IA-WORLD: RESIDUAL VLA POST-TRAINING WITH LATENT WORLD MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models provide strong priors for robotic manipulation, yet efficient online reinforcement learning adaptation remains challenging. Existing refinement methods typically lack explicit predictions of base-plan consequences, while sparse rewards provide limited feedback on task progress. We propose IA-World, a framework for residual VLA post-training that uses a latent world model to guide action correction and provide dense learning feedback. For each proposal from a frozen VLA, an Intent-Action Token (IA-Token) compresses proposal-associated action representations to condition a lightweight world model, which predicts the multi-step Base Future of executing the plan without correction. This predicted future directly conditions a residual policy to generate bounded corrections before execution. A goal-conditioned foresight scorer evaluates the predicted future at each decision point, and changes in this score across real transitions provide dense progress feedback beyond sparse environment rewards. Experiments on ManiSkill and LIBERO, using $\pi_{0.5}$ and Isaac GR00T respectively, show that IA-World improves task success over frozen base policies and evaluated RL-based post-training baselines. These results support base-plan predictions as a useful decision input for foresight-guided residual adaptation while preserving the pretrained VLA as a fixed behavioral prior.

1 INTRODUCTION

Vision-language-action (VLA) models have recently shown strong promise as generalist policies for robotic manipulation, mapping visual observations and language instructions to executable actions (Brohan et al., 2023; Kim et al., 2024; Team et al., 2024; Black et al., 2024; Intelligence et al., 2025b; Bjorck et al., 2025). Despite their strong pretrained priors, VLAs trained primarily through supervised imitation remain constrained by demonstration coverage and can fail when execution errors, distribution shifts, or long-horizon interactions drive them into unfamiliar states (Lu et al., 2025; Chen et al., 2025; Xu et al., 2026; Kim et al., 2026; Xiao et al., 2026). Reinforcement learning (RL) provides a natural mechanism for post-training these models through environment interaction and task feedback, enabling policies to improve beyond the behaviors contained in demonstrations (Lu et al., 2025; Intelligence et al., 2025a; Chen et al., 2025; Xu et al., 2026; Xiao et al., 2026). The key question is how to realize such adaptation efficiently while preserving the useful behavioral priors already encoded in large pretrained VLAs.

VLA RL post-training approaches differ in what they update. VLA-Updating Post-Training optimizes the VLA or its action-generating components using reward-driven objectives (Lu et al., 2025; Intelligence et al., 2025a; Chen et al., 2025). Updating these parameters requires balancing optimization cost, stability, and preservation of pretrained behaviors (Figure 1(a)). Frozen-VLA Action Refinement instead keeps the base VLA fixed and trains a compact auxiliary policy to refine its actions (Xu et al., 2026; Kim et al., 2026; Xiao et al., 2026; Yuan et al., 2025). This separation preserves a stable behavioral reference and limits online optimization to the auxiliary policy, supporting efficient task-specific post-training.

Representative frozen-VLA methods differ in how they implement refinement (Xu et al., 2026; Kim et al., 2026; Xiao et al., 2026; Yuan et al., 2025). RL Token (Xu et al., 2026) compresses VLA representations for a lightweight actor–critic; its actor conditions on a proposed action chunk and is regularized toward that proposal. Object-centric residual RL (Kim et al., 2026) instead learns

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

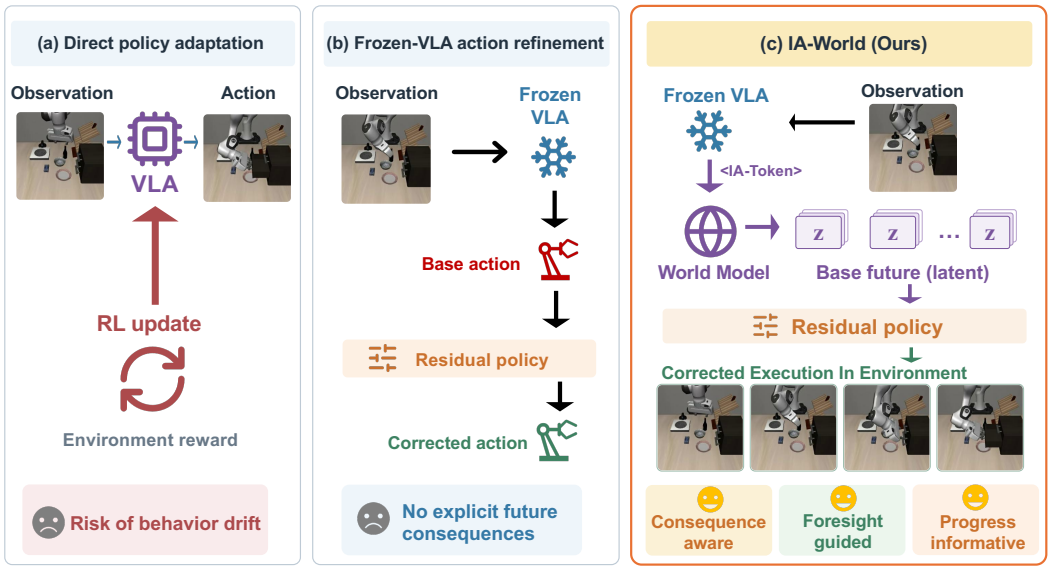


Figure 1: **IA-World: foresight-guided residual VLA post-training.** (a) Direct policy adaptation updates the VLA using environment rewards, with a risk of behavior drift. (b) Frozen-VLA action refinement corrects base actions without explicit predictions of their future consequences. (c) IA-World uses an IA-Token to condition a latent world model on the base plan; the predicted Base Future guides residual action correction and provides progress feedback for learning.

additive action corrections in simulation from object poses, robot state, and base actions, enabling zero-shot transfer to real robots. Both provide the refinement policy with current-state information and the base proposal, but neither explicitly supplies it with a predicted rollout of that proposal's consequences. We refer to this correction interface as reactive: the policy learns downstream effects implicitly from interaction, without an explicit base-plan prediction as a decision input. When task rewards are sparse, intermediate corrections also receive limited direct feedback about changes in task progress (Lu et al., 2025; Wang et al., 2024; Fang et al., 2026).

These limitations motivate two requirements for frozen-VLA refinement. First, correction should be consequence-aware: the policy should directly access predicted consequences of the current base plan before modifying it. Second, learning should be progress-informative: feedback should indicate whether the anticipated future becomes more favorable to the task goal after execution. Together, these requirements motivate foresight-guided residual correction while preserving the frozen VLA as a behavioral prior.

World2Act (Vuong et al., 2026) also freezes the VLA, training a residual module without environment rewards by contrastively aligning action latents with world-model dynamics conditioned on an initial observation and instruction. Whereas these predictions serve as latent-alignment targets, IA-World predicts the consequences of the current uncorrected base plan and supplies them directly to the residual policy at each decision point.

We introduce IA-World, a framework for residual VLA post-training with a latent world model (Figure 1). Given a base action plan from a frozen VLA, we distill its associated action representations into a compact Intent-Action Token (IA-Token). Conditioned on this token and the current latent state, a lightweight world model predicts the multi-step Base Future—the latent consequence of executing the plan without correction. Together with the current state and base action, this predicted future conditions a residual policy to generate a bounded correction before execution.

Base Future also supports dense progress feedback. A goal-conditioned foresight scorer evaluates the predicted future at each decision point; changes in this potential across real execution provide a dense shaping signal that complements sparse environment rewards. The next potential is computed from the new state's base proposal and predicted future. Residual action values are learned from real environment transitions, without model-predictive search over candidate corrections.

We evaluate IA-World on LIBERO (Liu et al., 2023) and ManiSkill (Tao et al., 2024) using base policies from both Isaac GR00T (Bjorck et al., 2025) and $\pi_{0.5}$ (Intelligence et al., 2025b). Across benchmarks and VLA backbones, IA-World consistently improves task success through residual correction and outperforms competing RL-based post-training approaches. Controlled ablations further demonstrate that explicitly conditioning the residual policy on the predicted Base Future improves action refinement beyond current-state and base-action information alone. These results support the use of base-plan predictions for residual action refinement in VLA post-training.

Our contributions are summarized as follows:

- We formulate an explicit base-plan foresight interface for residual VLA post-training, making the predicted consequences of the current uncorrected proposal directly available to the correction policy while retaining a frozen VLA as its behavioral reference.
- We propose IA-World, which distills proposal-associated VLA action representations into an IA-Token and uses a lightweight latent world model to predict the multi-step consequence of the uncorrected base plan. The predicted Base Future is explicitly provided to a bounded residual policy, enabling foresight-guided action refinement before execution.
- We construct dense progress feedback from changes in goal-conditioned predicted futures across real execution. Experiments on LIBERO and ManiSkill with Isaac GR00T and $\pi_{0.5}$ demonstrate consistent improvements over competing RL-based VLA post-training methods.

2 Related Work

2.1 Vision-language-action models.

Vision-language-action (VLA) models combine large-scale visual and language representations with robot demonstrations to learn generalist manipulation policies. Early efforts such as RT-2 and OpenVLA demonstrate that pretrained vision-language models can transfer semantic knowledge into robotic control, enabling language-conditioned action generation across diverse tasks (Brohan et al., 2023; Kim et al., 2024). Recent VLA systems, including the π family and Isaac GR00T, further improve generalization by scaling model capacity, training data, and embodiment diversity, establishing strong pretrained policies for robot manipulation (Black et al., 2024; Intelligence et al., 2025b; Bjorck et al., 2025).

Despite their strong pretrained priors, VLA models trained primarily through imitation learning remain constrained by demonstration coverage. When execution encounters unfamiliar states, accumulated errors, or long-horizon interactions beyond the demonstration distribution, the pretrained policy may fail to recover (Lu et al., 2025; Xu et al., 2026; Kim et al., 2026; Xiao et al., 2026; Ross et al., 2011). These limitations motivate reinforcement learning (RL) based post-training approaches, which aim to adapt pretrained VLA policies through environment interaction while preserving their generalizable behaviors.

2.2 RL fine-tuning of VLA models.

RL-based post-training methods for VLA models can be broadly categorized by whether the pretrained policy itself is updated or treated as a fixed behavioral prior. One line of work directly optimizes the pretrained VLA or its action-generation components using reward-driven objectives (Li et al., 2026). VLA-RL applies online RL to pretrained VLA models and introduces process-level reward modeling to alleviate sparse feedback, while RECAP incorporates demonstrations, autonomous experience, and human corrections through advantage-conditioned learning (Lu et al., 2025; Intelligence et al., 2025a). These approaches integrate improvements into the original policy representation, but adaptation stability, optimization cost, and preservation of pretrained capabilities must be jointly considered.

A complementary direction freezes the pretrained VLA and learns a lightweight auxiliary policy for action refinement. Earlier residual RL combined conventional feedback control with learned additive corrections (Johannink et al., 2019). Policy Decorator studies model-agnostic residual refinement for large pretrained policies, while object-centric residual RL learns additive corrections based on

object states, robot states, and base actions for zero-shot transfer (Yuan et al., 2025; Kim et al., 2026). More recently, PLD explores residual reinforcement learning as a mechanism for discovering failure cases and generating additional data for improving VLA policies (Xiao et al., 2026). These methods demonstrate that lightweight refinement can effectively exploit pretrained behaviors without fully updating the foundation model. DSRL instead adapts a frozen generative policy by learning to steer its latent-noise inputs (Wagenmaker et al., 2025).

Among VLA-specific refinement methods, RL Token introduces a compact interface between a frozen VLA and an actor–critic learner by compressing internal VLA representations into RL tokens. Its actor conditions on the proposed action chunk and is regularized toward the original policy behavior (Xu et al., 2026). Foresight Residual RL further incorporates predictive task-level information by using estimated downstream success probabilities to improve residual learning in long-horizon manipulation (Liu et al., 2026). However, existing refinement approaches generally provide the correction policy with current observations, compact representations, or reference actions, without explicitly predicting the future consequences of the specific sampled base action plan.

In contrast, our work studies a foresight-conditioned residual refinement framework, where the frozen VLA remains a behavioral prior while a lightweight residual policy receives an explicit prediction of the outcome of the current uncorrected base proposal.

2.3 World Models.

Latent world models support planning and policy optimization through imagined trajectories in PlaNet and Dreamer (Hafner et al., 2019b;a), and sample-efficient visual control on physical robots in DayDreamer (Wu et al., 2023). DINO-WM optimizes action sequences using predicted DINOv2 patch features (Zhou et al., 2024), while Imagination-Augmented Agents provide encoded model rollouts as policy context (Racanière et al., 2017).

World models also supply simulated rollouts for VLA policy optimization (Zhu et al., 2026; Li et al., 2025; Jiang et al., 2026). Seer uses predicted visual futures for inverse dynamics learning, while FLARE aligns policy representations with future observation embeddings for goal-conditioned action learning (Tian et al., 2025; Zheng et al., 2025). WCM combines latent future prediction with value estimation to improve VLA critic learning (Fei et al., 2026).

For reward and progress feedback, VIPER and Diffusion Reward derive rewards from learned visual dynamics (Escontrela et al., 2023; Huang et al., 2024), while RISE combines compositional world models with progress-based advantages to improve VLA policies through imagined outcomes (Yang et al., 2026). Complementary approaches VIP and LIV learn goal-conditioned visual representations that supply dense rewards (Ma et al., 2022; 2023).

World2Act post-trains frozen VLA policies by aligning action representations with instruction-conditioned latent dynamics (Vuong et al., 2026). IA-World instead predicts Base Future—the latent consequence of the current sampled VLA proposal without correction—and supplies it directly to the residual policy for action refinement. Changes in goal-conditioned future scores across real execution additionally provide progress feedback.

3 Method

3.1 Overview of IA-World

IA-World introduces an action-specific foresight interface for residual VLA post-training: it predicts the consequence of the current frozen VLA proposal before residual correction. At a decision point indexed by primitive control time t, the agent receives observation history $o_{\leq t}$, language instruction l, and proprioception p_t. The frozen VLA produces a base action chunk

$$A_t^{\text{base}} = \pi_{\text{VLA}}(o_{\leq t}, l, p_t). \tag{1}$$

The objective is to improve discounted environment return by learning a residual policy π_r while keeping π_{VLA} fixed. Its correction ΔA_t refines the execution prefix of the base chunk through an Action Composer, yielding the executed action chunk A_t^{final}.

Figure 2 summarizes the pipeline. A latent world model predicts the *Base Future* $\hat{Z}_t^{\text{base}}$, the latent consequence of executing the current proposal without residual correction. To condition this prediction on the proposal, the action-side hidden representation H_t^{act} is compressed into an Intent-Action Token (IA-Token) c_t. Together with state tokens Z_t, this token supplies the conditioning information for future prediction. The predicted Base Future enters the residual policy context h_t^r to guide correction before execution. TD3 then optimizes the residual policy from real environment interaction.

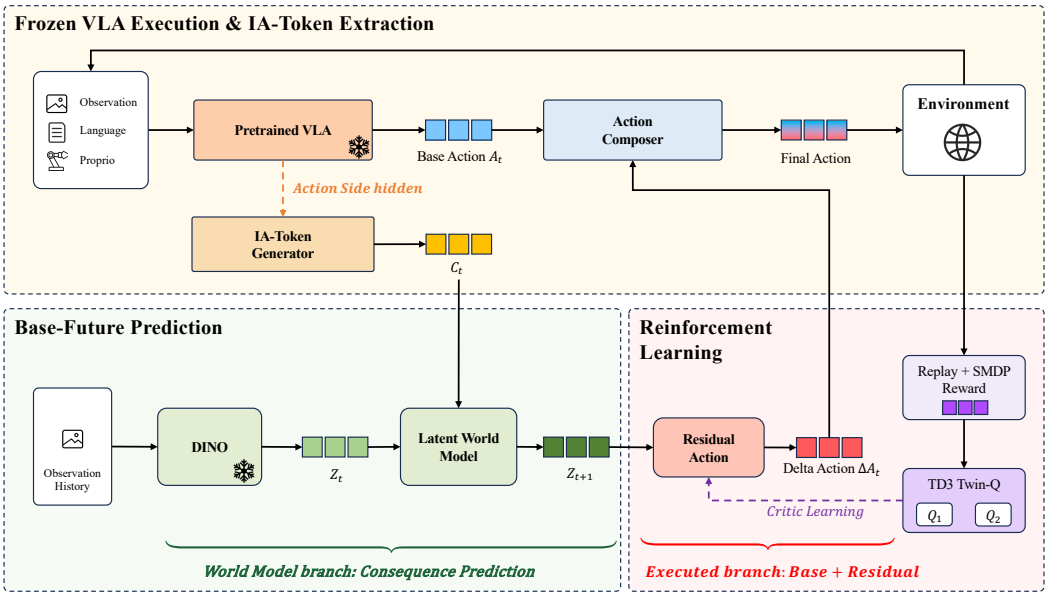


Figure 2: Overview of IA-World. The world-model output labeled Z_{t+1} schematically denotes the multi-step Base Future $\hat{Z}_t^{\text{base}}$, the predicted consequence of the current VLA proposal before residual correction. The IA-Token provides a compact proposal-specific conditioning interface; TD3 learns corrections from real execution.

3.2 Frozen VLA Execution and IA-Token Extraction

Base Future prediction requires conditioning information about the specific VLA decision being refined. Raw action coordinates specify commanded motion but do not expose the internal features used to generate that proposal. We therefore extract H_t^{act} from the action-generation branch in the same inference pass that produces A_t^{base}, preserving its association with the sampled proposal.

To supply this conditioning compactly, the IA-Token Generator G compresses the hidden sequence with a small transformer and attention pooling:

$$c_t = G(H_t^{\text{act}}). \tag{2}$$

Offline action and hidden-state reconstruction objectives encourage c_t to retain proposal-related information. The IA-Token thus serves as a compact interface that exposes proposal-specific information to the future predictor, with visual state information supplied separately by Z_t.

3.3 Latent World Model for Base Future Prediction

Base Future makes the expected consequence of the current VLA decision available before residual correction. We define *Base Future* as the latent consequence that would occur if the current frozen VLA action proposal were executed without residual correction. The prediction $\hat{Z}_t^{\text{base}}$ collects latent observations at multiple offsets within the proposal's execution prefix; t identifies the decision at which the prediction is made.

Unlike a forecast determined only by current observations, this target depends on the particular proposal sampled from those observations. While latent dynamics models may incorporate actions

as inputs, IA-World predicts the consequence of the specific proposal sampled by the frozen VLA before any residual correction and exposes this prediction to the correction policy.

A frozen visual encoder maps observations up to t to spatial state tokens Z_t. Conditioned on Z_t and c_t, the world model predicts

$$\hat{Z}_t^{\text{base}} = F_\theta(Z_t, c_t), \tag{3}$$

where F_θ is a transformer whose state-token processing is modulated by c_t through FiLM (Perez et al., 2018). This couples current state information with proposal-specific action-generation features without requiring the residual policy to infer the proposal's consequence solely from its action coordinates. The predicted future is not used for action search or planning; it serves as a foresight context for residual refinement.

The predictor is trained offline by minimizing mean-squared error against future tokens from uncorrected base executions, encoded by the same fixed state encoder. Each target is paired with the proposal that generated its action sequence, so supervision preserves the meaning of Base Future.

At each decision, the predicted Base Future augments the residual policy's current-state context: it indicates where the current VLA decision is expected to lead. This prediction precedes residual generation and does not describe the final corrected action's outcome; real interaction determines the value of the correction.

3.4 Foresight-conditioned Residual Reinforcement Learning

Residual RL exploits the predicted Base Future to decide whether and how to correct the current proposal using its anticipated consequence. The frozen VLA remains a stable behavioral prior, while the residual policy performs task-specific adaptation through bounded corrections. Let g denote visual goal prototypes obtained by encoding successful training endpoints with the frozen state encoder. We construct h_t^r from pooled state and predicted-future features, c_t, A_t^{base}, the mean of valid prototypes in g, proprioception p_t, and temporal context. A lightweight actor maps this context to a chunk-level correction:

$$\Delta A_t = \pi_r(h_t^r). \tag{4}$$

The Action Composer adds the correction to the base execution prefix and enforces action bounds:

$$A_t^{\text{final}} = \mathcal{C}(A_t^{\text{base}}, \Delta A_t). \tag{5}$$

We optimize the residual actor and twin critics with TD3 (Fujimoto et al., 2018) on real environment transitions. Chunk-level updates follow a semi-Markov decision process (SMDP) formulation, using a bootstrap discount of γ^k for k actually executed primitive steps. The critics evaluate the correction applied by the Action Composer in context h_t^r, and the actor learns corrections that increase this value. The VLA, state encoder, IA-Token Generator, and world model remain fixed during online learning.

Base Future also supplies dense feedback through an offline-trained, goal-conditioned transformer scorer S. It pools future tokens at each horizon and attends to the visual goal prototypes g, combining these features with the task instruction l. Offline supervision combines margin-based ranking of ordered training examples, binary supervision of successful versus unsuccessful endpoints, and score consistency between predicted and observed base-execution futures (Appendix A.5). Aggregated horizon scores define $\Phi_t = S(\hat{Z}_t^{\text{base}}, g, l)$, a learned surrogate for anticipated task progress, with higher scores indicating greater progress under this supervision. The scorer is frozen after offline training; its potential scale is calibrated on held-out data and fixed online. The corresponding chunk reward is

$$R_t = \sum_{j=0}^{k-1} \gamma^j r_{t+j}^{\text{env}} + \lambda_\Phi \left(\gamma^k \bar{\Phi}_{t+k} - \bar{\Phi}_t\right), \tag{6}$$

where r^{env} is the environment reward, γ is the primitive-step discount, λ_Φ controls shaping strength, and $\bar{\Phi}$ is the scaled and clipped potential. The shaping term follows the potential-based reward shaping form (Ng et al., 1999), with discounting over the executed chunk. The next potential is recomputed from the new proposal and Base Future at the state reached by real execution. This progress feedback supplements sparse rewards; the predicted latent future itself remains the actor's foresight input.

Table 1: Task success after online post-training. Entries report task success rate (%, higher is better). ManiSkill uses $\pi_{0.5}$, and LIBERO uses Isaac GR00T. Residual RL, RLT, and IA-World (Ours) are evaluated after 15,000 training steps in both settings. Evaluation uses 50 episodes per method on ManiSkill and 100 on LIBERO.

(a) ManiSkill
Peg Insertion

Method	Success (%)
Frozen Base	42
Residual RL	18
RLT	50
IA-World (Ours)	**62**

(b) LIBERO
Soup & Sauce in Basket

Method	Success (%)
Frozen Base	72
Residual RL	82
RLT	90
IA-World (Ours)	**94**

4 EXPERIMENTS

We evaluate how predicted Base Future supports residual refinement of frozen VLA policies. Task-success comparisons and future-input ablations assess the benefit in two environment–backbone configurations. Learning curves and qualitative rollouts show how this benefit develops during adaptation and appears in execution.

4.1 EXPERIMENTAL SETUP

Environments and base policies. We evaluate two distinct environment–backbone configurations: ManiSkill Peg Insertion with $\pi_{0.5}$ and LIBERO Soup & Sauce in Basket with Isaac GR00T. These configurations cover insertion and multi-object placement, respectively; the base VLA remains frozen during adaptation.

Comparison methods. **Frozen Base VLA** executes the pretrained policy without adaptation. **Residual RL** learns action corrections from current-state information and the base proposal, without predicted-future inputs. **RLT** is the comparison method based on RL Token, which uses compact VLA representations for action refinement. **IA-World (Ours)** uses predicted Base Future to condition residual action correction.

Training and evaluation protocol. Residual RL, RLT, and IA-World are evaluated after 15,000 training steps in each configuration. Success rate is the fraction of successful episodes, measured over 50 evaluation episodes per method on ManiSkill and 100 on LIBERO. ManiSkill control and optimization settings are provided in Appendix A.

4.2 MAIN RESULTS

Under the matched 15,000-step protocol, IA-World obtains the highest observed success rate in both evaluated configurations (Table 1). The consistent pattern is an improvement over the corresponding frozen base in each configuration.

IA-World improves over the frozen base by 20 percentage points on Peg Insertion and 22 points on Soup & Sauce in Basket, from base success rates of 42% and 72%, respectively. The comparable absolute gains show that successful residual adaptation is observed in both configurations, despite their different tasks and initial success rates. These results provide evidence that the benefit is not confined to a single evaluated setup, while preserving the pretrained VLA throughout adaptation.

Relative to RLT, IA-World obtains additional observed gains of 12 and 4 percentage points on the two tasks under the same online training-step budget. To examine the contribution of future conditioning more directly, Section 4.4 compares residual learning with and without predicted Base Future inputs.

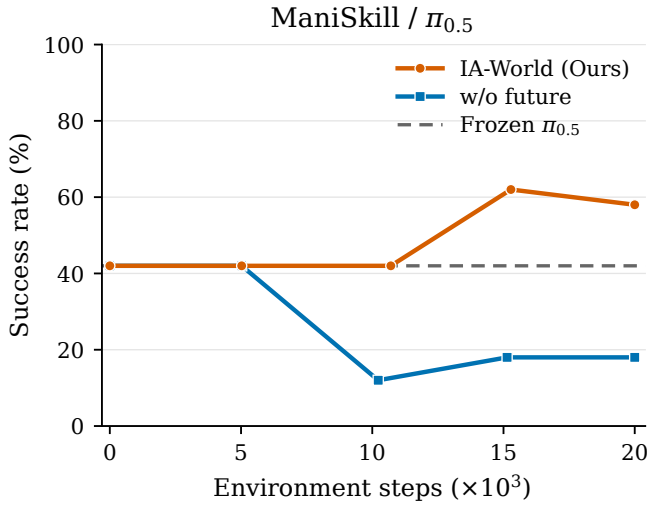


Figure 3: Online learning curves on ManiSkill with $\pi_{0.5}$. IA-World is compared with the variant without predicted-future inputs (*w/o future*) and the frozen base policy (dashed line). The horizontal axis counts environment steps in thousands; the vertical axis reports evaluation success rate (%).

Table 2: Ablation of predicted Base Future inputs with reward (including shaping), optimization, and policy architecture held fixed. The *w/o future* and *w/ future* columns report task success rate (%, higher is better). Gain is the difference between the two variants in percentage points (pp).

Task	w/o future	w/ future	Gain (pp)
Peg Insertion	18	**62**	+44
Soup & Sauce in Basket	82	**94**	+12

4.3 Dynamics of Online Adaptation

Figure 3 shows a separation in how the two residual learners use additional interaction. IA-World matches the frozen base in early evaluations, reaches 62% success at 15,000 environment steps, and retains an advantage over the base at 20,000 steps. The variant without future inputs falls below the base as training proceeds and remains below it at the later evaluations. Thus, the difference concerns the direction of adaptation as well as the final success rate: future-conditioned learning produces a net improvement over the pretrained behavior. This pattern supports the practical value of supplying predicted consequences during residual learning, with benefits visible at successive later checkpoints.

4.4 Ablation Study

We isolate future conditioning by holding the reward, including shaping, the optimization procedure, and the policy architecture fixed, varying only whether the residual learner receives the predicted Base Future. Under this controlled comparison, providing Base Future to the residual policy improves success by 44 percentage points on Peg Insertion and 12 points on Soup & Sauce in Basket (Table 2). On Peg Insertion, *w/ future* exceeds the frozen base while *w/o future* falls below it. On Soup & Sauce in Basket, future conditioning further improves a residual policy that already outperforms the base. The common finding is that future inputs are useful both for learning beneficial corrections and for improving corrections that are already effective.

These gains support the value of supplying predicted consequences as a control input under a shared reward signal. Base Future summarizes the expected outcome of the uncorrected proposal, giving the actor a consequence-oriented context for choosing a correction.

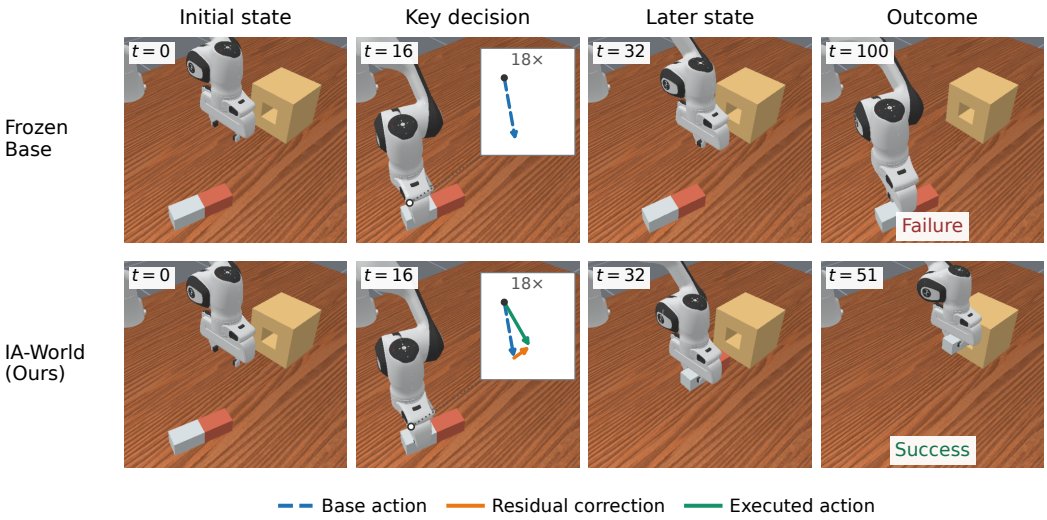


Figure 4: Qualitative comparison on ManiSkill Peg Insertion. The frozen base policy (top) and IA-World (bottom) are shown at the initial state, a key decision, a later state, and the outcome; frame labels indicate rollout time steps. Insets at $t = 16$ show the base action (blue dashed), residual correction (orange), and executed action (green), with arrows enlarged $18\times$ for visibility. IA-World succeeds at $t = 51$, while the frozen base rollout is unsuccessful at $t = 100$.

4.5 Qualitative Analysis

Figure 4 compares rollouts from the same initial scene on ManiSkill Peg Insertion. At $t = 16$, the action decomposition shows that the residual changes the direction of the proposed motion while keeping the executed command anchored to the base action. The correction therefore changes how the existing proposal is executed, illustrating the local intervention provided by the Action Composer.

By $t = 32$, IA-World has brought the peg toward the insertion site and subsequently succeeds at $t = 51$, while the frozen base remains unsuccessful at $t = 100$. The sequence links a directional adjustment to a rollout that reaches the goal, illustrating how residual corrections can turn the base policy's proposed behavior into successful execution. This gives a behavioral illustration of the aggregate gains, complementing the future-input comparison.

5 Conclusion

We presented IA-World, a framework for residual VLA post-training that explicitly exposes the predicted consequences of the current uncorrected VLA proposal to the correction policy. An IA-Token compresses proposal-associated action representations to condition a latent world model, which predicts the multi-step Base Future of the proposal. The predicted Base Future guides bounded residual corrections before execution, while changes in its goal-conditioned foresight score across real transitions provide dense progress feedback for learning. Empirical results demonstrate consistent improvements over the frozen base, while a controlled future-input ablation shows that predicted Base Future provides additional benefit beyond residual adaptation alone. These results support predicted base-plan consequences as a useful decision input for residual control, enabling foresight-guided adaptation while preserving the pretrained VLA as a fixed behavioral prior.

AI USE STATEMENT

We used generative AI tools to assist with language editing and polishing of the manuscript, including improving grammar, clarity, and readability. We take full responsibility for the final content of this paper, including all claims, results, and references, as well as any text prepared with AI assistance.

ETHICS STATEMENT

This work studies residual adaptation of vision-language-action policies in simulated LIBERO and ManiSkill environments. The method constrains residual corrections through explicit action bounds. Responsible deployment on physical robots requires task-specific safety validation, appropriate operational limits, and human supervision.

REPRODUCIBILITY STATEMENT

To support reproducibility, Section 3 describes the IA-Token interface, Base Future prediction, and residual reinforcement-learning formulation. Appendix A details the auxiliary training objectives, construction of prediction targets, action composition, and chunk-level learning procedure; Table 3 summarizes the ManiSkill control and optimization settings. Section 4.1 and Table 1 report the environment–backbone combinations, training-step budgets, success-rate definition, and evaluation episode counts for the main comparisons.

REFERENCES

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. Conrft: A reinforced fine-tuning method for vla models via consistency policy. *arXiv preprint arXiv:2502.05450*, 2025.

Alejandro Escontrela, Ademi Adeniji, Wilson Yan, Ajay Jain, Xue Bin Peng, Ken Goldberg, Youngwoon Lee, Danijar Hafner, and Pieter Abbeel. Video prediction models as rewards for reinforcement learning. *Advances in Neural Information Processing Systems*, 36:68760–68783, 2023.

Yu Fang, Wanxi Dong, Jiaqi Liu, Yue Yang, Mingxiao Huo, Yao Mu, Huaxiu Yao, Li Erran Li, Daniel Szafir, and Mingyu Ding. Densereward: Dense reward learning via failure synthesis for robotic manipulation. *arXiv preprint arXiv:2607.13033*, 2026.

Senyu Fei, Xiaopeng Yu, Siyin Wang, Xianzhong Zhao, Jingjing Gong, and Xipeng Qiu. Wcm: A world critic model for vision-language-action reinforcement learning. *arXiv preprint arXiv:2607.29613*, 2026.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. Pmlr, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019b.

Tao Huang, Guangqi Jiang, Yanjie Ze, and Huazhe Xu. Diffusion reward: Learning rewards via conditional video diffusion. In *European conference on computer vision*, pp. 478–495. Springer, 2024.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi^*_{0.6}$: a vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025a.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025b.

Zhennan Jiang, Shangqing Zhou, Yutong Jiang, Zefang Huang, Mingjie Wei, Yuhui Chen, Tianxing Zhou, Zhen Guo, Hao Lin, Quanlu Zhang, et al. Wovr: World models as reliable simulators for post-training vla policies with rl. *arXiv preprint arXiv:2602.13977*, 2026.

Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 international conference on robotics and automation (ICRA)*, pp. 6023–6029. IEEE, 2019.

Kinam Kim, Namiko Saito, Heecheol Kim, Katsushi Ikeuchi, Jaegul Choo, and Yasuyuki Matsushita. Object-centric residual rl for zero-shot sim-to-real vla enhancement. *arXiv preprint arXiv:2606.18953*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Yang Zhaohui, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, et al. Simplevla-rl: Scaling vla training via reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 125846–125869, 2026.

Hengtao Li, Pengxiang Ding, Runze Suo, Yihao Wang, Zirui Ge, Dongyuan Zang, Kexian Yu, Mingyang Sun, Hongyin Zhang, Donglin Wang, et al. Vla-rft: Vision-language-action reinforcement fine-tuning with verified rewards in world simulators. *arXiv preprint arXiv:2510.00406*, 2025.

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.

Yuhan Liu, Xinyu Zhang, Litao Liu, and Abdeslam Boularias. Foresight residual rl for long-horizon robot manipulation with vision-language-action models. *arXiv preprint arXiv:2607.16506*, 2026.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. In *International Conference on Machine Learning*, pp. 23301–23320. PMLR, 2023.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287. Citeseer, 1999.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Sébastien Racanière, Théophane Weber, David Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adrià Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, et al. Imagination-augmented agents for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, et al. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

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. In *International Conference on Learning Representations*, volume 2025, pp. 92033–92052, 2025.

An Dinh Vuong, Tuan Van Vo, Abdullah Sohail, Haoran Ding, Liang Ma, Xiaodan Liang, Anqing Duan, Ivan Laptev, and Ian Reid. World2act: Latent action post-training from world model dynamics. *arXiv preprint arXiv:2603.10422*, 2026.

Andrew Wagenmaker, Mitsuhiko Nakamoto, Yunchu Zhang, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. *arXiv preprint arXiv:2506.15799*, 2025.

Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. *arXiv preprint arXiv:2402.03681*, 2024.

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

Wenli Xiao, Haotian Lin, Andy Peng, Haoru Xue, Tairan He, Zhengyi Luo, Yuqi Xie, Fengyuan Hu, Jim Fan, Guanya Shi, et al. Self-improving vision-language-action models with data generation via residual rl. In *International Conference on Learning Representations*, volume 2026, pp. 13209–13236, 2026.

Charles Xu, Jost Tobias Springenberg, Michael Equi, Ali Amin, Adnan Esmail, Sergey Levine, and Liyiming Ke. Rl token: Bootstrapping online rl with vision-language-action models. *arXiv preprint arXiv:2604.23073*, 2026.

Jiazhi Yang, Kunyang Lin, Jinwei Li, Wencong Zhang, Tianwei Lin, Longyan Wu, Zhizhong Su, Hao Zhao, Ya-Qin Zhang, Li Chen, et al. Rise: Self-improving robot policy with compositional world model. *arXiv preprint arXiv:2602.11075*, 2026.

Xiu Yuan, Tongzhou Mu, Stone Tao, Yunhao Fang, Zhang Zhang, and Hao Su. Policy decorator: Model-agnostic online refinement for large policy model. In *International Conference on Learning Representations*, volume 2025, pp. 28129–28164, 2025.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

Fangqi Zhu, Zhengyang Yan, Zicong Hong, Quanxin Shou, Xiao Ma, and Song Guo. Wmpo: World model-based policy optimization for vision-language-action models. In *International Conference on Learning Representations*, volume 2026, pp. 62486–62502, 2026.

A METHOD AND MANISKILL IMPLEMENTATION DETAILS

This appendix expands the auxiliary objectives, target construction, and residual-learning procedure in Section 3. ManiSkill control and optimization settings are summarized in Table 3.

A.1 ACTION-SPECIFIC FEATURES AND RECONSTRUCTION OBJECTIVES

Action features H_t^{act} and the base plan A_t^{base} are extracted from the same frozen-VLA inference call. The IA-Token Generator processes valid action-side hidden positions with layer normalization, a linear projection, a transformer, and learned-query attention pooling; padding is masked throughout. For GR00T, these features are the action-decoder inputs at the final denoising iteration, retaining the first 16 valid slots paired with the decoded 16×7 chunk. The token summarizes the full plan, while the residual acts on its execution prefix.

Two auxiliary heads reconstruct the normalized base chunk and a compressed hidden-state target:

$$\mathcal{L}_{\text{rec}} = \lambda_a \, \text{MSE}_w\big(D_A(c_t), A_t^{\text{base}}\big) + \lambda_h \, \text{MSE}\big(D_H(c_t), \text{sg}[P(H_t^{\text{act}})]\big) \,, \tag{7}$$

where sg denotes stop-gradient and λ_a, λ_h weight the losses. The action loss assigns greater weight to the executable prefix and is normalized by the sum of valid time-and-coordinate weights. The fixed transform P averages valid hidden positions and applies PCA fitted on training data. Each head uses layer normalization, a hidden linear layer with GELU, and an output projection. These heads are used only during offline training.

A.2 VISUAL TOKENS AND BASE-FUTURE TARGETS

A frozen DINOv2-Base encoder (Oquab et al., 2023) maps center-cropped 224×224 RGB images to a 16×16 patch grid with 768 channels. Inputs are scaled to $[0, 1]$ and normalized with mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$. Spatial mean pooling to 4×4 precedes a fixed, data-independent orthogonal projection to 512 dimensions. ManiSkill concatenates main- and wrist-camera grids into 32 state tokens; LIBERO uses 16 main-camera tokens. Current observations, future targets, and goal prototypes share this frozen transform.

Figure 5 illustrates $F_\theta(Z_t, c_t)$. The IA-Token supplies the proposal condition: its encoded features generate FiLM scales and shifts for the layer-normalized state tokens before self-attention and feed-forward processing, with residual connections in both sublayers. The multi-horizon output preserves spatial-token order and associates each prediction with an offset within the base plan's execution prefix.

Offsets are measured in primitive control steps. For horizon set $\mathcal{T}$, let $Y_{t,\tau}$ be the encoded observation at offset τ under the same uncorrected proposal, and let $m_{t,\tau}$ mark valid targets. Training minimizes

$$\mathcal{L}_{\text{future}} = \frac{\sum_t \sum_{\tau \in \mathcal{T}} m_{t,\tau} \, \text{MSE}\Big(\hat{Z}_{t,\tau}^{\text{base}}, \text{sg}[Y_{t,\tau}]\Big)}{\sum_t \sum_{\tau \in \mathcal{T}} m_{t,\tau}}, \tag{8}$$

where MSE averages over spatial tokens and feature coordinates. Targets beyond the actual execution duration are masked, and each training batch contains valid targets. Corrected online rollouts supply TD3 transitions, while the predictor remains fixed after offline training on base executions.

A.3 RESIDUAL ACTOR AND ACTION COMPOSITION

The residual actor has hidden widths 512 and 256 with layer normalization and GELU activations. Its context h_t^r contains spatially pooled state and future features, the IA-Token, the flattened full base plan, a mean over valid goal prototypes, proprioception, and normalized elapsed and remaining episode time. In ManiSkill, the actor outputs 8×8 residual logits for the execution prefix of a 10×8 base chunk. Zero initialization of the output weights and bias makes its initial deterministic correction zero.

The Action Composer transforms logits with an elementwise $\tanh$, applies per-coordinate residual scales, and masks disabled coordinates. For ManiSkill joint-delta control, the seven arm coordinates have residual bound 0.05, while the gripper correction is fixed to zero. Exploration is added in these

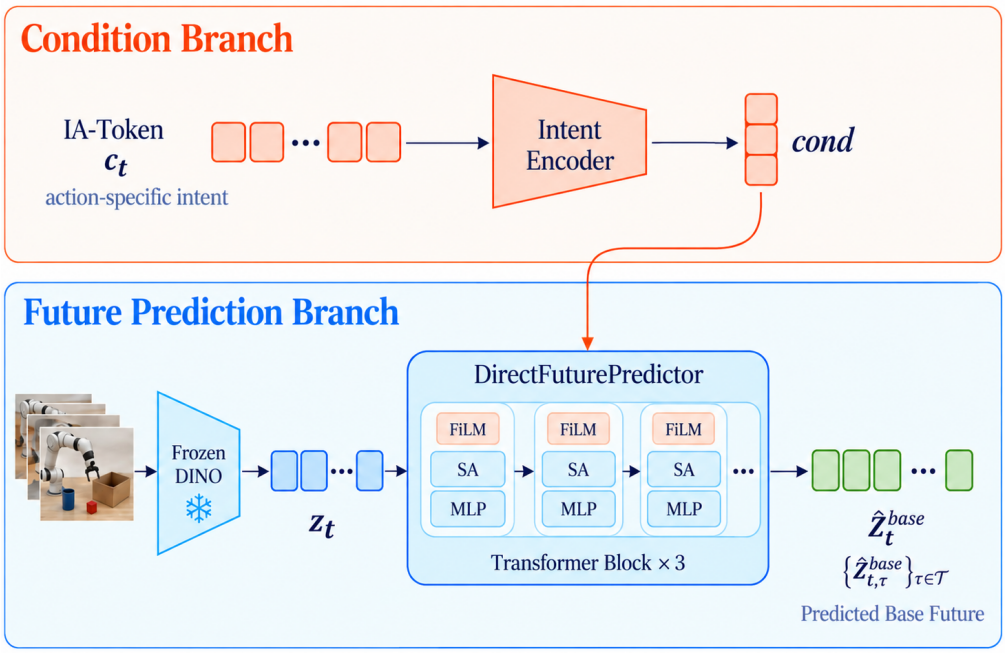


Figure 5: Latent world model conditioned on the IA-Token. The condition branch encodes c_t to modulate state-token processing through FiLM; the prediction branch maps frozen visual tokens to the uncorrected plan's latent future. The illustrated Base Future $\hat{Z}_t^{\text{base}}$ collects predictions $\{\hat{Z}_{t,\tau}^{\text{base}}\}_{\tau\in\mathcal{T}}$ at multiple offsets as defined in Section 3.3. SA denotes self-attention.

same action coordinates. The resulting command is projected onto the intersection of the residual bound and the remaining feasible range around the base action. In particular, for prefix position j and coordinate d, the applied correction lies in

$$\left[\max\left(-\alpha_d, a_d^{\min} - A_t^{\text{base}}[j,d]\right),\ \min\left(\alpha_d, a_d^{\max} - A_t^{\text{base}}[j,d]\right)\right], \tag{9}$$

where α_d is the residual bound and $[a_d^{\min}, a_d^{\max}]$ is the controller's action range. Disabled coordinates remain zero. This composition is shared by environment interaction, actor optimization, and target-policy smoothing.

Replay stores the effective correction ΔA_t^{eff}, defined as the composed action minus its base execution prefix. Each critic receives h_t^r concatenated with this correction. Base Future supplies the policy's foresight input, while scalar foresight scores supply the reward-shaping term in Eq. 6.

A.4 Chunk-Level TD3 and Episode Boundaries

Each replay item records a real executed chunk of duration $k \le 8$, with primitive-discounted reward R_t from Eq. 6 and bootstrap discount γ^k. The next context uses the reached observation, a new base proposal, and its Base Future; this context is also reused for the next behavior decision.

Let b_t be the bootstrap mask and $\bar{Q}_1, \bar{Q}_2$ the target critics. The TD3 target is

$$y_t = R_t + \gamma^k b_t \min_{i\in\{1,2\}} \bar{Q}_i\left(h_{t+k}^r, \Delta A_{t+k}^{\text{eff,target}}\right), \tag{10}$$

where $\Delta A_{t+k}^{\text{eff,target}}$ combines the target actor's output with clipped Gaussian smoothing noise through the Action Composer. Both critics minimize squared TD error; the actor maximizes the first critic through the composed action. Target networks follow Polyak updates after delayed actor updates. The VLA, visual encoder, IA-Token Generator, predictor, and scorer remain fixed.

For ManiSkill, $b_t = 0$ at true termination or the 100-step finite-horizon boundary and $b_t = 1$ otherwise. Execution stops at the boundary, k counts only applied actions, and replay retains the

final observation separately from the next reset. Replay warmup comprises 512 zero-residual and 512 bounded-exploration transitions. Critic updates require 128 replay items. Actor updates require 1,024 items, 512 nonzero-exploration transitions, a nonzero-exploration fraction of at least one half, and eight successful training episodes; thereafter, the actor updates every second critic update.

Table 3: ManiSkill control and residual-learning settings. Episode limits count primitive control steps; replay sizes count chunk transitions. Noise is applied only to enabled arm coordinates.

Setting	Value
Task	PegInsertionSideHardClearance-v1
Control mode	Joint-position delta
Base chunk / maximum executed prefix	10×8 / 8×8
Episode horizon / parallel environments	100 / 8
Actor and critic hidden widths	512, 256
Arm residual bound / gripper residual	0.05 / 0
Primitive-step discount γ	0.99
Replay capacity / minibatch size	10,000 / 128
Actor / critic learning rate	10^{-4} / 3×10^{-4}
Critic updates per collected transition	1
Actor delay / target-network update rate	2 / 0.005
Exploration / target-noise standard deviation	0.01 / 0.01 (arm coordinates)
Target-noise clipping bound / gradient-norm bound	0.02 / 10

A.5 Auxiliary Scorer Objective and Potential Calibration

The goal-conditioned scorer supplies the shaping potential in Section 3.4. It pools future tokens at each horizon, cross-attends to goal prototypes from training-success endpoints, and combines the resulting features with language and horizon embeddings. Padding is masked. A transformer and scalar head produce scores $s_{t,\tau}$, aggregated using learned logits a_τ and horizon mask $v_{t,\tau}$:

$$\Phi_t = \sum_{\tau \in \mathcal{T}} \rho_{t,\tau} s_{t,\tau}, \qquad \rho_{t,\tau} = \frac{v_{t,\tau} \exp(a_\tau)}{\sum_{\tau' \in \mathcal{T}} v_{t,\tau'} \exp(a_{\tau'})}. \tag{11}$$

Offline scoring uses observed-target validity; online scoring uses the prediction horizons available at the current decision. Each scored example has at least one valid horizon, and higher scores indicate greater anticipated progress.

The scorer objective combines margin-based ranking of ordered examples, binary supervision of successful versus unsuccessful endpoints, and alignment of scores on predicted and observed Base Futures:

$$\mathcal{L}_S = \lambda_{\text{rank}} \mathbb{E}[\max(0, m - s^+ + s^-)] + \lambda_{\text{end}} \mathcal{L}_{\text{BCE}} + \lambda_{\text{cons}} \mathbb{E}[(s^{\text{pred}} - \text{sg}[s^{\text{real}}])^2]. \tag{12}$$

Here s^+, s^- score a preferred and less-preferred example, $m > 0$ is the ranking margin, and $\mathcal{L}_{\text{BCE}}$ supervises endpoint logits. The consistency term pairs predicted and observed base-execution futures, with the observed-future score detached.

Potential normalization is fitted offline on held-out calibration values using an uncentered root-mean-square scale σ_Φ. The online learner uses

$$\bar{\Phi}_t = \text{clip}\left(\frac{\Phi_t}{\sigma_\Phi + \epsilon}, -B, B \right), \tag{13}$$

where $\epsilon > 0$ and $B > 0$ are fixed. The scale $\sigma_\Phi = \sqrt{n^{-1} \sum_i \Phi_i^2}$ is fitted on held-out calibration scores and frozen online. Continuing transitions recompute the next potential from the new proposal at the reached state. Terminal transitions, including the ManiSkill finite-horizon boundary, use zero next potential.

For decision times $t_0 = 0$ and $t_{i+1} = t_i + k_i$, this convention gives

$$\sum_{i=0}^{J-1} \gamma^{t_i} \left(\gamma^{k_i} \bar{\Phi}_{t_{i+1}} - \bar{\Phi}_{t_i} \right) = -\bar{\Phi}_0 + \gamma^{t_J} \bar{\Phi}_{t_J} = -\bar{\Phi}_0, \tag{14}$$

using the same primitive-step discount and actual chunk durations as the TD3 target.

B Additional Experimental Details

B.1 Tasks, Base Policies, and Data

ManiSkill pairs $\pi_{0.5}$ with Peg Insertion under joint-position delta control (Table 3). LIBERO pairs GR00T-N1.7-LIBERO with Soup & Sauce in Basket from LIBERO-10, requiring placement of both objects in the basket. Its seven-dimensional action comprises six end-effector motion coordinates and one gripper command.

Offline data. Uncorrected base rollouts provide paired hidden features, plans, and observation targets for reconstruction and prediction. Successful training endpoints provide goal prototypes; endpoint labels and ordered examples supervise the scorer. Representation fitting and auxiliary training use the training split; model selection and potential calibration use held-out data. Evaluation episodes are separate, and all frames from a trajectory remain within one split. Feature caches retain episode, decision, and horizon-validity indices.

LIBERO action interface. The adapter retains a 16×7 plan and executes up to eight actions. Residuals have shape 8×7, with bound 0.05 on the six motion coordinates and zero gripper correction or exploration. The Action Composer follows Eq. 9, then converts normalized actions to environment commands once. Execution stops immediately at termination.

B.2 Training and Evaluation Settings

Both environments use the chunk-level learner in Appendix A.4; Table 4 lists the LIBERO settings.

Table 4: LIBERO control and residual-learning settings. Action bounds and noise scales are in normalized action coordinates. The gripper follows the frozen base policy.

Setting	Value
Base chunk / maximum executed prefix	16×7 / 8×7
Control frequency / episode limit	20 Hz / 720 steps
State tokens / token width	16 / 512
Proprioceptive / temporal feature dimensions	8 / 2
Actor and critic hidden widths	512, 256
Motion residual bound / gripper residual	0.05 / 0
Primitive-step discount γ	0.99
Replay capacity / minibatch size	20,000 / 128
Actor / critic learning rate	3×10^{-4} / 3×10^{-4}
Critic updates per collected transition	1
Actor delay / target-network update rate	2 / 0.005
Exploration / target-noise standard deviation	0.01 / 0.01
Target-noise clipping bound / gradient-norm bound	0.02 / 10

Interaction accounting. Environment steps sum actually executed primitive actions across parallel environments, including warmup. Replay sizes and update ratios count chunk transitions. Evaluation interaction is excluded from the training budget.

Future-input ablation. The *w/o future* variant replaces the future slice of h_t^r with zeros at the same width during collection, replay construction, optimization, and evaluation, including both online and target critics. Architecture, other context fields, action composition, and reward shaping remain unchanged. The scorer retains access to Base Future in both variants.

Evaluation behavior. Evaluation fixes all weights and disables residual exploration while preserving the base VLA's inference procedure. The environment's success predicate is counted once per episode.

B.3 Evaluation Budget and Checkpoints

Main comparisons and the future-input ablation use 15,000 training steps and 50 evaluation episodes on ManiSkill or 100 on LIBERO. Success rate is the percentage of successful episodes, and gains are reported in percentage points. Figure 3 additionally evaluates successive checkpoints along the training trajectory.

C Prediction Metrics and Interface Consistency

Horizon-wise prediction error. For targets and masks from Appendix A.2, the error at offset τ is

$$E_\tau = \frac{\sum_t m_{t,\tau} \| \hat{Z}^{\text{base}}_{t,\tau} - Y_{t,\tau} \|_F^2}{ND \sum_t m_{t,\tau}}, \tag{15}$$

where N is the number of spatial tokens and D their width. This normalization retains a common per-coordinate scale across horizons. The mean of E_τ, weighted by valid-target counts, recovers the aggregate loss in Eq. 8.

Interface alignment. Hidden features and plans share one VLA inference call, targets belong to that plan's uncorrected execution, and cached and online features share a frozen visual transform. These alignments preserve the meaning of Base Future from offline supervision to online correction. The policy-input ablation applies the same future mask throughout the actor–critic learning and evaluation paths, as specified in Appendix B.

D Computational Cost

We account for forward computation, online updates, and replay storage. Each decision requires one VLA inference, visual encoding, IA-Token extraction, future prediction, and residual action composition. The scorer reuses Base Future for shaping; the next context is reused for the following behavior decision. Prediction occurs once per chunk, with no search over candidate corrections.

Table 5 summarizes module roles. Online gradient updates are confined to the residual actor and twin critics, so frozen modules require neither optimizer states nor backward activations. During fixed-policy execution, only the action-generation path is needed.

Table 5: Module roles in computational cost. "Fixed" modules are not updated by online RL. Target actor and critic parameters follow Polyak updates.

Component	Online learning	Fixed-policy execution
Base VLA and visual encoder	Fixed	Forward inference
IA-Token Generator and world model	Fixed	Forward inference
Reconstruction heads	Unused	Unused
Foresight scorer	Fixed; reward computation	Unused
Residual actor	Gradient updates	Forward inference
Twin critics	Gradient updates	Unused
Target actor and critics	Polyak updates	Unused
Action Composer	No learned parameters	Action composition

Interaction frequency and storage. For J executed chunks of duration $k_i \leq 8$, the primitive-step budget is $T = \sum_{i=1}^{J} k_i$. Forward computation occurs at these J decision points; TD3 updates follow the collected-transition and delayed-actor schedules. Replay stores fixed contexts and effective corrections, allowing minibatch updates without rerunning the VLA or visual encoder. With context width d_h, correction width d_a, and capacity M, these arrays require $O(M(2d_h + d_a))$ scalars, plus rewards, discounts, and episode metadata. Frozen goal features are reused across decisions.