

Article

Not peer-reviewed version

UI-OCEANUS: Scaling GUI Agents with Synthetic Environmental Dynamics

[Mengzhou Wu](#), Yuzhe Guo, Yuan Cao, Haochuan Lu, Songhe Zhu, Pingzhe Qu, Xin Chen, Kang Qin, Zhongpu Wang, Xiaode Zhang, Xinyi Wang, Wei Dai, Gang Cao, Yuetang Deng, Zhi Gong, Dezhi Ran^{*,†}, Linyi Li, [Wei Yang](#), Tao Xie^{*}

Posted Date: 12 March 2026

doi: 10.20944/preprints202603.0980.v1

Keywords: GUI agents; world models; vision-language models; synthetic data; continual pre-training



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

UI-OCEANUS: Scaling GUI Agents with Synthetic Environmental Dynamics

Mengzhou Wu^{1,2}, Yuzhe Guo^{1,2}, Yuan Cao^{1,2}, Haochuan Lu³, Songhe Zhu³, Pingzhe Qu³, Xin Chen³, Kang Qin³, Zhongpu Wang³, Xiaode Zhang³, Xinyi Wang³, Wei Dai³, Gang Cao³, Yuetang Deng³, Zhi Gong³, Dezhi Ran^{1,2,*†}, Linyi Li⁴, Wei Yang⁵ and Tao Xie^{1,2,6,7,*}

¹ Key Lab of HCST (PKU), MOE; School of Computer Science, Peking University, China

² Beijing Tongming Lake Information Technology Application Innovation Center, China

³ WeChat, Tencent Inc., China

⁴ Simon Fraser University, Canada

⁵ University of Texas at Dallas, USA

⁶ Fudan University Institute of Systems for Advanced Computing, China

⁷ Shanghai Institute of Systems for Open Computing, China

* Correspondence: dezhiaran@pku.edu.cn (D.R.); taoxie@pku.edu.cn (T.X.)

† Project Leader

Abstract

Scaling generalist GUI agents is hindered by the data scalability bottleneck of expensive human demonstrations and the “distillation ceiling” of synthetic teacher supervision. To transcend these limitations, we propose **UI-OCEANUS**, a framework that shifts the learning focus from mimicking high-level trajectories to mastering interaction physics via ground-truth environmental feedback. Through a systematic investigation of self-supervised objectives, we identify that forward dynamics, defined as the generative prediction of future interface states, acts as the primary driver for scalability and significantly outweighs inverse inference. **UI-OCEANUS** leverages this insight by converting low-cost autonomous exploration, which is verified directly by system execution, into high-density generative supervision to construct a robust internal world model. Experimental evaluations across a series of models demonstrate the decisive superiority of our approach: models utilizing Continual Pre-Training (CPT) on synthetic dynamics outperform non-CPT baselines with an average success rate improvement of 7% on offline benchmarks, which amplifies to a 16.8% gain in real-world online navigation. Furthermore, we observe that navigation performance scales with synthetic data volume. These results confirm that grounding agents in forward predictive modeling offers a superior pathway to scalable GUI automation with robust cross-domain adaptability and compositional generalization.

Keywords: GUI agents; world models; vision-language models; synthetic data; continual pre-training

1. Introduction

Generalist GUI agents [1–5] traditionally rely on Behavioral Cloning (BC) [6–10] to map visual observations to actions. However, acquiring trajectory-level data is prohibitively expensive. Unlike static image-text pairs, valid execution traces require long-horizon sequences where a single error can invalidate the entire trajectory. This sensitivity makes synthesizing reliable data difficult and necessitates labor-intensive human annotation [7,9,11,12], creating a major bottleneck for data scalability.

To bypass this bottleneck, recent approaches [3,13] have adopted distillation-based Continual Pre-training (CPT), distilling knowledge from foundation models via static objectives like UI captioning [14] or grounding [15]. Critically, these methods suffer from a “distillation ceiling” [16]: they enhance the agent’s *static semantic understanding* but fail to capture the *temporal dynamics* of the environment. For instance, an agent might correctly identify a “submit” button (semantics) but stubbornly assume that

clicking it leads to submission, failing to anticipate that an incomplete form will actually trigger a validation error (dynamics). Lacking such grounded interaction physics, these agents remain bound by static semantic priors, unable to adapt when dynamic environmental states contradict their shallow heuristics. Therefore, there is a pressing need for a scalable CPT objective that derives supervision directly from autonomous environmental feedback.

To address these limitations, we propose **UI-OCEANUS**, a self-supervised training framework designed to equip agents with a robust GUI world model (Figure 1). The core insight in **UI-OCEANUS** is that the underlying set of atomic transitions (i.e., single-step observation-action-outcome tuples) in the GUI environment is both relatively *finite* and inherently *self-supervised*, providing a natural task-agnostic scaling dimension for CPT. The *finite* nature ensures that the environmental dynamics can be efficiently explored without an expert policy model or explicit task goal to reach high coverage. The *self-supervised* nature enables the environment to act as a rigorous verifier where the observed transition following the action serves as a ground-truth label of consequence. Both features enable agents to utilize task-agnostic exploration heuristics [17–21] to efficiently harvest vast quantities of these atomic transitions. Crucially, these atomic transitions ground the learning process in mastering the *fundamental mechanics* of the interface first. This effectively separates the acquisition of interaction physics from the execution of user intent: the agent learns “what is possible” (dynamics) from massive synthetic data before learning “what is desirable” (intent) from scarce expert demonstrations.

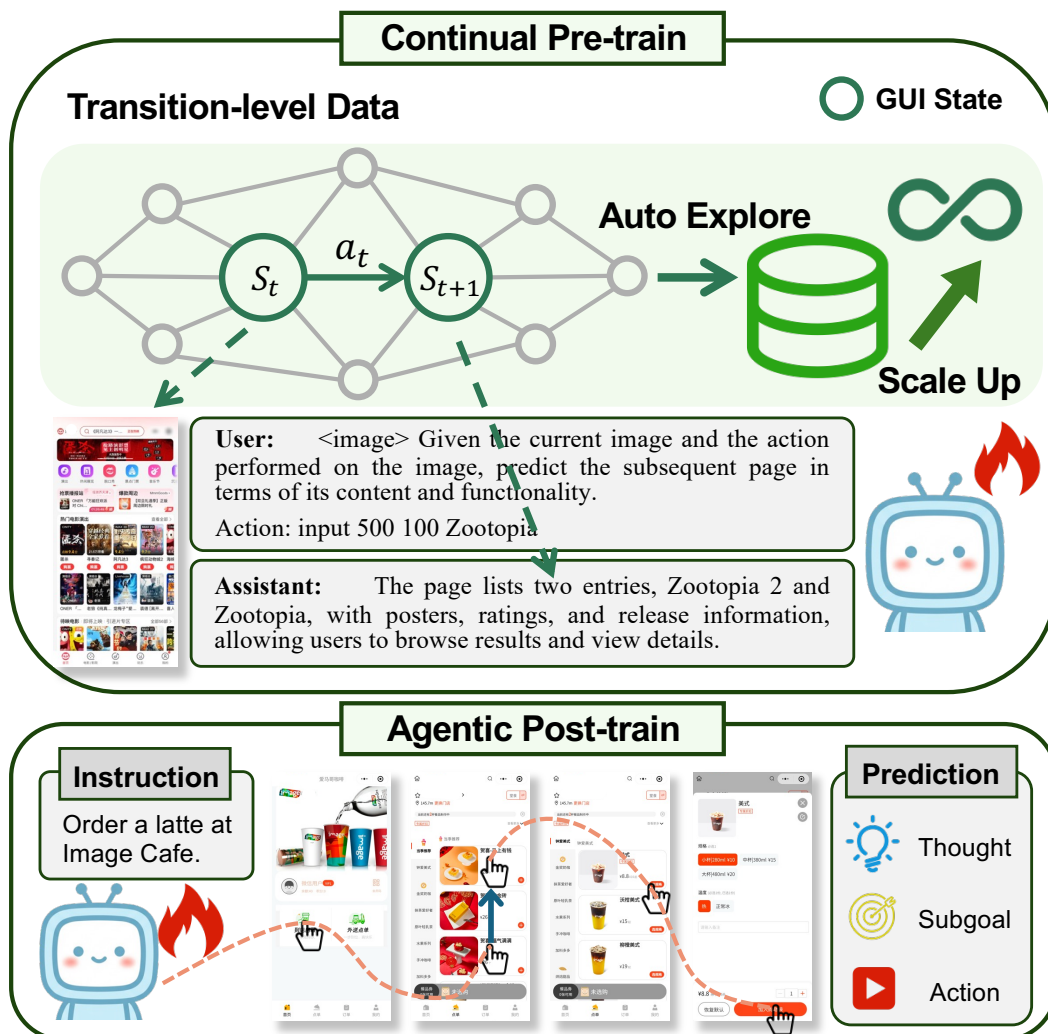


Figure 1. Constructing Generalist GUI Agents via Scalable World Model Learning. (Top) We first establish a robust physical foundation by learning a forward dynamics world model from massive, autonomously explored transitions. (Bottom) We then leverage this internalized world model to instantiate a generalist GUI agent through agentic post-training.

To realize this framework, we develop a scalable data engine capable of converting low-cost autonomous exploration into high-density supervision. We systematically investigate the spectrum of environmental dynamics and uncover a critical insight: forward dynamics, defined as the generative prediction of future interface states, serves as the key objective enabling performance scaling with data volume, significantly outperforming inverse dynamics and backward dynamics. Consequently, **UI-OCEANUS** prioritizes *forward dynamics prediction* as the generative objective [22,23]. Technically, we capture raw interaction traces, isolate unique and meaningful atomic transitions via strict hashing filtering and semantic filtering, and employ a Visual-Language Model (VLM) to synthesize precise descriptions of environmental changes. Finally, we implement a two-stage training strategy: the model is first continually pre-trained on a mixture of GUI dynamics, grounding, and general data to establish a robust world model, and subsequently specialized via agentic post-training to align this physical intuition with complex instruction following.

We conduct extensive empirical evaluations within the WeChat mini-program ecosystem [?], one of the world's largest and most diverse digital platforms. Hosting millions of distinct mini-programs for over a billion monthly active users (MAU), this ecosystem presents a virtually infinite space of interaction logic and visual layouts, demanding that agents master universal interaction dynamics rather than memorizing specific app interfaces. Notably, **UI-OCEANUS** demonstrates robust scalability on the WeChat mini-program offline benchmark, achieving an average improvement of +7% across seven different VLM backbones compared to strong baselines with identical Supervised Fine-Tuning. Crucially, these benefits are amplified in online navigation: **UI-OCEANUS** boosts the real-world success rate by 21.9% during the cold-start phase and maintains an average 15% lead even after both step-level and multi-step GRPO [24] alignment. Furthermore, our experiments yield three critical insights: (1) **Data scaling**: Performance scales log-linearly with synthetic data volume, showing no saturation up to 32B parameters and 3.2B tokens, confirming the scalability of atomic dynamics supervision. (2) **Mechanism**: Forward dynamics proves superior to inverse dynamics, as its higher predictive difficulty compels the model to learn more robust, semantic visual representations. (3) **Generalization**: The learned world model captures universal GUI physics effectively, enabling robust generalization to unseen mini-programs and Android environments, as well as compositional reasoning on complex multi-step tasks never seen during training.

We highlight main contributions as below:

- We pioneer a shift in GUI agent learning by separating the acquisition of atomic interaction physics from high-level instruction following. To the best of our knowledge, this is the first work to systematically scale environment dynamics as a self-supervised CPT objective for GUI agents, fundamentally bypassing the data scalability limitations of trajectory-based behavioral cloning.
- We propose **UI-OCEANUS**, a scalable framework designed to mine training supervision directly from intrinsic environmental signals.
- We conduct extensive evaluations validating the performance scaling of **UI-OCEANUS**. Our results demonstrate robust generalization across both domains and horizons, while ablation studies identify forward dynamics as the optimal objective for GUI world models.

2. Background and Related Work

Generalist GUI agents autonomously execute user instructions by interacting with Graphical User Interfaces (GUIs), thereby significantly boosting user productivity in daily tasks [25]. Pretrained from VLMs, generalist GUI agents are post-trained using Behavioral Cloning (BC) [3,9,10,26] or multi-step Reinforcement Learning (RL) [2,27,28]. However, post-training methods struggle under the capability ceiling of the base model [29,30], underscoring the necessity of Continual Pre-Training (CPT) prior to policy alignment. A fundamental challenge for CPT is the data scalability bottleneck, where collecting high-quality human demonstrations is prohibitively expensive. To bypass the challenge, existing research mainly uses the following two learning tasks in CPT: (1) static UI understanding task such as element grounding or screen captioning [3,13], which requires powerful foundation models to label

GUI screenshots and lacks grounding in real execution; (2) dynamic interacting with environments under instructions [31–34], which is notoriously difficult due to the low yield rates for high-quality long-horizon samples. Our **UI-OCEANUS** proposes a new CPT task: learning on forward dynamics of atomic transitions. Such dynamic learning task enables **UI-OCEANUS** to overcome the bias of distillation caused by lack of grounding, and the fragility of long-horizon trajectory synthesis.

Conceptualizing the GUI as a dynamic system aligns with the broader literature on world models. In the specific GUI domain, prior works have explored this direction through visual state prediction [35,36] and action prediction from state changes [37]. However, there still lacks a scalable, task-agnostic CPT objective to master global environment dynamics. We distinguish our work by leveraging autonomous exploration and intrinsic environmental feedback to internalize a GUI world model of the environment via CPT, establishing a robust dynamics foundation before agentic post-training.

Appendix A discusses related work in more detail.

3. Problem Formulation

We formalize the GUI navigation problem not merely as trajectory execution, but as learning the underlying transition dynamics of the interface.

GUI as a state transition graph.

We model the GUI environment as a directed state transition graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$. Each node $s \in \mathcal{S}$ represents a unique UI state, formally defined as a tuple $s = (I, X)$, where $I \in \mathbb{R}^{H \times W \times 3}$ is the pixel-level screenshot and X denotes structural metadata (e.g., the Accessibility Tree). A directed edge $e \in \mathcal{E}$ represents an atomic transition (s_t, a_t, s_{t+1}) . Here, an action $a_t \in \mathcal{A}$ is defined as a tuple $a_t = (\tau, \theta)$, where τ denotes the operation type (e.g., *click*, *scroll*, *input*) and θ represents the corresponding parameters (e.g., screen coordinates or input text content), with the full specification of \mathcal{A} detailed in Appendix C. Executing a_t on state s_t leads to state s_{t+1} .

Forward dynamics (outcome prediction).

This objective constitutes the agent’s world model, predicting the next state s_{t+1} given the current state s_t and action a_t . We formulate this by minimizing the negative log-likelihood:

$$\mathcal{L}_{\text{fwd}} = -\mathbb{E}_{(s_t, a_t, s_{t+1}) \sim \mathcal{D}} [\log P_{\theta}(s_{t+1} | s_t, a_t)] \quad (1)$$

Inverse dynamics (action inference).

Inverse dynamics aims to infer the action a_t responsible for the transition between s_t and s_{t+1} . The loss function is defined as:

$$\mathcal{L}_{\text{inv}} = -\mathbb{E}_{(s_t, a_t, s_{t+1}) \sim \mathcal{D}} [\log P_{\theta}(a_t | s_t, s_{t+1})] \quad (2)$$

Backward dynamics (precondition inference).

Complementary to forward dynamics, this objective learns to infer the precondition state s_t given the executed action a_t and the outcome s_{t+1} . We minimize:

$$\mathcal{L}_{\text{bwd}} = -\mathbb{E}_{(s_t, a_t, s_{t+1}) \sim \mathcal{D}} [\log P_{\theta}(s_t | s_{t+1}, a_t)] \quad (3)$$

4. The UI-OCEANUS Framework

To instantiate the theoretical world model defined in Section 3, we introduce **UI-OCEANUS**, a unified framework for learning GUI dynamics from large-scale autonomous interaction. As illustrated in Figure 2, **UI-OCEANUS** consists of four sequential stages: scalable acquisition via autonomous exploration, strict multi-step data filtering pipeline, grounded instruction generation based on envi-

ronmental feedback, and continual pre-training that optimizes a GUI dynamics world model on the constructed supervision.

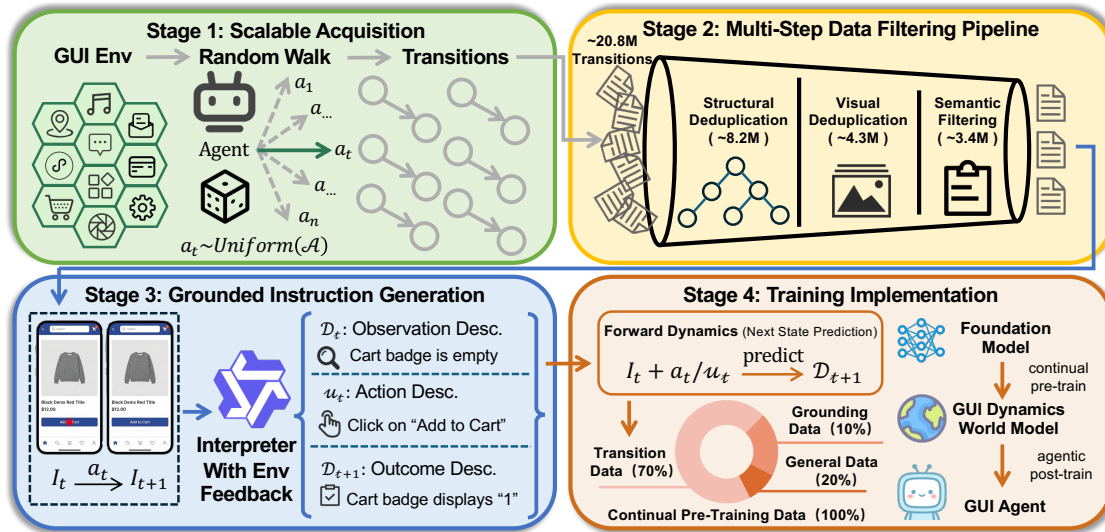


Figure 2. Overview of the proposed UI-OCEANUS framework. UI-OCEANUS consists of four sequential stages: (1) **Scalable Acquisition**, which autonomously explores diverse GUI applications to generate large-scale raw interaction trajectories; (2) **Multi-Step Data Filtering Pipeline**, which systematically filters and deduplicates raw interactions based on structural, visual, and semantic criteria; (3) **Grounded Instruction Generation**, which synthesizes multimodal instructions by interpreting transitions grounded in actual environmental feedback; and (4) **Training Implementation**, which employs forward dynamics for continual pre-training of the world model, followed by agentic post-training to finalize the GUI agent.

4.1. Scalable Acquisition via Autonomous Exploration

We autonomously explore the WeChat mini-program ecosystem to harvest open-domain transitions. Using a distributed fleet of 50 nodes, we employ a structured random walk policy: agents interact with UI elements identified via accessibility trees without human intervention. For each step t , we record the transition tuple (s_t, a_t, s_{t+1}) , where $s_t = (I_t, X_t)$ denotes the screenshot and structural metadata. This process yielded over 20.8M raw transitions over one month.

4.2. Multi-Step Data Filtering Pipeline

We refine \mathcal{D}_{raw} into a high-quality corpus \mathcal{D}_{cpt} via a three-stage pipeline(details in Appendix D): (1) **Structural Deduplication**: MinHash signatures on accessibility trees remove repeated content templates, reducing data to 8.2M. (2) **Visual Deduplication**: pixel-level pHash/dHash identify visually static transitions (e.g., buffering or invisible DOM updates), narrowing it to 4.3M. (3) **Semantic Flitering**: A VLM filters system errors and rendering artifacts by verifying action-feedback consistency, resulting in a final set of 3.4M transitions.

4.3. Grounded Instruction Generation

Following the data filtering stage, we convert the cleaned transition tuples into high-quality multimodal instructions using a vision-language model (Qwen3-VL-235B-A22B-Instruct [38]) as a semantic synthesizer. Unlike standard distillation methods, our approach utilizes the actual next state I_{t+1} provided directly by the environment, with the synthesizer acting solely as an interpreter to articulate the observed transition. This strategy effectively prevents the synthesizer from generating hallucinations typically found in purely model-based approaches.

For each transition (s_t, a_t, s_{t+1}) , we construct a visual prompt that includes the pre-transition screenshot I_t , annotated with a visual marker $m(a_t)$ indicating the action location (e.g., a circle highlighting the clicked area), along with the post-transition screenshot I_{t+1} . Given this prompt, the synthesizer generates a structured output consisting of three distinct components: (1) **Observation**

description (\mathcal{D}_t): a concise description of the initial GUI state before the action. (2) **Action description** (u_t): a brief natural-language summary describing the executed action. (3) **Outcome description** (\mathcal{D}_{t+1}): a precise description of the resulting GUI state after the action.

By generating instructions directly grounded in actual environmental state transitions, this stage produces large-scale, reliable, and high-fidelity data for CPT.

4.4. Training Implementation

4.4.1. Continual Pretraining

Using the curated dataset \mathcal{D}_{cpt} and generated grounded instructions, we construct a unified training corpus designed to facilitate understanding of GUI dynamics.

We define three distinct categories of training tasks to comprehensively capture GUI dynamics:

Forward dynamics: Given the screenshot of an initial state and an action (either as an exact action or a natural-language summary), the model predicts the resulting GUI state description. *Input:* $\mathcal{V} = [I_t, a_t]$ or $\mathcal{V} = [I_t, u_t]$; *Target:* $\mathbf{y} = \mathcal{D}_{t+1}$.

Inverse dynamics: Given screenshots of the initial and resulting states, the model predicts the action connecting them. *Input:* $\mathcal{V} = [I_t, I_{t+1}]$; *Target:* $\mathbf{y} = u_t$ or $\mathbf{y} = a_t$.

Backward dynamics: Given the screenshot of a resulting state and an action summary, the model predicts a description of the initial GUI state. *Input:* $\mathcal{V} = [u_t, I_{t+1}]$; *Target:* $\mathbf{y} = \mathcal{D}_t$.

Data Mixing. We construct the final training set \mathcal{D}_{mix} by strategically combining three distinct data sources: (1) **GUI dynamics (70%):** Informed by our ablation study (Section 5.3), we exclusively utilize Forward Dynamics ($I_t, a_t \rightarrow \mathcal{D}_{t+1}$) to maximize generative world modeling capabilities; (2) **General multimodal data (20%):** General data [39] acts as a regularizer to prevent the catastrophic forgetting of foundational capabilities; (3) **UI grounding data (10%):** Grounding data is incorporated to preserve fine-grained spatial localization skills, ensuring that precise execution capabilities are not compromised. We detail the specific training hyperparameters in Appendix F.1.

4.4.2. Agentic Post-Training

Following the CPT phase, we conduct an agentic post-training stage. In this phase, the model is fine-tuned on high-quality GUI agent trajectories to explicitly align the GUI world model with user intents, bridging the gap between physical understanding and task execution. We detail the specific training hyperparameters in Appendix F.2.

5. Experiments

In this section, we systematically evaluate the **UI-OCEANUS** framework. Our evaluation is structured to answer four critical questions: (1) **Scalability:** Can autonomous exploration serve as a scalable data source that yields consistent performance gains? (Section 5.1) (2) **Online efficacy:** Does the learned world model translate effectively to real-world online navigation and act as a persistent prior that synergizes with agentic post-training? (Section 5.2) (3) **Mechanism:** Which self-supervised objective yields the most significant performance gains for downstream GUI navigation? (Section 5.3) (4) **Generalization:** Does the learned world model generalize to unseen apps, novel domains (Android), and complex compositional tasks? (Section 5.4)

5.1. Scaling Laws of UI-OCEANUS

A central premise of our framework is that autonomous environmental feedback can overcome the data scarcity bottleneck. We first investigate the scaling properties of **UI-OCEANUS**.

5.1.1. Setup

Training configurations. During the CPT phase, we utilize the data mixture detailed in Section 4.4, totaling approximately 5M samples (3.2B tokens). For the subsequent agentic post-training phase, we employ 8K high-quality GUI navigation samples. To verify data scaling laws, we vary the volume of CPT data injected from 0% to 100%. **Evaluation Protocol.** To support large-scale experiments, we

evaluate performance on a held-out offline benchmark comprising 8K diverse mini-program tasks not seen during training. Following prior works [3,8,12], we report Exact Match (EM), which requires both the action type and its parameters to be correct. We assess performance across Qwen3-VL [38] (2B, 4B, 8B, 32B) and Qwen2.5-VL [40] (3B, 7B, 32B) families. We compare our method against two distinct baselines: (1) **Base Models**: We evaluate raw base models in a zero-shot setting. To decouple planning from grounding, we also report a “w/ grounding” setting, where an external specialized grounding model based on Qwen2.5-VL-32B is employed to predict an action based on the natural language action descriptions; (2) **SFT-only Baselines**: Models fine-tuned solely on the post-training data without any dynamics CPT (denoted as 0% data scale).

5.1.2. Results

As presented in Table 1 and visualized in Figure 3, our empirical findings validate the existence of scaling laws for GUI world models across both data and model dimensions.

Table 1. Scaling Laws of UI-OCEANUS. Evaluation of Exact Match (EM) scores on the held-out offline benchmark. The “0% (SFT)” column represents the standard Supervised Fine-Tuning baseline. By scaling the volume of continual pre-training dynamics data from 12.5% to 100%, UI-OCEANUS achieves consistent performance gains across all seven model backbones. The results validate a clear positive correlation between dynamics data volume and downstream navigation capability.

Model	Size	Reference		UI-OCEANUS Data Scale (+ SFT)				
		Base (Zero-shot)	w/ grounding	0% (SFT)	12.5%	25%	50%	100% (Ours)
<i>I. Qwen3-VL Series with UI-OCEANUS</i>								
Qwen3-VL	2B	2.4	36.5	47.9	49.4	51.3	51.7	52.7
Qwen3-VL	4B	41.7	52.0	57.3	58.3	59.0	59.3	60.0
Qwen3-VL	8B	42.3	54.9	58.1	58.0	59.6	60.4	60.8
Qwen3-VL	32B	47.1	61.2	61.5	61.0	63.7	62.5	65.0
<i>II. Qwen2.5-VL Series with UI-OCEANUS</i>								
Qwen2.5-VL	3B	1.1	27.9	47.7	49.2	48.5	49.6	50.6
Qwen2.5-VL	7B	13.0	37.8	51.0	53.5	53.3	53.4	55.7
Qwen2.5-VL	32B	33.9	56.5	60.5	61.6	63.2	64.2	64.8

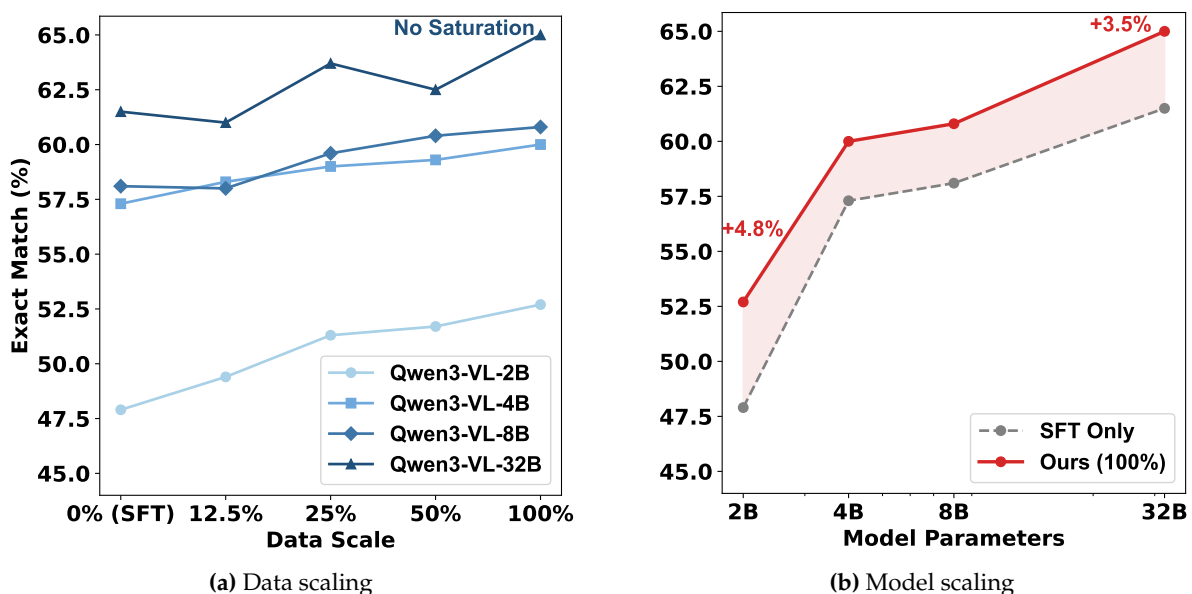


Figure 3. Scaling behavior of Qwen3-VL series models.

Data scaling.

We observe a consistent positive correlation between the volume of pre-training dynamics data and downstream performance. Starting from the standard SFT baseline (0% data scale), the injection of our dynamics data yields consistent improvements across all model sizes. For instance, the Qwen3-VL-2B model improves from 47.9% (0%) to 52.7% (100%), while the larger Qwen3-VL-32B advances from 61.5% to 65.0%. The performance curves exhibit a strong log-linear trend, confirming that autonomous exploration serves as a scalable data source.

Model scaling.

UI-OCEANUS demonstrates robust scalability with model size. Comparing the Qwen3-VL series, the EM score improves steadily from 52.7% (2B) to 65.0% (32B) as parameters increase. Crucially, the performance gains over the SFT baseline remain robust across all scales. This indicates that the internalized interaction dynamics provide fundamental physical knowledge that complements even larger capacity models, rather than diminishing as model size grows.

Absence of saturation.

Notably, we observe no signs of performance saturation across either dimension. Even at the maximal data scale (5M samples, 3.2B tokens) and model size (32B), the trend lines maintain their upward trajectory. This suggests that further scaling would likely yield continued gains.

5.2. Online Evaluation: Synergy with Post-Training

While offline metrics quantify predictive accuracy, the ultimate test lies in online execution. We investigate the effectiveness of **UI-OCEANUS** when integrated into a progressive post-training pipeline.

5.2.1. Setup

Training configurations. We conduct a comparative analysis between agents initialized with and without our proposed CPT (**UI-OCEANUS** vs. w/o CPT). The post-training pipeline consists of three progressive stages: (1) **SFT (cold start)**: Supervised fine-tuning using simple samples from the SFT dataset to establish basic instruction-following capabilities; (2) **Step-level GRPO**: Optimization on hard samples from the SFT dataset using dense step-wise rewards to refine complex reasoning; (3) **Multi-step GRPO**: Further aligning using approximately 1K instructions in the real environment, utilizing a VLM as an outcome reward model to provide sparse trajectory-level reward.

Evaluation protocol. We report the Success Rate (SR) on a held-out benchmark of 149 instructions covering diverse mini-programs. All evaluation episodes are executed in the real online environment, and task success is manually verified by human annotators to ensure rigorous correctness.

5.2.2. Results

As shown in Table 2, **UI-OCEANUS** provides a robust performance foundation that synergizes effectively with the entire post-training pipeline.

Table 2. Online Evaluation across Training Stages. Success Rate (SR) on live mini-programs. Δ shows the gain from UI-OCEANUS.

Training Stage	SR (%)		Δ
	w/o CPT	UI-OCEANUS	
<i>Qwen3-VL-8B</i>			
SFT + Step-level GRPO	27.5	30.9	+12.4%
<i>Qwen2.5-VL-32B</i>			
SFT (Cold Start)	24.2	29.5	+21.9%
+ Step-level GRPO	32.2	38.9	+20.8%
+ Multi-step GRPO	39.6	44.3	+11.9%

Consistent superiority across post-training stages.

UI-OCEANUS establishes an immediate advantage in the initial SFT phase and, crucially, maintains this lead throughout the subsequent RL alignment steps. Across both step-level and multi-step GRPO stages, agents initialized with our world model consistently outperform the baselines (e.g., a +20.8% relative gain in step-level GRPO). This persistence demonstrates that the internalized physical understanding is not overwritten by optimization but rather serves as an intrinsic prior that facilitates more efficient policy learning.

Amplified benefits in online environments.

Notably, the performance gains yielded by **UI-OCEANUS** are more pronounced in the online setting compared to the offline benchmark (Section 5.1). While offline evaluation relies on static matching, online execution demands resilience to dynamic rendering variations and error recovery. The amplified gain in this challenging environment highlights that our world model equips the agent with the robust physical understanding necessary for real-world interaction.

5.3. Mechanism: The Superiority of Forward Dynamics

We dissect the contributions of different CPT objectives to understand the source of the performance gains.

5.3.1. Setup

Training configuration. To isolate the impact of objective formulation, we conduct controlled experiments across three model scales (Qwen3-VL-2B, 4B, and Qwen2.5-VL-3B). We compare the three distinct categories of self-supervised tasks detailed in Section 4.4, while keeping the underlying UI transitions corpus constant (5M samples).

Evaluation protocol. We adhere to the identical SFT and evaluation dataset settings established in Section 5.1. In addition to the previously defined EM, we report Type Match (TM), which solely evaluates the classification accuracy of the predicted action type against the ground truth.

5.3.2. Results and Analysis

As detailed in Table 3, our ablation study reveals critical insights into how different physical objectives shape the agent’s representations.

Table 3. Ablation Study on Continual Pre-training Objectives. We compare different dynamics tasks against the non-CPT baseline. **Key Insight:** Forward dynamics ($I_t, u_t/a_t \rightarrow \mathcal{D}_{t+1}$) consistently achieve the highest performance across all scales. The “Overall” column reports the average TM and EM scores across the three model scales. Best results are bolded, and second-best are underlined.

Dynamics	Formulation	Qwen3-VL-2B		Qwen3-VL-4B		Qwen2.5-VL-3B		Overall	
		TM	EM	TM	EM	TM	EM	TM	EM
<i>Baseline (w/o CPT)</i>		75.9	47.9	79.2	57.3	75.7	47.7	76.9	51.0
Forward	$I_t, u_t \rightarrow \mathcal{D}_{t+1}$	77.7	52.7	81.0	60.3	<u>76.7</u>	48.8	78.5	<u>53.9</u>
	$I_t, a_t \rightarrow \mathcal{D}_{t+1}$	<u>77.6</u>	52.7	<u>80.3</u>	<u>60.0</u>	76.9	50.6	<u>78.3</u>	54.4
Inverse	$I_t, I_{t+1} \rightarrow u_t$	75.8	49.5	75.8	49.5	75.3	<u>49.0</u>	75.6	49.3
	$I_t, I_{t+1} \rightarrow a_t$	75.6	46.4	77.8	50.9	75.6	47.2	76.3	48.2
	$I_t, \mathcal{D}_{t+1} \rightarrow u_t$	76.3	50.2	79.0	57.4	76.2	48.3	77.2	52.0
	$I_t, \mathcal{D}_{t+1} \rightarrow a_t$	75.7	46.6	78.0	52.5	75.8	47.6	76.5	48.9
Backward	$u_t, I_{t+1} \rightarrow \mathcal{D}_t$	77.1	<u>50.4</u>	80.0	58.7	76.3	47.5	77.8	52.2

The dominance of forward dynamics.

Forward dynamics dominates, with the simplest formulation $I_t, a_t \rightarrow \mathcal{D}_{t+1}$ objective achieving the highest EM (54.4%). Predicting “*what happens next*” forces the model to encode the causal logic of the environment. Crucially, distinct from methods employing separate world models for planning [35,41], our approach embeds this transition reasoning directly into the agent’s representation. This eliminates the need for external models, allowing the agent to reason about dynamics implicitly.

The failure of inverse dynamics.

A counter-intuitive finding is the poor performance of Inverse Dynamics. Despite sharing the identical action space to the navigation task, variants like $I_t, I_{t+1} \rightarrow a_t$ yield an Overall EM of only 48.2%, performing even worse than the non-CPT baseline (51.0%). We attribute this degradation to two factors:

- **Sparse supervision signal:** Unlike forward dynamics which generates rich textual descriptions, the inverse task only outputs action types and coordinates. This limited token coverage and lack of diversity restrict the model’s ability to learn dense, semantically meaningful representations during CPT.
- **Insufficient task difficulty:** As illustrated in Figure A1, inverse dynamics converges rapidly to a low loss value. This indicates that the task lacks sufficient predictive complexity. Upon rapid convergence, the objective ceases to provide a meaningful gradient signal, rendering it ill-suited for effective CPT where sustained predictive challenge is essential [42–45].

Backward dynamics and temporal directionality.

While backward dynamics ($u_t, I_{t+1} \rightarrow \mathcal{D}_t$) also requires learning environment physics, it lags behind forward dynamics (52.2% vs. 54.4%). This indicates that while “reconstructing the past” offers some regularization benefits, it is less effective than “predicting the future”, which directly aligns with the agent’s inferential goal of planning subsequent steps based on current observations.

5.4. Cross-Domain and Compositional Generalization of World Model

Finally, we investigate whether the agent has merely memorized specific transition pairs or internalized a robust world model capable of cross-domain transfer and compositional reasoning. To strictly assess this, we define two levels of tasks: Level 1 (L1, atomic) utilizes standard single-step transitions ($t \rightarrow t + 1$) to evaluate cross-domain transfer performance; Level 2 (L2, compositional) introduces multi-step chaining ($t \rightarrow t + 2$) which never seen during training to assess compositional generalization, testing the ability to combine atomic physical rules for long-horizon prediction.

5.4.1. Setup

Dataset construction. We construct a controlled evaluation benchmark using atomic transitions from mini-programs, partitioned at the application level into seen and unseen groups. We randomly sample 1.2M transitions for training, while reserving 10K samples for L1 test sets and constructing 10K samples for L2 test sets within each group. To evaluate out-of-distribution (OOD) robustness, we additionally sample 10K random transitions from AndroidControl [12].

Training configurations. We train two specialized variants of **UI-OCEANUS-7B** using the same data mixture as in Section 5.3. Specifically, **UI-OCEANUS-7B-F** is trained on L1 forward tasks ($I_t, u_t \rightarrow \mathcal{D}_{t+1}$), and **UI-OCEANUS-7B-I** is trained on L1 inverse tasks ($I_t, I_{t+1} \rightarrow u_t$). Both models are trained on approximately 1.7M total samples.

Evaluation protocol. We employ VLM-as-a-Judge to handle the open-ended nature of generative predictions. For forward dynamics, we explicitly query the model to predict 5 distinct UI elements of the outcome page to facilitate objective verification via a VLM judge. For inverse dynamics, the model infers the first action given the start and end states, with correctness verified by a VLM judge.

5.4.2. Results

As shown in Table 4, **UI-OCEANUS** demonstrates robustness across both domain shifts and reasoning horizons.

Table 4. Cross-Domain and Compositional Generalization. We evaluate the learned world model across varying distribution shifts (mini-programs \rightarrow Android) and reasoning horizons (atomic L1 \rightarrow compositional L2). **UI-OCEANUS** demonstrates robust cross-domain transfer and the ability to chain atomic rules for multi-step reasoning. We abbreviate mini-programs as mini-P.

Lvl	Task	Model	Seen mini-P.	Unseen mini-P.	OOD Android
L1	Forward $I_t, u_t \rightarrow \mathcal{D}_{t+1}$	Qwen2.5-VL-7B	41.2	43.1	46.5
		Qwen2.5-VL-32B	48.5	49.6	65.8
		UI-OCEANUS-7B-F	53.1	56.9	49.1
	Inverse $I_t, I_{t+1} \rightarrow u_t$	Qwen2.5-VL-7B	17.0	18.4	31.1
		Qwen2.5-VL-32B	37.0	37.5	67.3
		UI-OCEANUS-7B-I	62.3	58.9	55.3
L2	Forward $I_t, u_t, u_{t+1} \rightarrow \mathcal{D}_{t+2}$	Qwen2.5-VL-7B	41.3	41.1	/
		Qwen2.5-VL-32B	46.9	46.3	/
		UI-OCEANUS-7B-F	47.0	47.2	/
	Inverse $I_t, I_{t+2} \rightarrow u_t$	Qwen2.5-VL-7B	6.7	6.2	/
		Qwen2.5-VL-32B	26.5	26.1	/
		UI-OCEANUS-7B-I	38.7	37.6	/

Cross-domain generalization.

Our model exhibits strong transfer capabilities across distribution shifts. Notably, on L1 Forward tasks, even under a challenging format shift where the model must adapt from generating holistic descriptions (training) to listing specific elements (inference), **UI-OCEANUS-7B** still achieves 56.9 on unseen mini-programs, surpassing the significantly larger Base-32B (49.6). This indicates that our method learns universal interaction logic rather than memorizing app-specific layouts. Furthermore, on the challenging OOD Android benchmark, **UI-OCEANUS-7B** consistently outperforms its base model across all tasks, confirming effective cross-platform transfer.

Compositional generalization.

Crucially, although trained strictly on single-step transitions ($s_t \rightarrow s_{t+1}$), **UI-OCEANUS** exhibits robust compositional generalization on Level 2 tasks. This implies that **UI-OCEANUS** does not merely memorize single-step transitions but enables the agent to mentally “chain” atomic physical rules. This capability equips the agent with the look-ahead simulation capabilities necessary for long-horizon planning.

6. Conclusion

We introduce **UI-OCEANUS**, a framework that overcomes the data scalability bottleneck by learning a GUI world model from autonomous exploration. By shifting from trajectory imitation to mastering atomic interaction physics, we unlock a scalable supervision source independent of human annotation. Our results identify forward dynamics as the critical objective for this capability, yielding log-linear performance scaling on the massive WeChat ecosystem with no signs of saturation up to 32B parameters and 3.2B tokens. Crucially, this internalized world model acts as a persistent policy prior: it boosts real-world online performance by over 15% and serves as a robust foundation for agentic post-training. By separating the acquisition of interaction physics from the execution of user intent, **UI-OCEANUS** provides a scalable pathway toward generalist agents capable of navigating complex, open-domain interfaces.

Appendix A. Extended Discussion on Related Work

This appendix extends the discussion of background and related work in Section 2.

Appendix A.1. Learning Paradigms for GUI Agents

Recent advancements in generalist GUI agents have primarily focused on refining policy optimization paradigms. While initial efforts relied on straightforward Behavioral Cloning (BC) [3,9,10,26], the field has rapidly shifted towards multi-step Reinforcement Learning (RL) [2,27,28] integrated with Chain-of-Thought (CoT) [46] reasoning to handle non-stationary, long-horizon tasks. However, recent studies on RL reveal a critical bottleneck: RL alone struggles to surpass the inherent capability ceiling of the base model [29,30]. This underscores the necessity of elevating the foundation model's intrinsic capabilities via Continual Pre-Training (CPT) prior to policy alignment. Existing CPT approaches for GUI agents, however, are largely confined to static UI understanding, such as element grounding or screen captioning [3,13]. While these methods improve perception, they fail to capture the causal mechanics of interaction. Distinct from these static objectives, our work targets a foundational dynamic objective: the internalization of a "World Model" via CPT on atomic transitions. By grounding the model in the forward dynamics of the interface (i.e., predicting consequences), we effectively elevate the capability ceiling of the base model, providing a scalable path to establishing this essential foundation for generalist GUI agents.

Appendix A.2. Scalable Data Acquisition for GUI Agents

A fundamental challenge to training generalist GUI agents is the data scalability bottleneck of Behavioral Cloning (BC). Practically, collecting high-quality human demonstrations is prohibitively expensive. To bypass these limitations, the field has pivoted toward synthetic data generation, primarily following two directions. One line of work attempts to synthesize execution trajectories by deploying agents to interact with the environment under specific instructions [31–34]. However, due to the strict long-horizon dependencies of GUI tasks, generating valid data is notoriously difficult; a single intermediate error invalidates the entire trajectory, resulting in extremely low yield rates for high-quality samples. Alternatively, other approaches focus on static tasks such as screen captioning or VQA by leveraging powerful foundation models to label GUI screenshots [1,3,13]. Yet, this relies on Model-based Distillation, creating an inherent "distillation ceiling": the agent is bounded by the teacher model's hallucinations and lacks objective grounding in real execution. In contrast, our **UI-OCEANUS** framework shifts the source of supervision from fallible teacher models to intrinsic environmental dynamics. By treating state transitions as objective verification, we overcome the fragility of long-horizon synthesis and the bias of distillation, ensuring that performance scales robustly with autonomous exploration.

Appendix A.3. World Models and Dynamics in Digital Interfaces

Conceptualizing the GUI as a dynamic system aligns with the broader literature on world models in model-based Reinforcement Learning and representation learning, where frameworks [47–49] enable agents to plan by modeling future latent states. In the specific domain of GUIs, prior works such as ViMo [35] and VAGEN [36] have explored this direction through explicit visual state prediction, while GUI-Shift [37] employs an inverse dynamics objective to infer actions from state changes. Recent concurrent work [50] proposes utilizing implicit world modeling as an auxiliary task, yet remains constrained by local branching from expert demonstrations, lacking a scalable, task-agnostic Continual Pre-Training objective to master global environment dynamics. We distinguish our work by leveraging autonomous exploration and intrinsic environmental feedback to internalize a GUI world model via Continual Pre-Training, establishing a robust dynamics foundation before agentic post-training.

Appendix B. Prompt Templates

We present the specific prompt templates utilized for the synthetic environmental dynamics Continual Pre-Training (CPT) of **UI-OCEANUS** and the downstream GUI agent navigation tasks.

Appendix B.1. Synthetic Environmental Dynamics Tasks

The CPT tasks are designed to enable the model to master the underlying transition dynamics of the GUI environment. We adopt the following notations: I_t denotes the screenshot at step t ; u_t denotes the natural language description of the action; a_t denotes the atomic action (e.g., coordinates); and \mathcal{D}_t denotes the textual description of the interface state.

Forward Dynamics ($I_t, u_t \rightarrow \mathcal{D}_{t+1}$ and $I_t, a_t \rightarrow \mathcal{D}_{t+1}$).

This task requires the model to predict the future state description given the current image and an action.

Forward Dynamics Prompt Template

System: You are a GUI operation expert.

User: <image> Given the current image and the action performed on the image, predict the subsequent page in terms of its content and functionality.

Input: {action_summary} or {atomic_action}

Inverse Dynamics ($I_t, I_{t+1} \rightarrow u_t/a_t$).

This task infers the action performed between two states.

Inverse Dynamics Prompt Template

System: You are a GUI operation expert.

User: <image> <image> Given the before and after images, predict the operation performed by the user.

Constraint: Describe the operation in natural language OR select from atomic actions (click $x y$, input $x y t$, scroll $x y direction$).

Inverse Dynamics with Goal ($I_t, \mathcal{D}_{t+1} \rightarrow u_t/a_t$).

The agent infers the action required to reach a specific described state.

Inverse Dynamics (State Description) Prompt Template

System: You are a GUI operation expert.

User: <image> Given the current interface image, predict the operation performed by the user that would lead to the page described below.

Target Description: {after_screenshot_summary}

Backward Dynamics ($u_t, I_{t+1} \rightarrow D_t$).

This task reconstructs the previous state description.

Backward Dynamics Prompt Template

System: You are a GUI operation expert.

User: <image> Given the resulting page image and the intermediate action description, predict the content and functionality of the previous page.

Input: {action_summary}

Appendix B.2. GUI Navigation Task

For the downstream navigation task, we utilize a Chain-of-Thought (CoT) approach.

Navigation Prompt

<image> <image>

You are an advanced GUI navigation assistant capable of perceiving and interacting with mobile interfaces. The provided image sequence represents the history of operations, with the last image showing the current UI state.

Current Instruction: {instruction}

User History: {history}

Based on the visual context and history, predict the next optimal operation. Your output must strictly adhere to the following XML-style format:

<think>...</think><sub_goal>...</sub_goal><answer>...</answer>

1. Thinking Process (<think>)

Conduct a step-by-step reasoning process: (1) Analyze the current UI layout and identify interactive elements; (2) Verify if previous actions were successful; (3) Reflect on any potential errors or deviations; (4) Formulate a high-level strategy to align with the user's instruction.

2. Sub-goal Planning (<sub_goal>)

Translate your strategy into a concise, natural language description of the immediate next step (e.g., "Click the search icon on the top right", "Scroll down to find the settings").

3. Action (<answer>)

Map the sub-goal to a precise executable primitive from the following 5 categories:

1. `click x y` – Tap at coordinates (x, y) .
2. `input x y t` – Focus at (x, y) and type text t .
3. `finish` – Terminate the task successfully.
4. `scroll x y dir` – Scroll from (x, y) in direction dir .
5. `wait` – Pause for system response.

Appendix B.3. Generalization Evaluation Prompts

To rigorously assess cross-domain transfer (Level 1) and compositional generalization (Level 2) as discussed in Section 5.4, we employ specialized prompt templates that facilitate objective verification via a VLM judge.

Forward Dynamics Generalization (L1 & L2).

Unlike the standard CPT objective which generates a holistic description, the evaluation prompts require the model to explicitly predict 5 distinct UI elements.

Forward Dynamics L1 (Atomic) Evaluation Prompt

System: You are a GUI operation expert.

User: <image> Given the current image and an action description, predict the 5 key elements that will appear on the subsequent page.

Action Description: {action_description}

Please output exactly five elements following the format below:

```
<element_1>Description of element 1, be concise</element_1>
<element_2>Description of element 2, be concise</element_2>
<element_3>Description of element 3, be concise</element_3>
<element_4>Description of element 4, be concise</element_4>
<element_5>Description of element 5, be concise</element_5>
```

Forward Dynamics L2 (Compositional) Evaluation Prompt

System: You are a GUI operation expert.

User: <image> Given the current image and two consecutive action descriptions, predict the 5 key elements that will appear on the subsequent page after executing Action 1 on the current page, followed by Action 2 on the transitioned page.

Action Description 1: {action_description_1}

Action Description 2: {action_description_2}

Please output exactly five elements following the format below:

```
<element_1>Description of element 1, be concise</element_1>
<element_2>Description of element 2, be concise</element_2>
<element_3>Description of element 3, be concise</element_3>
<element_4>Description of element 4, be concise</element_4>
<element_5>Description of element 5, be concise</element_5>
```

Inverse Dynamics Generalization (L2).

For Level 2 Inverse Dynamics, the model must infer the initial action in a two-step chain, given only the start and end states (skipping the intermediate observation).

Inverse Dynamics L2 (Compositional) Evaluation Prompt

System: You are a GUI operation expert.

User: <image> <image> Given the first and the third images, please describe in natural language the operation performed on the first image that causes it to generate an intermediate page, which is subsequently transformed into the third image via another operation?

Appendix B.4. VLM-based Verification Judges

To automate the evaluation of open-ended generation in our generalization experiments (Section 5.4), we employ two distinct VLM judges. Both judges utilize a standardized structure to ensure rigorous and consistent verification.

Judge for Inverse Dynamics (Action Consistency).

This judge evaluates whether the predicted action is visually and functionally equivalent to the ground truth (Binary Score: 0 or 1).

Inverse Dynamics Judge Prompt

System: You are a professional UI automation judge and visual semantic evaluation expert.

Task: Determine whether the “Model Predicted Action” is visually and functionally consistent with the “Ground Truth Action” based on the provided UI screenshot.

Input:

- **UI Screenshot:** <image>
- **Ground Truth Action:** {ground_truth}
- **Model Predicted Action:** {prediction}

Evaluation Criteria:

- **Visual Alignment:** The predicted action must target the same visual element (e.g., same icon/button) as the ground truth.
- **Action Compatibility:** The action type (e.g., Click, Scroll) must be consistent.
- **Flexibility:** Synonyms (e.g., “Submit” vs “Confirm”) and unambiguous location-based descriptions are accepted.

Output Format:

<reason>Brief analysis of visual alignment.</reason>
<score>1 (Pass) or 0 (Fail)</score>

Judge for Forward Dynamics (Element Prediction).

This judge evaluates the accuracy of the predicted next-state elements by calculating the precision of the top-5 predictions (Discrete Score: 0, 0.2, ..., 1).

Forward Dynamics Judge Prompt

System: You are a professional UI automation judge and visual semantic evaluation expert.

Task: Verify whether the UI elements listed in the “Predicted Answer” are actually present in the provided “Reference Image” (Ground Truth Next State).

Input:

- **Reference Screenshot:** <image>
- **Predicted Answer:** {predicted_elements}

Evaluation Criteria:

- **Truncation Rule:** Evaluate **only the first 5 elements** in the predicted list. Ignore any subsequent elements.
- **Existence Check:** For each element, check if it is clearly visible in the Reference Screenshot.
- **Scoring Metric:** Calculate the score as $\frac{\text{Count of Existing Elements}}{5}$.
- **Score Set:** The final score must be exactly one of: {0, 0.2, 0.4, 0.6, 0.8, 1}.

Output Format:

<reason>Briefly list which of the top-5 elements exist in the image.</reason>
<score>Score</score>

Appendix C. Action Space Specification

We define the action space \mathcal{A} as a set of atomic operations supported by the agent. Each action is represented as a tuple $a_t = (\tau, \theta)$, where τ specifies the primitive type and θ contains the necessary parameters. The representation of spatial coordinates (x, y) adapts to the underlying vision-language backbone:

- **Qwen2.5-VL Series:** Utilizes **absolute pixel coordinates** $(x, y) \in [0, W] \times [0, H]$ based on the original screenshot resolution.
- **Qwen3-VL Series:** Utilizes **normalized coordinates** quantized to the integer range $[0, 1000]$.

The specific primitives and their parameter structures are summarized in Table A1.

Table A1. Definition of the Action Space.

Action Type (τ)	Parameters (θ)	Description
CLICK	(x, y) Absolute pixels or normalized to $[0, 1000]$	Tap at the specified coordinates on the screen.
INPUT	(x, y, text)	Focus on the target element at (x, y) and type the provided text string.
SCROLL	(x, y, dir) $\text{dir} \in \{\text{UP}, \text{DOWN}, \text{LEFT}, \text{RIGHT}\}$	Perform a scroll gesture starting from (x, y) towards the specified direction.
FINISH	\emptyset	Terminate the current episode immediately.

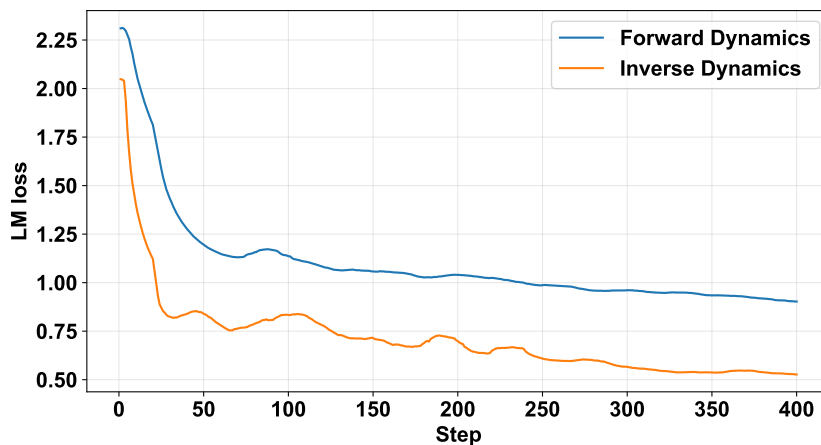


Figure A1. Training Loss Comparison. Inverse Dynamics (orange) exhibits rapid saturation, indicating insufficient task difficulty. In contrast, Forward Dynamics (blue) maintains a higher loss level, providing the sustained gradient signal necessary for effective representation learning.

Appendix D. Details of Data Filtering Pipeline

This section details the implementation of the deduplication stages using Locality-Sensitive Hashing (LSH) for both graph-structured accessibility trees and high-dimensional screenshots.

Appendix D.1. Structural Transition Deduplication

To identify interface states with identical logic but varying content, we model transitions as sets of structural tokens.

Tokenization & MinHash.

We extract tokens from the accessibility tree nodes based on semantic attributes (e.g., tag, xpath, events). A transition is represented by the union of hashed tokens from both the pre- and post-action states. We generate MinHash signatures to efficiently estimate Jaccard similarity, reducing variable-length token sets to fixed-length vectors.

LSH Indexing.

We employ a banding strategy to index these signatures. Transitions colliding in any band are retrieved as candidates. We then verify candidates against a Jaccard similarity threshold, retaining only unique representatives for the final dataset.

Appendix D.2. Visual Transition Deduplication

For opaque components (e.g., WebViews), we utilize a composite visual fingerprinting strategy combining frequency and gradient domains.

Composite Hashing.

For each transition, we concatenate the **Perceptual Hash (pHash)** and **Difference Hash (dHash)** of the start and end screenshots. This combination ensures robustness against rendering noise while remaining sensitive to significant layout changes.

Clustering Strategy.

We employ bit-sampling LSH to project high-dimensional hashes into lower-dimensional buckets for indexing. Candidates are clustered using a Union-Find data structure based on Hamming distance thresholds. For each connected component, the transition with the highest data source priority is retained.

Appendix D.3. Semantic Filtering

We utilize Qwen3-VL-235B-Instruct as a semantic verifier. The model is prompted to output a binary validity score, retaining only transitions where the screen changes logically correspond to the annotated action.

Appendix E. Training Dynamics Analysis

Figure A1 illustrates the training dynamics discussed in Section 5.3. The precipitous drop in Inverse Dynamics loss confirms that the task is computationally trivial, failing to provide meaningful gradient updates early in training. Conversely, Forward Dynamics imposes a sustained predictive challenge, driving continuous optimization of the world model's representations.

Appendix F. Hyperparameter Settings

We detail the hyperparameter configurations for both the Continual Pre-Training (CPT) phase and the Agentic Post-Training phase.

Appendix F.1. Continual Pre-Training (CPT)

The hyperparameters for the CPT phase are summarized in Table A2.

Table A2. Hyperparameters for Continual Pre-Training.

Hyperparameter	32B Models	7B-8B Models	2B-4B Models
Optimizer		AdamW	
Global Batch Size	2048	1024	1024
Peak Learning Rate	1×10^{-5}	2×10^{-5}	3×10^{-5}
LR Scheduler		Cosine	
Warmup Ratio		0.03	
Max Sequence Length		4096	
Weight Decay	0.1	0.01	0.01
Epochs		1	

Appendix F.2. Agentic Post-Training: SFT

Table A3 lists the settings for the Supervised Fine-Tuning stage.

Table A3. Hyperparameters for SFT.

Hyperparameter	32B Models	7B-8B Models	2B-4B Models
Optimizer		AdamW	
Global Batch Size		64	
Peak Learning Rate	3×10^{-6}	6×10^{-6}	1×10^{-5}
LR Scheduler		Constant	
Max Sequence Length		4096	
Weight Decay		0	
Epochs		2	

Appendix G. SOTA Comparisons

We evaluate the performance ceiling of our framework by comparing it against state-of-the-art proprietary models and large-scale open-weight baselines on both offline and online benchmarks.

Appendix G.1. Offline Scalability Benchmark Comparison

Table A4 presents the performance comparison between our **UI-OCEANUS** framework and state-of-the-art proprietary models (including GPT-5.2, Claude-4.5-Opus, and Gemini-3-Flash) as well as large-scale open-weight baselines on the offline scalability benchmark.

Table A4. Offline Benchmark Performance vs. Proprietary & Large-Scale Models. We report Exact Match (EM) and Type Match (TM) scores across different model families.

Model	Exact Match (EM)	Type Match (TM)
<i>Proprietary Models</i>		
GPT-5.2 [51] (w/ Grounding)	61.1	76.9
Claude-4.5-Sonnet [52] (w/ Grounding)	64.8	81.2
Claude-4.5-Opus [53] (w/ Grounding)	69.2	82.7
Gemini-3-Flash [54] (w/ Grounding)	69.2	82.0
<i>Open-Weight Baselines</i>		
Qwen2.5-VL-32B	33.9	56.6
Qwen2.5-VL-72B	51.4	77.2
Qwen3-VL-32B	47.1	79.2
Qwen3-VL-235B-A22B	47.5	79.4
Qwen2.5-VL-32B (w/ Grounding)	56.5	77.9
Qwen2.5-VL-72B (w/ Grounding)	56.8	77.2
Qwen3-VL-32B (w/ Grounding)	61.2	79.3
Qwen3-VL-235B-A22B (w/ Grounding)	61.5	79.3
<i>Ours</i>		
Qwen3-VL-32B + UI-OCEANUS (CPT+SFT)	65.0	81.1
Qwen2.5-VL-32B + UI-OCEANUS (CPT+SFT)	64.8	82.1
Qwen2.5-VL-32B + UI-OCEANUS (CPT+SFT+GRPO)	68.9	84.5

Appendix G.2. Online Evaluation Benchmark Comparison

Complementing the offline metrics, Table A5 presents the Success Rate (SR) on the online benchmark. We list the performance of **UI-OCEANUS** across three progressive post-training stages and compare them against representative external models, including state-of-the-art proprietary models.

Table A5. Online Success Rate vs. Proprietary Models. Comparison of Success Rate (SR) on live mini-programs between our progressive pipeline and external baselines.

Category	Model / Stage	SR (%)	Gap w/ Best Ours
<i>Ours</i> (32B)	UI-OCEANUS (SFT Cold Start)	29.5	-14.8
	+ Step-level GRPO	38.9	-5.4
	+ Multi-step GRPO	44.3	–
<i>Proprietary & External</i>	Seed-1.8 [55] (w/ Grounding)	27.5	-16.8
	Claude-4.5-Opus [53] (w/ Grounding)	46.3	+2.0
	Gemini-3-Flash [54] (w/ Grounding)	49.0	+4.7

References

1. Qin, Y.; Ye, Y.; Fang, J.; Wang, H.; Liang, S.; Tian, S.; Zhang, J.; Li, J.; Li, Y.; Huang, S.; et al. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326* **2025**.
2. Wang, H.; Zou, H.; Song, H.; Feng, J.; Fang, J.; Lu, J.; Liu, L.; Luo, Q.; Liang, S.; Huang, S.; et al. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. *arXiv preprint arXiv:2509.02544* **2025**.
3. Wu, Z.; Wu, Z.; Xu, F.; Wang, Y.; Sun, Q.; Jia, C.; Cheng, K.; Ding, Z.; Chen, L.; Liang, P.P.; et al. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218* **2024**.
4. Xu, Y.; Wang, Z.; Wang, J.; Lu, D.; Xie, T.; Saha, A.; Sahoo, D.; Yu, T.; Xiong, C. Aguviz: Unified Pure Vision Agents for Autonomous GUI Interaction. In Proceedings of the ICML, 2025.
5. Hong, W.; Wang, W.; Lv, Q.; Xu, J.; Yu, W.; Ji, J.; Wang, Y.; Wang, Z.; Dong, Y.; Ding, M.; et al. Cogagent: A visual language model for GUI agents. In Proceedings of the CVPR, 2024, pp. 14281–14290.
6. Torabi, F.; Warnell, G.; Stone, P. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954* **2018**.
7. Rawles, C.; Li, A.; Rodriguez, D.; Riva, O.; Lillicrap, T. Androidinthewild: A large-scale dataset for Android device control. *NeurIPS* **2023**, 36, 59708–59728.

8. Zhang, J.; Wu, J.; Teng, Y.; Liao, M.; Xu, N.; Xiao, X.; Wei, Z.; Tang, D. Android in the zoo: Chain-of-action-thought for GUI agents. *arXiv preprint arXiv:2403.02713* **2024**.
9. Lu, Q.; Shao, W.; Liu, Z.; Meng, F.; Li, B.; Chen, B.; Huang, S.; Zhang, K.; Qiao, Y.; Luo, P. GUI odyssey: A comprehensive dataset for cross-app GUI navigation on mobile devices. *arXiv preprint arXiv:2406.08451* **2024**.
10. Chen, W.; Cui, J.; Hu, J.; Qin, Y.; Fang, J.; Zhao, Y.; Wang, C.; Liu, J.; Chen, G.; Huo, Y.; et al. Guicourse: From general vision language models to versatile GUI agents. *arXiv preprint arXiv:2406.11317* **2024**.
11. Zhang, D.; Zhang, S.; Yang, Z.; Zhu, Z.; Zhao, Z.; Cao, R.; Chen, L.; Yu, K. ProgRM: Build Better GUI Agents with Progress Rewards. *arXiv preprint arXiv:2505.18121* **2025**.
12. Li, W.; Bishop, W.E.; Li, A.; Rawles, C.; Campbell-Ajala, F.; Tyamagundlu, D.; Riva, O. On the effects of data scale on UI control agents. *NeurIPS* **2024**, 37, 92130–92154.
13. Tang, L.; Dong, S.; Huang, Y.; Xiang, M.; Ruan, H.; Wang, B.; Li, S.; Xi, Z.; Cao, Z.; Pang, H.; et al. Magicgui: A foundational mobile gui agent with scalable data pipeline and reinforcement fine-tuning. *arXiv preprint arXiv:2508.03700* **2025**.
14. Li, Y.; Li, G.; He, L.; Zheng, J.; Li, H.; Guan, Z. Widget captioning: Generating natural language description for mobile user interface elements. *arXiv preprint arXiv:2010.04295* **2020**.
15. Cheng, K.; Sun, Q.; Chu, Y.; Xu, F.; Li, Y.; Zhang, J.; Wu, Z. Seeclck: Harnessing GUI grounding for advanced visual GUI agents. *arXiv preprint arXiv:2401.10935* **2024**.
16. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *International journal of computer vision* **2021**, 129, 1789–1819.
17. Wu, M.; Wang, H.; Ren, J.; Cao, Y.; Li, Y.; Jiang, A.; Ran, D.; Hu, Y.; Yang, W.; Xie, T. Skill-Adaptive Imitation Learning for UI Test Reuse. *arXiv preprint arXiv:2409.13311* **2024**.
18. Ran, D.; Wang, H.; Wang, W.; Xie, T. Badge: prioritizing UI events with hierarchical multi-armed bandits for automated UI testing. In *Proceedings of the ICSE, 2023*, pp. 894–905.
19. Gu, T.; Sun, C.; Ma, X.; Cao, C.; Xu, C.; Yao, Y.; Zhang, Q.; Lu, J.; Su, Z. Practical GUI testing of Android applications via model abstraction and refinement. In *Proceedings of the ICSE, 2019*, pp. 269–280.
20. Google. Android Monkey. <https://developer.android.com/studio/test/monkey>, 2021. Accessed: 2026-01-03.
21. Su, T.; Meng, G.; Chen, Y.; Wu, K.; Yang, W.; Yao, Y.; Pu, G.; Liu, Y.; Su, Z. Guided, stochastic model-based GUI testing of Android apps. In *Proceedings of the Proceedings of the 2017 11th joint meeting on foundations of software engineering, 2017*, pp. 245–256.
22. Ha, D.; Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122* **2018**, 2.
23. Hafner, D.; Lillicrap, T.; Ba, J.; Norouzi, M. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603* **2019**.
24. Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300* **2024**.
25. Wang, S.; Liu, W.; Chen, J.; Zhou, Y.; Gan, W.; Zeng, X.; Che, Y.; Yu, S.; Hao, X.; Shao, K.; et al. Gui agents with foundation models: A comprehensive survey. *arXiv preprint arXiv:2411.04890* **2024**.
26. Lin, K.Q.; Li, L.; Gao, D.; Yang, Z.; Wu, S.; Bai, Z.; Lei, S.W.; Wang, L.; Shou, M.Z. Showui: One vision-language-action model for gui visual agent. In *Proceedings of the Proceedings of the Computer Vision and Pattern Recognition Conference, 2025*, pp. 19498–19508.
27. Ye, J.; Zhang, X.; Xu, H.; Liu, H.; Wang, J.; Zhu, Z.; Zheng, Z.; Gao, F.; Cao, J.; Lu, Z.; et al. Mobile-Agent-v3: Fundamental Agents for GUI Automation. *arXiv preprint arXiv:2508.15144* **2025**.
28. Xu, Y.; Liu, X.; Liu, X.; Fu, J.; Zhang, H.; Jing, B.; Zhang, S.; Wang, Y.; Zhao, W.; Dong, Y. MobileRL: Online Agentic Reinforcement Learning for Mobile GUI Agents, 2025, [[arXiv:cs.LG/2509.18119](https://arxiv.org/abs/2509.18119)].
29. Zhang, C.; Neubig, G.; Yue, X. On the Interplay of Pre-Training, Mid-Training, and RL on Reasoning Language Models, 2025, [[arXiv:cs.CL/2512.07783](https://arxiv.org/abs/2512.07783)].
30. Yue, Y.; Chen, Z.; Lu, R.; Zhao, A.; Wang, Z.; Song, S.; Huang, G. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837* **2025**.
31. Lin, M.; Liu, M.; Lu, T.; Yuan, L.; Liu, Y.; Xu, H.; Miao, Y.; Chao, Y.; Li, Z. GUI-ReWalk: Massive Data Generation for GUI Agent via Stochastic Exploration and Intent-Aware Reasoning. *arXiv preprint arXiv:2509.15738* **2025**.
32. Sun, Q.; Cheng, K.; Ding, Z.; Jin, C.; Wang, Y.; Xu, F.; Wu, Z.; Jia, C.; Chen, L.; Liu, Z.; et al. Os-genesis: Automating gui agent trajectory construction via reverse task synthesis. In *Proceedings of the Proceedings*

- of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 5555–5579.
33. Ramrakhya, R.; Szot, A.; Attia, O.; Yang, Y.; Nguyen, A.; Mazouze, B.; Gan, Z.; Agrawal, H.; Toshev, A. Scaling synthetic task generation for agents via exploration. *arXiv preprint arXiv:2509.25047* 2025.
 34. Pahuja, V.; Lu, Y.; Rosset, C.; Gou, B.; Mitra, A.; Whitehead, S.; Su, Y.; Hassan, A. Explorer: Scaling exploration-driven web trajectory synthesis for multimodal web agents. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 6300–6323.
 35. Luo, D.; Tang, B.; Li, K.; Papoudakis, G.; Song, J.; Gong, S.; Hao, J.; Wang, J.; Shao, K. ViMo: A Generative Visual GUI World Model for App Agents. *arXiv preprint arXiv:2504.13936* 2025.
 36. Wang*, K.; Zhang*, P.; Wang*, Z.; Gao*, Y.; Li*, L.; Wang, Q.; Chen, H.; Wan, C.; Lu, Y.; Yang, Z.; et al. VAGEN: Reinforcing World Model Reasoning for Multi-Turn VLM Agents, 2025.
 37. Gao, L.; Zhang, L.; Xu, M. UIShift: Enhancing VLM-based GUI Agents through Self-supervised Reinforcement Learning. *arXiv preprint arXiv:2505.12493* 2025.
 38. Bai, S.; Cai, Y.; Chen, R.; Chen, K.; Chen, X.; Cheng, Z.; Deng, L.; Ding, W.; Gao, C.; Ge, C.; et al. Qwen3-VL Technical Report 2025. [[arXiv:cs.CV/2511.21631](https://arxiv.org/abs/cs.CV/2511.21631)].
 39. Zhang, Y.; Ni, B.; Chen, X.S.; Zhang, H.R.; Rao, Y.; Peng, H.; Lu, Q.; Hu, H.; Guo, M.H.; Hu, S.M. Bee: A High-Quality Corpus and Full-Stack Suite to Unlock Advanced Fully Open MLLMs. *arXiv preprint arXiv:2510.13795* 2025.
 40. Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923* 2025.
 41. Li, S.; Kallidromitis, K.; Gokul, A.; Kato, Y.; Kozuka, K.; Grover, A. MobileWorldBench: Towards Semantic World Modeling For Mobile Agents. *arXiv preprint arXiv:2512.14014* 2025.
 42. Velasco, D.J.; Roque, M.T. Rethinking the role of text complexity in language model pretraining. In Proceedings of the Proceedings of the First BabyLM Workshop, 2025, pp. 1–28.
 43. Agrawal, A.; Singh, S. Corpus complexity matters in pretraining language models. In Proceedings of the Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP), 2023, pp. 257–263.
 44. Marion, M.; Üstün, A.; Pozzobon, L.; Wang, A.; Fadaee, M.; Hooker, S. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564* 2023.
 45. Geirhos, R.; Jacobsen, J.H.; Michaelis, C.; Zemel, R.; Brendel, W.; Bethge, M.; Wichmann, F.A. Shortcut learning in deep neural networks. *Nature Machine Intelligence* 2020, 2, 665–673.
 46. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D.; et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 2022, 35, 24824–24837.
 47. Hafner, D.; Pasukonis, J.; Ba, J.; Lillicrap, T. Mastering diverse control tasks through world models. *Nature* 2025, pp. 1–7.
 48. Assran, M.; Duval, Q.; Misra, I.; Bojanowski, P.; Vincent, P.; Rabbat, M.; LeCun, Y.; Ballas, N. Self-supervised learning from images with a joint-embedding predictive architecture. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 15619–15629.
 49. LeCun, Y. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review* 2022, 62, 1–62.
 50. Zhang, K.; Chen, X.; Liu, B.; Xue, T.; Liao, Z.; Liu, Z.; Wang, X.; Ning, Y.; Chen, Z.; Fu, X.; et al. Agent learning via early experience. *arXiv preprint arXiv:2510.08558* 2025.
 51. OpenAI. Update to GPT-5 System Card: GPT-5.2. Technical report, OpenAI, 2025. Accessed: 2026-01-29.
 52. Anthropic. Claude Sonnet 4.5, 2025. Accessed: 2026-01-29.
 53. Anthropic. Claude Opus 4.5, 2025. Accessed: 2026-01-29.
 54. Google. Gemini 3 Flash, 2025. Accessed: 2026-01-29.
 55. ByteDance-Seed. Seed1.8 Model Card: Towards Generalized Real-World Agency, 2025. Accessed: 2026-01-29.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.