WorldAgen: Unified State-Action Prediction with Test-Time World Model Training

Anonymous Author(s)
Affiliation
Address
email

Abstract

How can vision-language-action (VLA) models adapt to new environments where world dynamics shift? While recent research has combined world modeling and action prediction to improve VLA performance, existing methods largely rely on pretraining in static datasets, without mechanisms for active adaptation to new environments. As a result, these models often fail to generalize when deployed in unseen scenarios with novel object configurations or dynamics. We present **WorldAgen**, a unified framework that jointly learns world modeling and action prediction while enabling **test-time training (TTT)** to adapt to new environments. WorldAgen employs a shared Transformer backbone with two heads: (1) a **world-model** head that predicts future states from past state-action trajectories, and (2) an **agent-model** head that predicts actions conditioned on task instructions. During test time, WorldAgen samples exploratory actions, collects ground-truth state transitions, and performs lightweight TTT updates to refine its world model. This adaptation improves the model's understanding to the environments and leads to more accurate action predictions. Experiments on the CALVIN and LIBERO benchmarks demonstrate that our baseline model achieves comparable, and in some cases superior, performance to current state-of-the-art approaches. Moreover, with TTT on a small number of samples, our method surpasses existing state-of-the-art models, highlighting effectiveness of adapting world models at inference time.

1 Introduction

Vision-language-action (VLA) models have emerged as a powerful and popular paradigm for robotic manipulation [1, 2], enabling agents to follow natural language instructions and act directly from raw visual observations. Recent work has explored joint state-action prediction techniques [3–7], allowing models not only to predict the next actions but also to anticipate how their actions will transform the environment. This joint formulation has improved data efficiency and scene understanding, leading to stronger performance across manipulation benchmarks.

However, we argue that current methods remain fundamentally limited by training on static datasets [8–10]. Once pretrained, these models lack mechanisms to adapt their internal representations of world dynamics when deployed in novel environments. In realistic settings, distribution shifts such as new object layouts, lighting conditions, or physical properties are inevitable, and static world modeling fails to capture these variations [11, 12]. This leads to a key question: **How can we enable VLA models to actively adapt their understanding of the environment during test time?**

As shown in Figure 1, we address this question with WorldAgen, a unified framework that combines joint state-action prediction with a novel test-time training (TTT) strategy:

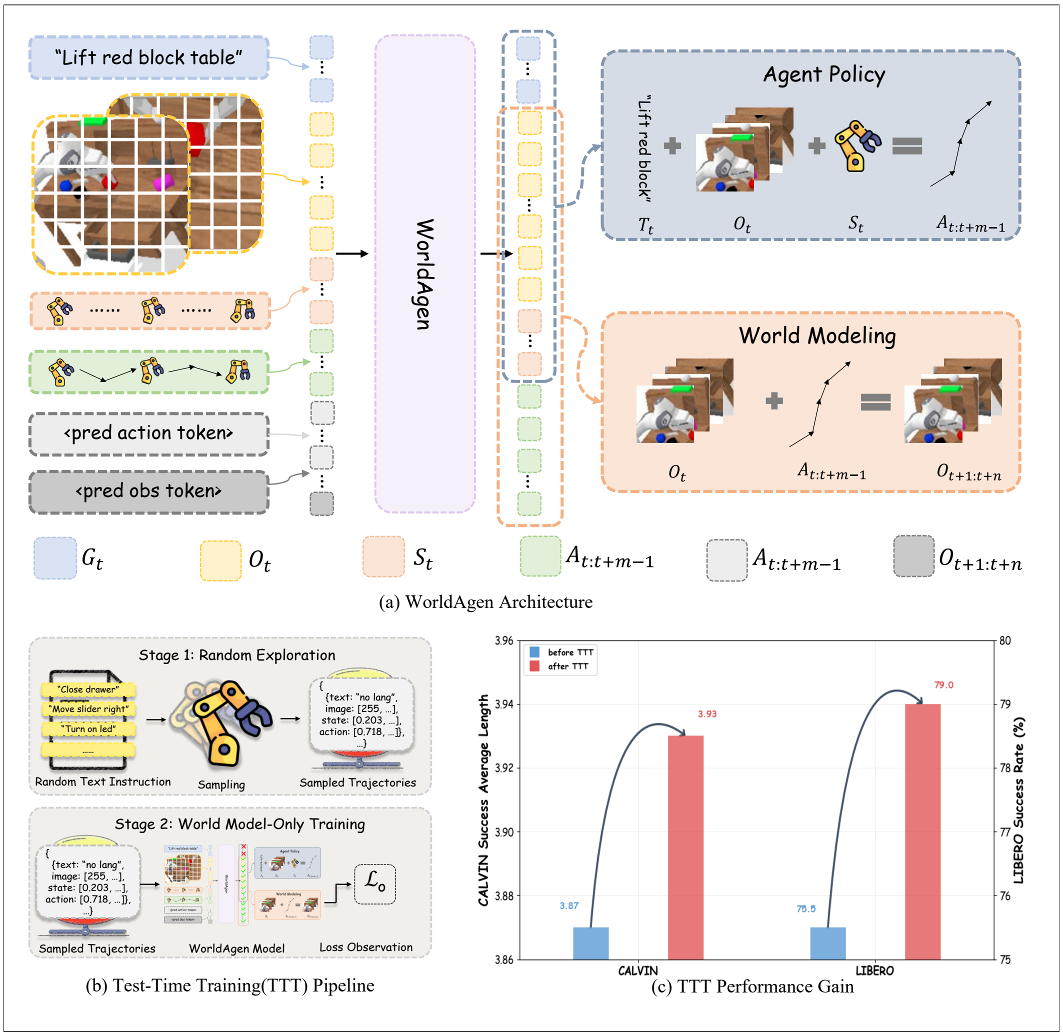


Figure 1: Overview of WorldAgen. (a) **WorldAgen Architecture.** WorldAgen unifies an *agent policy* head for task-conditioned action prediction and a *world modeling* head for task-agnostic state prediction within a single Transformer backbone. At timestep t, the agent receives a goal instruction G_t, observation O_t, and robot state S_t, and predicts an action chunk $A_{t:t+m-1}$. In parallel, the world model predicts the future observation chunk $O_{t+1:t+n}$ given O_t and A_t, refining the shared representation of environment dynamics. (b) **Two-stage Test-Time Training (TTT) pipeline.** Stage 1 performs random exploration in the new environment to collect unlabeled trajectories. Stage 2 adapts the world model using LoRA-based fine-tuning on the observation loss $\mathcal{L}_o$, improving environment modeling without modifying the agent policy. (c) **TTT Performance** We tested WorldAgen and WorldAgen-TTT on CALVIN and LIBERO benchmark, which show great performance gain.

- **Joint State-Action Modeling.** WorldAgen uses a shared Transformer backbone with two heads. The world-model head predicts future states from historical state-action trajectories, while the agent-model head predicts actions conditioned on task instructions. By training these tasks jointly, WorldAgen aligns scene understanding with control.
- **Test-Time Training for Scene Adaptation.** During test time, WorldAgen executes exploratory actions to collect ground-truth state transitions. It then performs lightweight, LoRA-based TTT updates to refine its world model. These updates improve the internal representation of the environment, indirectly boosting the agent's ability to generate effective actions in novel scenarios, which is depicted in.

Our work reframes world modeling from a passive pretraining objective into an active test time adaptation mechanism, bridging the gap between offline training and real-world generalization.

In sum, we make the following contributions:

- **Unified joint state-action prediction architecture.** We present a single Transformer backbone that integrates world modeling and action prediction, enabling shared representations and tighter coupling between perception and control.
- **Active test-time adaptation for VLA models.** We introduce a TTT paradigm that transforms world modeling from a static pretraining into an active, test time adaptation mechanism, allowing VLA models to refine their scene understanding on the fly.
- **Empirical validation across challenging benchmarks.** Our baseline achieves performance comparable to, or even better than, state-of-the-art methods. Furthermore, by fine-tuning the world modeling component with only a small number of samples at test time, our method attains state-of-the-art results on both the CALVIN and LIBERO benchmark.

We believe these results demonstrate that continuous adaptation during test time, rather than merely scaling model size or pretraining data, is a key ingredient for robust and generalizable robotic manipulation, and pave the way for future research on adaptive and lifelong VLA models.

2 Method

In this section, we introduce **WorldAgen**, a unified VLA framework that integrates (1) an agent model for task-conditioned action prediction and (2) a world model for task-agnostic environment dynamics prediction. Both components share a single Transformer backbone and are further enhanced by a two-stage test-time training (TTT) strategy that enables online adaptation to new environments.

2.1 Task Formulation

We frame our task as a partially observable Markov decision process (POMDP). At each timestep t, the agent interacts with the environment based on two inputs: a partial observation O_t of the environment and a task instruction G_t. The agent predicts an action chunk $A_{t:t+m-1}$ of length m, which is then executed by the environment. After execution, the environment transitions forward m steps, producing new observations $O_{t+1:t+m}$ and updating the underlying environment state $S_{t+1:t+m}$, where each O_t is a partial observation of the corresponding state S_t.

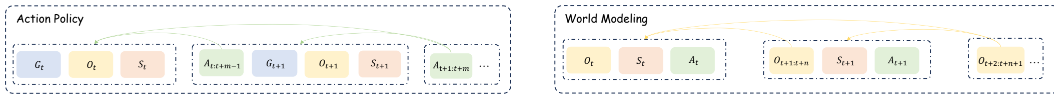
We also maintain two types of histories:

- **Agent history:** $h_t^a = (O_{t-k:t-1}, G_{t-k:t-1}, A_{t-k:t-1})$, containing the last k observations, task goals, and actions, used by the agent model for task-aware action prediction.
- **World history:** $h_t^w = (O_{t-k:t-1}, A_{t-k:t-1})$, containing the last k observations and actions, used by the world model to predict task-agnostic environment dynamics.

With these definitions, our two predictive processes are:

$$p_a(A_{t:t+m-1} \mid O_t, G_t, h_t^a), \qquad p_w(O_{t+1:t+n} \mid O_t, A_t, h_t^w),$$

where p_a is the agent model and p_w is the world model, m and n are the action and observation chunk lengths. WorldAgen jointly optimizes these two objectives, allowing the agent to learn task-aware policies while grounding them in an explicit model of environment dynamics.



(a) Agent model. (b) World model.

Figure 2: **Joint architecture of WorldAgen.** WorldAgen integrates an agent policy head for action prediction and a world modeling head for environment dynamics prediction within a shared Transformer backbone. The agent model is task-conditioned, while the world model is task-agnostic, enabling the two to share representations and reinforce each other.

2.2 Joint State-Action Prediction

WorldAgen has two predictive components: a task-conditioned agent model and a task-agnostic world model, as shown in Figure 2. The two components share a single Transformer backbone for a unified representation between both tasks.

Agent Model. The agent model predicts the next m actions based on the current observation O_t, the task instruction G_t, and the agent history h_t^a:

$$p(A_{t:t+m-1} \mid O_t, G_t, h_t^a),$$

where $h_t^a = (O_{t-k:t-1}, G_{t-k:t-1}, A_{t-k:t-1})$ contains the past k observations, task goals, and actions.

World Model. The world model predicts how the environment evolves in response to the agent's actions. It generates the next n observations given the current observation O_t, the executed action A_t, and the world history h_t^w:

$$p(O_{t+1:t+n} \mid O_t, A_t, h_t^w),$$

where $h_t^w = (O_{t-k:t-1}, A_{t-k:t-1})$ contains only past observations and actions, making the world model task-agnostic.

2.3 Mixed Unidirectional Attention Mask

To ensure that the world modeling and agent policy tasks are trained jointly without information leakage, we introduce a mixed unidirectional attention masking mechanism as shown in Figure 3. This mechanism integrates two complementary components: (1) a local mask that isolates the two tasks at each time step, and (2) a global mask that enforces temporal causality.

Local Mask. The local mask is used within each time step to prevent cross-task information leakage between the world modeling and agent policy heads, which ensures:

- The world modeling head cannot attend to the action tokens it is supposed to predict.
- The agent policy head cannot attend to the observation tokens it is supposed to predict.

This task-level separation allows the two prediction heads to share the Transformer backbone while maintaining independence in their outputs.

Global Mask. The global mask enforces temporal causality across time for both tasks. For a sequence of tokens, the global mask matrix M^{global} is defined as:

$$M_{ij}^{\text{global}} = \begin{cases} 0, & j \le i \\ -\infty, & j > i, \end{cases}$$

which is added to the attention logits before the softmax operation:

$$\text{Attn}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d}} + M^{\text{global}}\right) V.$$

This prevents both the world modeling and agent policy heads from attending to future tokens, ensuring strictly causal prediction.

By combining the local mask (task separation within each time step) and the global mask (temporal causality), we make sure clean multi-task training in a shared Transformer backbone while preserving the correct conditioning structure for each task.

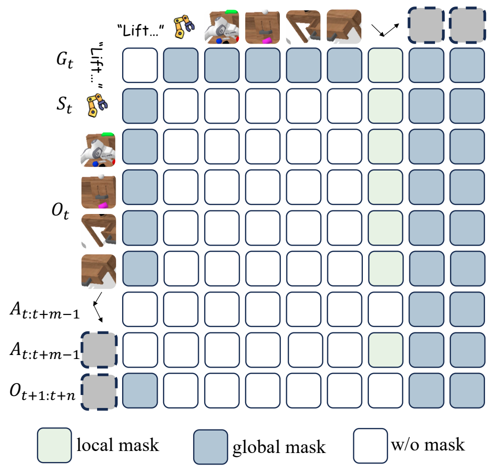


Figure 3: **Mixed unidirectional attention masking.** WorldAgen applies: (1) **local masks** at each time step to prevent cross-task information leakage between world modeling and agent policy, and (2) **global masks** to enforce temporal causality across time.

2.4 Pretraining

We pretrain WorldAgen using large-scale trajectory data collected from robot demonstrations. Each trajectory consists of observations O_t, actions A_t, robot states S_t, and language instructions G_t. During pretraining, we jointly optimize the agent model and world model using teacher forcing:

$$\mathcal{L} = \mathcal{L}_a + \lambda \mathcal{L}_o,$$

where $\mathcal{L}_a$ is the cross-entropy loss for predicting future action chunks, $\mathcal{L}_o$ is the reconstruction loss for predicting future observation chunks, and λ is a weighting factor balancing the two objectives.

This pretraining stage enables the agent model to learn task-conditioned action generation and the world model to learn task-agnostic environment dynamics, all within a shared Transformer backbone.

2.5 Test-Time Training (TTT)

While joint pretraining improves world understanding, distribution shifts in novel environments can still degrade performance. To address this, we introduce a two-stage test-time training (TTT) strategy that adapts the world model online, refining its understanding of environment dynamics and indirectly improving downstream action prediction. The whole pipeline is given in Figure 1.

Stage 1: Random Exploration and Data Collection. At deployment, the agent first performs exploratory rollouts in the target environment. During this phase, it executes sampled action chunks and records trajectories of (O_t, S_t, A_t). To ensure that adaptation focuses on environment dynamics rather than task-specific instructions, all collected trajectories are relabeled with a generic "no-lang" token in place of language instructions.

Stage 2: World Model Adaptation. Using the collected trajectories, we adapt only the world modeling head with LoRA-based parameter-efficient fine-tuning, while keeping the agent policy head and shared backbone frozen. The update rule is:

$$\theta'_w \leftarrow \theta_w - \eta \nabla_{\theta_w} \mathcal{L}_o,$$

where θ_w are the parameters of the world model head and η is the learning rate.

This targeted update improves the predictive accuracy of the world model, which in turn enhances the shared representations leveraged by the agent model for action prediction. By decoupling adaptation from the task goal and restricting it to the world model, our TTT procedure enables robust scene adaptation without requiring additional task-specific annotations, paving the way for scalable test-time adaptation in VLA models.

Table 1: Performance comparison on CALVIN benchmark. We report the success rate (%) for completing 1 to 5 consecutive tasks and the average sequence length (Avg. Len.).

Method	T1	T2	T3	T4	T5	Avg. Len. ↑
RoboFlamingo	82.4	61.9	46.6	33.1	23.5	2.47
SuSIE	87.0	69.0	49.0	38.0	26.0	2.69
GR-1	85.4	71.2	59.6	49.7	40.1	3.06
3D Diffusor Actor	92.2	78.7	63.9	51.2	41.2	3.27
CLOVER	96.0	83.5	70.8	57.5	45.4	3.53
Seer	93.0	82.4	72.3	62.6	53.3	3.64
Seer-Large	92.7	84.6	76.1	68.9	60.3	3.83
WorldAgen	96.3	87.7	76.8	67.3	59.1	3.87
WorldAgen-TTT	**96.6**	**88.5**	**78.5**	**68.7**	**60.5**	**3.93**

3 Experiments

In this section, we conduct comprehensive experiments to evaluate the effectiveness of WorldAgen across CALVIN and LIBERO benchmarks. We first present the main results comparing our baseline model and the TTT-enhanced model. Then we perform detailed ablation studies examining the impact of image and action chunk length, LoRA parameter configurations, and the volume of TTT data.

3.1 Datasets

CALVIN [13] is an open-source simulated benchmark designed for learning long-horizon language-conditioned robot manipulation tasks. The benchmark requires agents to solve complex manipulation tasks by understanding a series of unconstrained language instructions in sequence.

LIBERO [14] is a comprehensive benchmark for lifelong learning in robot manipulation that emphasizes knowledge transfer across diverse tasks.

3.2 Implementation Details

Pretraining Given the powerful modeling capability and flexibility of Qwen3 [15], we adopt it as the network backbone for WorldAgen. Following the configuration established in [3], we configure the transformer architecture with 12 transformer heads and 24 transformer layers, resulting in a total network size of 370M parameters with 120M trainable parameters.

On CALVIN benchmark, we employ an image chunk length of 1, action chunk length of 5, and trajectory length of 16. For LIBERO, we use an image chunk length of 1, action chunk length of 3, and trajectory length of 7. More details are shown in the subsequent section. Model pretraining is performed on a 4×H100 GPU server, taking approximately 60 hours for CALVIN and 5 hours for LIBERO.

Test-Time Training Test-time training represents a crucial component of WorldAgen that enables adaptive performance improvements during inference. In the TTT phase, we apply LoRA fine-tuning to the backbone of Qwen3. Specifically, we apply LoRA to the attention projection layers (q_proj, k_proj, v_proj, o_proj) and the MLP projection layers (gate_proj, up_proj, down_proj) of Qwen3.

For **CALVIN**, since the test scenarios are relatively uniform within each environment, we do not need to perform TTT for every individual sample. Instead, we conduct environment-level adaptation by sampling only on the first sample's scenario. We select 34 text instructions that have appeared in the training set as guidance and perform random exploration for 60 frames. We then uniformly sample 6 times within these 60 frames, generating 204 samples for TTT. Following [16], we use single-epoch single-step optimization with LoRA rank set to 128, employing the AdamW optimizer with a learning rate of 0.005 and weight decay of 0.01.

For **LIBERO**, the evaluation involves 10 different test scenarios in the LIBERO-10 dataset, necessitating a different TTT strategy. We perform scene-level TTT sampling in each individual test scenario to adapt to the specific environmental conditions. Similar to CALVIN, we set LoRA rank to 64,

Table 2: Performance comparison on LIBERO benchmark. We report the success rate (%) across different manipulation tasks. Our method **WorldAgen** corresponds to the Seer model.

Method	Avg. Success ↑	Put soup and box in basket	Put box and butter in basket	Turn on stove and put pot	Put bowl in drawer and close it	Put mugs on left and right plates	Pick book and place it in back	Put mug on plate and put pudding to right	Put soup and sauce in basket	Put both pots on stove	Put mug in microwave and close it
MT-ACT	41.0	30.0	50.0	75.0	85.0	20.0	75.0	0.0	0.0	10.0	65.0
MVP	68.2	83.3	**90.0**	80.0	88.3	46.7	63.3	45.0	78.3	**60.0**	46.7
MPL	77.3	66.6	86.6	96.6	95.0	83.3	83.3	56.6	86.6	40.0	78.3
OpenVLA	54.0	35.0	95.0	65.0	45.0	40.0	80.0	60.0	45.0	20.0	55.0
Seer	78.7	80.0	**90.0**	91.7	81.7	85.0	65.0	**86.7**	88.3	51.7	**66.7**
WorldAgen	75.5	70.0	75.0	95.0	100	85.0	90.0	60.0	100	45.0	35.0
WorldAgen-TTT	**79.0**	**85.0**	75.0	**95.0**	**100**	**90.0**	**90.0**	50.0	**100**	45.0	60.0

explore for 60 frames and uniformly sample for 6 times, which generates 36 samples per scenario. Also we employ single-epoch single-step optimization using the AdamW optimizer with a learning rate of 0.005 and weight decay of 0.01.

TTT, including sampling and LoRA adaptation, can be executed on an RTX 4090 GPU and require approximately 8 minutes per task.

3.3 Results

Our experimental results demonstrate the effectiveness of WorldAgen across both CALVIN and LIBERO benchmarks.

CALVIN As shown in Table 1, we compare WorldAgen with several SOTA methods. Prior works include models that fine-tune large vision-language backbones with policy heads for manipulation (**RoboFlamingo** [17], **GR-1** [18], **Seer** [3]), or leverage diffusion-based planners and action learners, either through image editing or 3D scene understanding (**SuSIE** [19], **3D Diffusor Actor** [20], **CLOVER** [21]).

Our method achieves consistent performance improvements across five consecutive tasks. This indicates that

TTT enhances the world modeling capability of VLA models, thereby improving their understanding of the environment and boosting performance in long-horizon robotic manipulation tasks.

LIBERO As shown in Table 2, we compare WorldAgen with several SOTA baselines. Prior methods include multi-modal transformer architectures that combine vision, proprioception, and instructions for policy learning (**MT-ACT** [22], **Seer** [3]), as well as pretraining or predictive approaches that leverage large-scale visual data (**MVP** [23], **MPI** [24], **OpenVLA** [10]).

We observe improvements or maintained performance across almost every task compared to the baseline. These consistent gains across diverse manipulation scenarios mirror our findings on CALVIN, demonstrating ***robustness and generalization ability*** of our TTT approach across different benchmarks and scenario dynamics.

3.4 Ablation Study

To systematically assess the components underlying the effectiveness of **WorldAgen**, we design a series of ablation studies across six critical dimensions: (1) the impact of image and action chunk lengths on representation learning, (2) the contribution of world modeling to downstream policy optimization, (3) the role of LoRA parameterization during the test-time training (TTT) phase, (4) the effect of random-sampling data volume on TTT stability and performance, (5) the trajectory length, measured as the number of chunks, and (6) a comparison between lightweight LoRA adaptation and full finetuning strategies. Our primary analyses are carried out on the CALVIN benchmark owing to its standardized evaluation protocol, with selected results validated on LIBERO for generality. For clarity of exposition, results pertaining to (1) and (2) are reported in the main text, whereas ablations on (3)–(6) are presented in the Appendix.

3.4.1 Image and Action Chunk Length

WorldAgen offers flexible input-output interfaces, accommodating image and action chunks of variable lengths. We evaluate the effect of different image and action chunk lengths on model performance. The results are shown in Table 3.

Table 3: Ablation study on image and action chunk lengths across CALVIN and LIBERO benchmarks. I. Chunk and A. Chunk represent image chunk length and action chunk length, respectively

Dataset	I. Chunk	A. Chunk	Avg. Success ↑
CALVIN	1	3	3.82
	1	**5**	**3.87**
	1	7	3.43
	1	9	3.16
	3	5	3.30
	5	5	1.78
LIBERO	**1**	**3**	**78.0%**
	1	5	74.5%
	1	7	47.0%

We first fix the image chunk length to 1, as adjacent image frames contain redundant information and predicting more images increases computational overhead, thereby reducing model efficiency. By varying the action chunk length, we observe that ***longer action sequences lead to degraded performance***. We attribute this to error accumulation in action predictions, where mistakes in early actions compound over longer sequences. However, excessively short action chunk lengths also result in performance degradation, as the network learns less information from the dataset.

On LIBERO, we obtain consistent results, with performance deteriorating as action chunk length increases. Based on these findings, we adopt action chunk lengths of 5 for CALVIN and 3 for LIBERO in our main experiments.

3.4.2 World Modeling Ability

To validate the contribution of world modeling to agent policy learning, we conduct an ablation study comparing models with and without image prediction capabilities. The results are presented in Table 4.

Table 4: Ablation study on world modeling. We compare models with and without image prediction to evaluate the contribution of world modeling to agent policy learning.

Dataset	Img. Pred. Head	Avg. Success ↑
CALVIN	×	2.96
	✓	**3.87**
LIBERO	×	46.5%
	✓	**78.0%**

Specifically, we control the presence of world modeling capability by determining whether to include image prediction tokens in the input. The results demonstrate that world modeling significantly enhances agent policy performance across both benchmarks. On CALVIN, incorporating image prediction improves the average success length from 2.961 to 3.87, representing a substantial 30.7% improvement. On LIBERO, the improvement is even more pronounced, with success rates increasing from 46.5% to 78.0%, a remarkable 67.7% gain. These findings confirm that

World modeling ability helps develop better internal representations of environment dynamics, which in turn facilitates more effective action prediction and policy learning.

4 Related Work

Vision-Language-Action Models. Vision Language Action (VLA) models unify perception, language, and control for robotic manipulation [2, 25]. Early works such as RT-1 [26] and RT-2 [27] demonstrated the potential of large-scale, transformer-based policies, while PaLM-E [28] and LLaVA [29] incorporated multimodal grounding. However, most VLA models focus on direct action prediction and neglect explicit modeling of world dynamics, limiting their ability to generalize to novel environments [30].

World Models. World models learn predictive dynamics to support planning and decision-making [31, 32]. Neural approaches such as World Models [33] and Dreamer [32, 34] have improved sample efficiency and long-horizon reasoning. Yet, in robotics, most world models are used for state prediction or model-based planning, largely decoupled from language-grounded action generation [35].

Test-Time Training. Test-time training (TTT) adapts models to distribution shifts during inference by leveraging self-supervised signals [36, 37]. While TTT has been explored in vision and NLP [38], robotic adaptation has mainly focused on perception modules or low-level policies [39]. Our work is the first to incorporate TTT into a unified VLA framework by adapting the world model itself during deployment.

5 Conclusion

In this work, we present **WorldAgen**, a unified vision-language-action framework that combines joint world modeling and action prediction with test-time training. The core innovation lies in WorldAgen's shared Transformer backbone architecture with dual specialized heads: a world-model head that predicts future states from past state-action trajectories, and an agent-model head that predicts actions conditioned on task instructions. During test time, WorldAgen samples exploratory actions, collects ground-truth state transitions, and performs lightweight TTT updates to refine its world model understanding. By transforming world modeling from a static pretraining objective into an active adaptation mechanism, WorldAgen improves environment understanding during deployment and achieves consistent gains on CALVIN and LIBERO benchmarks, surpassing state-of-the-art VLA models. With TTT adaptation using only a small number of samples, our method substantially outperforms existing state-of-the-art models, highlighting the effectiveness of adapting world models at inference time rather than relying solely on static pretraining. WorldAgen represents a paradigm shift toward adaptive VLA systems that can continuously improve their understanding of new environments through active exploration and learning, opening new directions for robust deployment of vision-language-action models in dynamic real-world scenarios.

6 Limitations and Future Work

While effective, our approach has several limitations. TTT adapts only the world model head and relies on random exploration, which may limit adaptation efficiency. Our evaluation is restricted to simulation benchmarks, and real-robot experiments remain unexplored. Future work includes developing task-aware or uncertainty-driven exploration strategies, enabling joint adaptation of both agent and world model heads, and validating WorldAgen on real robotic platforms to further enhance its generalization and deployment readiness.

References

[1] Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied ai, 2025. URL https://arxiv.org/abs/2405.14093.

[2] Muhayy Ud Din, Waseem Akram, Lyes Saad Saoud, Jan Rosell, and Irfan Hussain. Vision language action models in robotic manipulation: A systematic review, 2025. URL https://arxiv.org/abs/2507.10672.

[3] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation, 2024. URL https://arxiv.org/abs/2412.15109.

[4] Yanjiang Guo, Yucheng Hu, Jianke Zhang, Yen-Jen Wang, Xiaoyu Chen, Chaochao Lu, and Jianyu Chen. Prediction with action: Visual policy learning via joint denoising process. *Advances in Neural Information Processing Systems*, 37:112386–112410, 2024.

[5] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation, 2025. URL https://arxiv.org/abs/2410.07864.

[6] Zhou Xian and Nikolaos Gkanatsios. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *Conference on Robot Learning/Proceedings of Machine Learning Research*. Proceedings of Machine Learning Research, 2023.

[7] Rosa Wolf, Yitian Shi, Sheng Liu, and Rania Rayyes. Diffusion models for robotic manipulation: A survey, 2025. URL https://arxiv.org/abs/2504.08438.

[8] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025. URL https://arxiv.org/abs/2502.19645.

[9] Puhao Li, Yingying Wu, Ziheng Xi, Wanlin Li, Yuzhe Huang, Zhiyuan Zhang, Yinghan Chen, Jianan Wang, Song-Chun Zhu, Tengyu Liu, and Siyuan Huang. Controlvla: Few-shot object-centric adaptation for pre-trained vision-language-action models, 2025. URL https://arxiv.org/abs/2506.16211.

[10] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

[11] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pages 2226–2240. PMLR, 2023.

[12] JB Lanier, Kyungmin Kim, Armin Karamzade, Yifei Liu, Ankita Sinha, Kat He, Davide Corsi, and Roy Fox. Adapting world models with latent-state dynamics residuals, 2025. URL https://arxiv.org/abs/2504.02252.

[13] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7327–7334, 2022.

[14] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[15] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui

Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

[16] Yuto Kojima, Jiarui Xu, Xueyan Zou, and Xiaolong Wang. Lora-ttt: Low-rank test-time training for vision-language models, 2025. URL `https://arxiv.org/abs/2502.02069`.

[17] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023.

[18] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023.

[19] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

[20] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024.

[21] Qingwen Bu, Jia Zeng, Li Chen, Yanchao Yang, Guyue Zhou, Junchi Yan, Ping Luo, Heming Cui, Yi Ma, and Hongyang Li. Closed-loop visuomotor control with generative expectation for robotic manipulation. *Advances in Neural Information Processing Systems*, 37:139002–139029, 2024.

[22] Siddhant Haldar, Zhuoran Peng, and Lerrel Pinto. Baku: An efficient transformer for multi-task policy learning. *Advances in Neural Information Processing Systems*, 37:141208–141239, 2024.

[23] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[24] Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang, Di Hu, Ping Luo, et al. Learning manipulation by predicting interaction. *arXiv preprint arXiv:2406.00439*, 2024.

[25] Ranjan Sapkota, Yang Cao, Konstantinos I. Roumeliotis, and Manoj Karkee. Vision-language-action models: Concepts, progress, applications and challenges, 2025. URL `https://arxiv.org/abs/2505.04769`.

[26] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023. URL `https://arxiv.org/abs/2212.06817`.

[27] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

[28] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. Palm-e: An embodied multimodal language model. 2023.

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

[30] Ji Zhang, Shihan Wu, Xu Luo, Hao Wu, Lianli Gao, Heng Tao Shen, and Jingkuan Song. Inspire: Vision-language-action models with intrinsic spatial reasoning, 2025. URL https://arxiv.org/abs/2505.13888.

[31] Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pages 216–224. Elsevier, 1990.

[32] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, pages 1–7, 2025.

[33] David Ha and Jürgen Schmidhuber. World models. *CoRR*, 2018.

[34] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models, 2022. URL https://arxiv.org/abs/2010.02193.

[35] Ryo Sakagami, Florian S Lay, Andreas Dömel, Martin J Schuster, Alin Albu-Schäffer, and Freek Stulp. Robotic world models—conceptualization, review, and engineering best practices. *Frontiers in Robotics and AI*, 10:1253049, 2023.

[36] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts, 2020. URL https://arxiv.org/abs/1909.13231.

[37] Jing Ma. Improved self-training for test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23701–23710, 2024.

[38] Hai Ye, Yuyang Ding, Juntao Li, and Hwee Tou Ng. Robust question answering against distribution shifts with test-time adaptation: An empirical study, 2023. URL https://arxiv.org/abs/2302.04618.

[39] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, pages 737–744. IEEE, 2020.

[40] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1 (2):3, 2022.

[41] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[42] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.

Appendix

This appendix provides comprehensive supplementary materials for our WorldAgen framework. We present detailed implementation specifications including benchmark descriptions (LIBERO and CALVIN), model architecture components (encoder, decoder, and Qwen3 backbone), and trajectory processing methods for variable-length chunks. Additionally, we report extensive experimental results examining the effects of different chunk configurations on baseline performance, and comprehensive Test-Time Training (TTT) analyses comparing LoRA [40] versus full fine-tuning approaches across various parameter settings and data sizes.

A Implementation Details

A.1 Benchmark

LIBERO Benchmark [14] is a comprehensive benchmark for lifelong learning in robot manipulation that consists of four distinct task suites designed to evaluate different aspects of knowledge transfer. The benchmark includes LIBERO-Spatial (10 tasks focusing on spatial relationship transfer), LIBERO-Object (10 tasks emphasizing object-centric knowledge transfer), LIBERO-Goal (10 tasks targeting procedural knowledge generalization), and LIBERO-100 (100 tasks with highly entangled knowledge requirements). Each task suite is accompanied by high-quality human teleoperation demonstrations to support sample-efficient learning.

We use LIBERO-100, which is the most challenging and comprehensive subset, containing 100 diverse manipulation tasks that require the transfer of mixed declarative and procedural knowledge. This suite is further divided into LIBERO-90, consisting of 90 short-horizon tasks used for pretraining policies, and LIBERO-10 (also referred to as LIBERO-Long), which contains 10 long-horizon tasks specifically selected for evaluating downstream lifelong learning performance.

CALVIN Benchmark [13] is an open-source simulated benchmark designed for learning long-horizon language-conditioned robot manipulation tasks. The dataset contains 34 manipulation tasks that are more complex than existing vision-and-language datasets in terms of sequence length, action space, and language complexity. The environment features a Franka Emika Panda robot with a parallel-jaw gripper operating in a desktop workspace containing various interactive objects, including a sliding door, drawer, colored blocks, LED, and light bulb that can be manipulated according to unconstrained natural language instructions.

CALVIN provides four different environments (A, B, C, D) that vary in desk colors and object configurations, enabling evaluation across different visual contexts and supporting flexible specification of sensor suites. The benchmark's evaluation protocol requires agents to solve sequences of up to five consecutive tasks by understanding and executing a series of language instructions, such as "open the drawer... pick up the blue block... now push the block into the drawer... now open the sliding door." This sequential task completion paradigm makes CALVIN particularly challenging for assessing the generalization capabilities and long-horizon reasoning of language-conditioned manipulation policies.

A.2 Model Architecture

Leveraging the flexible architecture of Transformers, we achieve unified processing of image, language, action, and robot state modalities, supporting variable-length image chunks and action chunks for both input and output.

Encoder In the encoder component, we employ a MAE-pretrained ViT-B [41] as the visual encoder, utilizing dual-view RGB inputs from static and wrist cameras. Similar to Seer [3], we incorporate a Perceiver Resampler [42] to reduce the number of image tokens, thereby decreasing computational load and improving efficiency. The Perceiver Resampler is an attention-based feature compression module that uses a set of learnable query tokens to extract the most important information from a large number of image features. Through this approach, it compresses the originally large number of image tokens into a fixed number of compact representations, preserving essential visual information while significantly reducing the computational complexity of subsequent network layers.

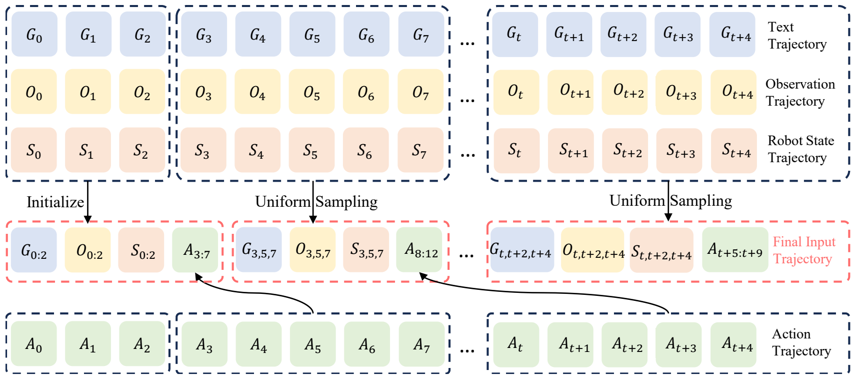


Figure 4: This figure shows how we design the input trajectory of WorldAgen and split an original robot trajectory data to satisfy the variable image and action chunk length

For language processing, we utilize the text encoder from CLIP ViT-B/32. For robot state and action, we employ MLPs to project them into the same embedding space. Both robot state and action are represented as 7-dimensional vectors, where the first 6 dimensions encode the arm state representing the end effector's position and orientation, and the last dimension represents the gripper state indicating the open/close status. We use separate MLP layers to process arm state and gripper state independently, then concatenate the results.

We initialize predicted tokens as zero-filled tensors with the same dimensions as the target actions and images to be predicted, and insert them after the input data at each time step.

Decoder In the decoder, we index the predicted tokens based on the input and slice out the image and action tokens. For image tokens, we employ the Vision Transformer encoder architecture [41] for decoding, followed by a linear layer to predict pixel patches. For the action decoder, similarly, we use an MLP to reduce the action-corresponding vectors to 7 dimensions. For the arm state and gripper state components, we employ different linear layers for decoding. For the gripper state, we apply binary thresholding at 0.5 to represent the gripper's open/close status (0 for closed, 1 for open).

Backbone Qwen3 [15] represents the latest advancement in the Qwen large language model family, comprising both dense and Mixture-of-Experts (MoE) architectures with parameter scales ranging from 0.6 to 235 billion. Utilizing Qwen3 as a backbone architecture offers significant advantages for large language model applications, primarily through its innovative parameter efficiency where Qwen3 dense base models achieve performance comparable to much larger Qwen2.5 models while using fewer parameters, with Qwen3-4B matching the performance of Qwen2.5-72B-Instruct. The MoE variants provide exceptional computational efficiency, as Qwen3-MoE base models deliver similar performance to Qwen2.5 dense base models while utilizing only 10% of the active parameters, resulting in significant savings in both training and inference costs.

A.3 Trajectory Processing

Our model supports variable-length image and action chunks as input, which we achieve through preprocessing of the training data. We assume that robot trajectory data of length t generally contains continuous text instructions ($G_{0:t}$), robot states ($S_{0:t}$), actions ($A_{0:t}$), and observations ($O_{0:t}$), where subscripts denote time indices. Given a configuration where the image chunk length is n (we maintain equal lengths for robot state chunks, text chunks, and image chunks) and the action chunk length is m, our processing pipeline works as follows.

Trajectory Splitting First, we extract the initial n text instructions ($G_{0:n}$), robot states ($S_{0:n}$), actions ($A_{0:n}$), and observations ($O_{0:n}$) from the trajectory as the initial chunk. This chunk serves as history for predicting the subsequent m consecutive actions ($A_{n:n+m}$). Subsequently, considering the information redundancy between consecutive image frames, we uniformly sample n times from the time steps $T_{n:n+m}$ to obtain the images for the next prediction target. This alternating prediction scheme enables the model to generate coherent robot trajectories by iteratively predicting action sequences and corresponding visual observations.

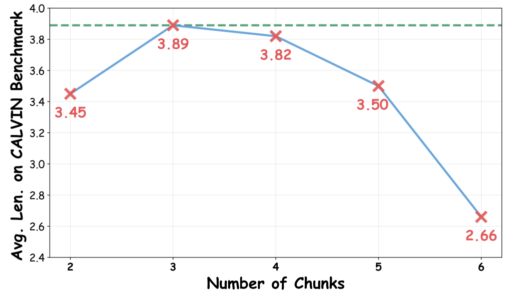


Figure 5: This figure shows the performance of the WorldAgen model on CALVIN as the number of chunks changes.

Example As shown in Figure 4, consider a configuration where the image chunk length is 3 and the action chunk length is 5. We first take the initial three frames ($T_{0:2}$) as the initial chunk, then predict the following 5 action frames ($A_{3:7}$). Next, we uniformly sample 3 times from $T_{3:7}$, obtaining T_3, T_5, and T_7 as the target images for prediction. And then we repeat this pipeline to get the whole trajectory. Note that during training, both action and image predictions employ teacher forcing.

B Baseline Results

In this section, we present more results of our baseline on CALVIN benchmark, including different number of chunks and different backbone.

B.1 Number of Chunks

The number of chunks in an input trajectory is given by:

$$N = \frac{L - n}{m} \tag{1}$$

where N is the number of chunks, L is the whole trajectory length, n and m represent the image and action chunk length.

To investigate the effect of the number of chunks, we fix the image chunk length to 1 and the action chunk length to 3, conducting comprehensive experiments with the number of chunks varying from 2 to 6. As illustrated in Figure 5, we observe the **same pattern** as in our experiments with image chunk lengths and action chunk lengths.

A larger number of chunks require the model to predict more action steps, which leads to accumulation of prediction errors. And the decreasing rate becomes larger when the number of chunks increases. However, an excessively small number of chunks limits the model's ability to learn the relationship between world modeling and actions during training.

According to our results, setting the number of chunks to 3 achieves the best performance.

We apply this finding to the configuration with image chunk length and action chunk length equal to 1 and 5, thereby achieving state-of-the-art results. This demonstrates

Both the scalability and robustness of our baseline approach, validating that the optimal chunk configuration principles discovered through systematic experimentation can be effectively transferred to different parameter settings to achieve superior performance.

C Test-Time Training Results

C.1 LoRA Configuration

We investigate the effect of LoRA rank on TTT performance while keeping other hyperparameters constant, including the amount of training data and learning rate. The results are shown in Table 5.

Table 5: Ablation study on LoRA rank during TTT. All other parameters are kept constant.

LoRA Rank	Avg. Len. ↑
16	3.928
32	3.918
64	3.918
128	**3.930**
256	3.923

As shown in Table 5, when the training data volume and learning rate are fixed, LoRA rank demonstrates minimal sensitivity to the results. The performance variations across different ranks are marginal (within 0.02), suggesting that ***under fixed learning rate and training data volume, TTT results are not sensitive to LoRA rank.*** Based on these findings, we adopt a LoRA rank of 128 for our main experiments as it achieves the best performance while maintaining computational efficiency.

C.2 TTT Data Sampling Volume

We further conducted ablation studies on the amount of test-time training data. We examine how the amount of data collected during the TTT phase affects model performance. The results are presented in Table 6.

Table 6: Ablation study on TTT data sampling volume. We vary the amount of test-time training data, defined as the product of number of samples and repeat times.

Test Time Training Data	Avg. Len. ↑
6	3.871
90	3.922
204	**3.928**
340	3.917

The results reveal an interesting trend regarding TTT data volume. As the number of training samples increases from 6 to 204, the average sequence length improves from 3.871 to 3.928, suggesting that more diverse exploration data provides richer information for world model adaptation. However, increasing the training data further to 340 leads to a slight drop (3.917). This suggests that:

Moderately increasing the training data volume during TTT improves performance. Excessive training samples may lead to overfitting on image generation rather than action prediction, highlighting the need for an optimal balance in TTT data collection.

C.3 LoRA vs. Full Fine-tuing

We conducted a comparative study between LoRA and Full Fine-Tuning (FFT) on the CALVIN dataset. In our experiments, we set the LoRA rank to 128. Both LoRA fine-tuning and FFT used identical hyperparameters: learning rate $lr = 0.0005$, Adam optimizer, and weight decay of 0.01.

Table 7: Comparison of test-time training with LoRA and FFT on CALVIN dataset. Avg. Len. represents the average sequence length.

Method	Avg. Len.
WorldAgent	3.87
WorldAgent + LoRA	**3.93**
WorldAgent + FFT	3.85

As shown in Table 7, our results reveal that FFT does not improve model performance and actually leads to a decline in performance. We attribute this phenomenon to the fact that

FFT causes the network to overfit to the test scenarios, whereas LoRA enables the model to acquire scenario-specific features while preserving the original scene perception capabilities. Therefore,

Table 8: TTT experiments under different number of chunks (N), image chunk length (n), and action chunk length (m) configurations.

N	n	m	LoRA Rank	Data Size	Before TTT After TTT
3	1	3	64	170	3.89 **3.90**
5	1	3	256	204	3.50 **3.67**
3	1	5	128	204	3.87 **3.93**

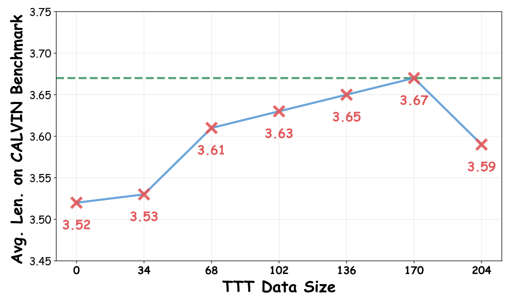


Figure 6: This figure shows that as the amount of data increases, the TTT effect gradually increases.

LoRA demonstrates superior performance by striking a better balance between adaptation to new scenarios and retention of pretrained knowledge.

C.4 Scalability and Robustness

To investigate the scalability and robustness of our method, we select three parameters crucial to TTT: number of chunks (N), image chunk length (n), and action chunk length (m). We conducted TTT experiments across different combinations of N, n, and m values.

We choose three representative settings with different configurations to demonstrate the generalizability of our approach across various parameter combinations.

To investigate the scalability and robustness of our method, we select three parameters crucial to TTT: number of chunks (N), image chunk length (n), and action chunk length (m). We conducted TTT experiments across different combinations of N, n, and m values.

We choose three representative settings with different configurations to demonstrate the generalizability of our approach across various parameter combinations.

As demonstrated in the Table 8, TTT consistently improves performance across different settings using only around 200 samples. This validates

The scalability and robustness of our TTT approach and it can effectively adapt to various architectural configurations while maintaining consistent improvement patterns.

C.5 Test-Time Training Data Size

Furthermore, during the grid search process for the experimental configuration with $n = 1$, $m = 3$, $N = 5$, as illustrated in the Figure 6, we discovered that TTT exhibits linear performance improvement with small amounts of data. However, when the data volume becomes excessive, we observe a phenomenon where TTT performance degrades. These experiments were conducted with LoRA rank = 256 and learning rate = 0.0005.

We attribute this phenomenon to the fact that excessive training samples cause LoRA to overfit on the world modeling component, making the model more biased towards predicting world observations rather than executing correct actions. This finding highlights

The importance of carefully balancing the amount of test-time training data to achieve optimal performance without compromising the model's action execution capabilities.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See Abstract and Section 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Section 2 and 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Open access to the data and code, with sufficient instructions could be provided.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: See Section 3.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Section 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper strictly conform with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Section 5.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: See References.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: New assets introduced in the paper could be documented and the documentation could be provided.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: It is not an important, original, or non-standard component of the core methods in this research.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.