

Andreas Lugmayr, Martin Danelljan, Andrés Romero, F. Yu, Radu Timofte, and L. Gool. Repaint: Inpainting using denoising diffusion probabilistic models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461, 2022.

Dezhao Luo, Bohan Tang, Kang Li, Georgios Papoudakis, Jifei Song, Shaogang Gong, Jianye Hao, Jun Wang, and Kun Shao. Vimo: A generative visual gui world model for app agent. *ArXiv*, abs/2504.13936, 2025.

Dang Nguyen, Jian Chen, Yu Wang, Gang Wu, Namyong Park, Zhengmian Hu, Hanjia Lyu, Junda Wu, Ryan Aponte, Yu Xia, Xintong Li, Jing Shi, Hongjie Chen, Viet Dac Lai, Zhouhang Xie, Sungchul Kim, Ruiyi Zhang, Tong Yu, Md Mehrab Tanjim, Nesreen K. Ahmed, Puneet Mathur, Seunghyun Yoon, Lina Yao, B. Kveton, T. Nguyen, Trung Bui, Tianyi Zhou, Ryan A. Rossi, and Franck Dernoncourt. Gui agents: A survey. *ArXiv*, abs/2412.13501, 2024.

Vikram S. Voleti, Alexia Jolicoeur-Martineau, and C. Pal. Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation. *ArXiv*, abs/2205.09853, 2022.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Da-Wei Liu, De mei Li, Hang Zhang, Hao Meng, Hu Wei, Ji-Li Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Min Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiao-Xue Xu, Yi Wang, Yichang Zhang, Yong-An Zhu, Yujian Wu, Yu-Jiao Cai, and Ze-Yang Liu. Qwen-image technical report. *ArXiv*, abs/2508.02324, 2025.

Zijun Zhou, Yingying Deng, Xiangyu He, Weiming Dong, and Fan Tang. Multi-turn consistent image editing. *ArXiv*, abs/2505.04320, 2025.