WAVE: WORLD-MODEL-AUGMENTED VALUE LEARNING FOR ACTION-CHUNK REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Action chunking provides a temporal abstraction for long-horizon reinforcement learning, but a chunk critic must infer what an entire H-step action sequence will accomplish from reward and temporal-difference supervision alone. A world model can predict these consequences, but substituting its predicted endpoints into the Bellman backup lets model error change the preferred chunk: we construct an example in which a small endpoint error yields a suboptimal policy despite exact rewards and an exactly solved Bellman equation. With observed endpoints, by contrast, the Bellman fixed point is optimal for any world model, so model error can affect only what the critic represents. We therefore introduce **WAVE** (World-Model-Augmented Value Learning), which conditions the critic on a causal world-model posterior through a zero-initialized residual adapter and trains its features to predict the model's current latent state and the H-step consequence of the same chunk, while every Bellman target uses real rewards and observed endpoints. On two OGBench manipulation tasks with three seeds each, WAVE improves on Q-Chunking, raising the area under the success curve over 1M online interactions from 0.832 to 0.929 on Cube and from 0.922 to 0.956 on Scene and reaching 70% and 80% success with 76–85% fewer real interactions on average; substituting predicted endpoints from the same world model instead yields near-zero success. Ablations attribute most of the auxiliary objective's gain to predicting future rather than current latent states.

1 INTRODUCTION

Action chunking lets robot policies produce temporally coordinated sequences of low-level actions (Zhao et al., 2023). Q-Chunking (QC) brings this abstraction to reinforcement learning by learning values over entire action sequences, propagating returns across multiple environment steps per backup (Li et al., 2025); recent variants improve reactivity and combine chunk-level with single-step value learning (Li et al., 2026; Song et al., 2026). Because real interaction is costly, these learners are most attractive when pretrained on offline trajectories and then adapted online with a limited budget; this paper aims to make that adaptation more sample-efficient.

A chunk critic must learn what an H-step action sequence will accomplish from only a scalar H-step return and a bootstrapped value. The offline trajectories also record how actions change the environment, which a world model can learn (Hafner et al., 2019; 2020). Model-based methods typically turn such predictions into training data or value targets, through imagined trajectories (Hafner et al., 2020; Janner et al., 2019) or model-based value expansion (Feinberg et al., 2018; Buckman et al., 2018), and therefore need the model to be accurate wherever it enters a target. For a chunk critic, the most direct analogue, which we call Direct-WM, substitutes the predicted chunk endpoint into the H-step Bellman backup (Figure 1(a)). An incorrect endpoint then changes the bootstrap value and, through value learning, the preferred chunk; in our experiments, Direct-WM fails on both tasks although it uses the same world model as our method (Figure 1(c)).

Where a prediction enters value learning determines what its errors can do. In a two-state construction with a fixed model, endpoint substitution yields a suboptimal chunk policy even with exact rewards and an exactly solved Bellman equation (Lemma 1). If the backup instead keeps the ob-

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.

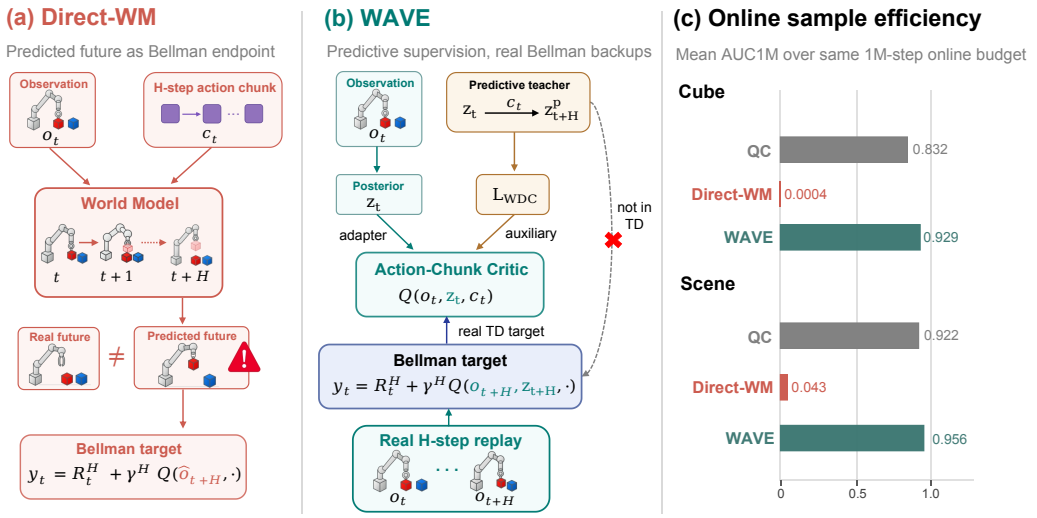


Figure 1: **Using a world model in action-chunk value learning.** **(a)** Direct-WM substitutes the model-predicted H-step endpoint into the Bellman backup, so value learning depends directly on prediction accuracy. **(b)** WAVE conditions the critic on the causal posterior and uses the action-conditioned H-step prediction only as an auxiliary WDC target; Bellman targets use real H-step replay. **(c)** Mean AUC over 1M real interactions: WAVE exceeds QC on both tasks, whereas Direct-WM stays near zero.

served endpoint, its fixed point is the optimal value function for any world model, and the model can influence the critic only through its inputs and training signals (Proposition 2). A world model should therefore shape what the critic represents, not what it is trained to fit.

Based on this insight, we propose **WAVE** (World-Model-Augmented Value Learning), which uses an offline-pretrained recurrent world model to shape the chunk critic's representation but never to replace real transitions in its targets (Figure 1(b)). The world model's causal posterior, which summarizes real interaction history, conditions the critic through a zero-initialized residual adapter, so the augmented critic equals the pretrained critic when online adaptation begins. Its H-step prior predicts the latent consequence of the complete chunk being valued, and an auxiliary objective, world-model dynamics consistency (WDC), trains the critic's representation to predict this consequence together with the current posterior. Unlike self-predictive objectives for critics over individual actions (Schwarzer et al., 2021; Hansen et al., 2022; Fujimoto et al., 2025), WAVE takes its targets from a separately trained world model and aligns them with the chunk horizon. Bellman targets use real rewards, observed endpoints, and posteriors inferred from real history: *predict with the model, bootstrap from reality*.

We instantiate WAVE on QC and evaluate it offline-to-online on two OGBench manipulation tasks (Park et al., 2025a) with three seeds per method. Over 1M real interactions, WAVE raises the area under the success curve from 0.832 to 0.929 on Cube and from 0.922 to 0.956 on Scene, and on average reaches 70% and 80% success with 76–85% fewer interactions than QC. Ablations on Cube attribute most of the WDC gain to predicting future rather than current latent states, and WAVE improves over QC at both offline-data scales we test.

Our contributions are:

- WAVE, which conditions an action-chunk critic on a causal world-model posterior without changing the pretrained critic at initialization and supervises its representation with the chunk's predicted H-step consequence, while keeping Bellman targets on real replay.
- An analysis of two uses of a chunk-level prediction: as a Bellman endpoint, a small error can reverse the chunk ranking, whereas with real-replay targets model error leaves the optimal fixed point unchanged.

- Experiments on two OGBench tasks showing that WAVE improves online sample efficiency over QC while endpoint substitution with the same world model fails, with component, predictive-target, and data-scale ablations.

2 Related Work

Temporal abstraction and action-chunk learning. Temporal abstraction has long been studied in reinforcement learning, including the options framework (Sutton et al., 1999). In robot learning, ACT and Diffusion Policy predict temporally coherent action sequences (Zhao et al., 2023; Chi et al., 2023). Q-Chunking extends action chunking to reinforcement learning by learning values over complete action sequences (Li et al., 2025), and recent variants improve policy reactivity or combine chunk-level and single-step value learning (Li et al., 2026; Song et al., 2026). WAVE is complementary: it keeps the temporal abstraction and studies how a world model can improve chunk-value learning.

World-model predictions in value targets. World models have been used for latent planning, imagined rollouts, and model-predictive control (Hafner et al., 2019; 2020; Hansen et al., 2022; 2024; Schrittwieser et al., 2020). When predictions enter training data or value targets, model error enters with them; model-based value expansion and MBPO therefore limit rollout horizons or weight model-based targets by uncertainty (Feinberg et al., 2018; Buckman et al., 2018; Janner et al., 2019). Theory bounds how model error propagates to values (Kearns & Singh, 2002), and work on objective mismatch and value-aware models argues that models should be trained for the decisions they support (Farahmand et al., 2017; Grimm et al., 2020; Lambert et al., 2020). Lemma 1 (Section 4.4) is a chunk-level instance of this sensitivity, in which an endpoint error of size ε_H suffices to reverse the chunk ranking; WAVE avoids it by keeping predictions out of the target rather than making them more accurate.

Predictive representations with real-data targets. A second line uses prediction to shape the representation while value targets stay on real transitions. Value Prediction Networks and the Predictron learn value-oriented predictive models (Oh et al., 2017; Silver et al., 2017); SPR, TD-MPC, SALE, and MR.Q train encoders of states or state–action pairs to predict their own future latent states (Schwarzer et al., 2021; Hansen et al., 2022; Fujimoto et al., 2023; 2025). WAVE belongs to this line but differs in three respects: its targets come from a separately trained world model that receives no critic gradients, the target is the consequence of the entire H-step chunk being valued, and the critic is also conditioned on the model's history posterior, which relates it to recurrent model-free RL (Ni et al., 2022).

Offline-to-online reinforcement learning. Offline-to-online RL studies how prior datasets can accelerate online adaptation through policy regularization, replay design, conservative initialization, or improved optimization (Nair et al., 2020; Lee et al., 2022; Ball et al., 2023; Nakamoto et al., 2023). WAVE instead uses an offline-pretrained world model to give an action-chunk critic temporal context and predictive supervision, without a new replay strategy or model-generated targets.

3 Problem Setup

Offline-to-online chunked RL. The agent is pretrained on a fixed dataset $\mathcal{D}_{\text{off}}$ and then adapts with a limited budget of real interactions, stored with the offline data in the replay buffer $\mathcal{D} = \mathcal{D}_{\text{off}} \cup \mathcal{D}_{\text{on}}$; no model-generated transitions enter $\mathcal{D}$. Observations are low-dimensional states: 46-D on Cube and 40-D on Scene. Every H steps, the agent observes $\mathbf{o}_t$, selects an action chunk $\mathbf{c}_t = [\mathbf{a}_t^\top, \dots, \mathbf{a}_{t+H-1}^\top]^\top$ that is executed open-loop, and receives the discounted chunk reward $R_t^{(H)} = \sum_{j=0}^{H-1} \gamma^j r_{t+j}$. Q-Chunking (QC) (Li et al., 2025) trains a chunk critic $Q_{\boldsymbol{\theta}}(\mathbf{o}_t, \mathbf{c}_t)$ by minimizing the squared error $\mathcal{L}_Q$ to the target

$$y_t = R_t^{(H)} + \gamma^H m_{t+H}\, \bar{Q}_{\bar{\boldsymbol{\theta}}}\big(\mathbf{o}_{t+H}, \mathbf{c}'_{t+H}\big), \tag{1}$$

where $m_{t+H} \in \{0, 1\}$ is zero if the episode terminates within the chunk, $\bar{Q}_{\bar{\boldsymbol{\theta}}}$ is the arithmetic mean of two target critics, and $\mathbf{c}'_{t+H}$ is the highest-valued of N chunks proposed by the actor $\pi(\cdot \mid \mathbf{o}_{t+H})$ under the mean of the two current critics; the same best-of-N rule selects chunks during control.

Chunk-level world model. We use a recurrent state-space model (Hafner et al., 2019; 2020) that transitions once per chunk rather than once per primitive action (Appendix C). It provides a causal posterior of real history and a prior prediction one chunk ahead:

$$\mathbf{z}_t = e_{\boldsymbol{\phi}}\big(\mathbf{o}_{\le t}, \mathbf{a}_{<t}\big), \quad \mathbf{z}^p_{t+H} = f_{\boldsymbol{\phi}}\big(\mathbf{z}_t, \mathbf{c}_t\big). \tag{2}$$

The posterior $\mathbf{z}_t$ uses only observations and actions up to t, whereas the prior $\mathbf{z}^p_{t+H}$ predicts the chunk's latent outcome without observing $\mathbf{o}_{t+H}$. Besides its latent dynamics, the model decodes the next boundary observation and predicts the chunk return and a task-progress signal defined by task-specific completion predicates at the chunk endpoint (Appendix C.4), so its latent state is also trained with task-relevant predictive supervision. The world model is trained with its own objective and receives no gradients from the critic.

4 WAVE: World-Model-Augmented Value Learning

4.1 Overview

WAVE augments a pretrained QC critic with two world-model signals while keeping its Bellman target on real replay (Figure 2): the causal posterior $\mathbf{z}_t$ enters through a zero-initialized residual adapter (Section 4.2), and the H-step prior $\mathbf{z}^p_{t+H}$ of the chunk being valued supervises the critic's representation (Section 4.3); Section 4.4 analyzes why neither enters the backup.

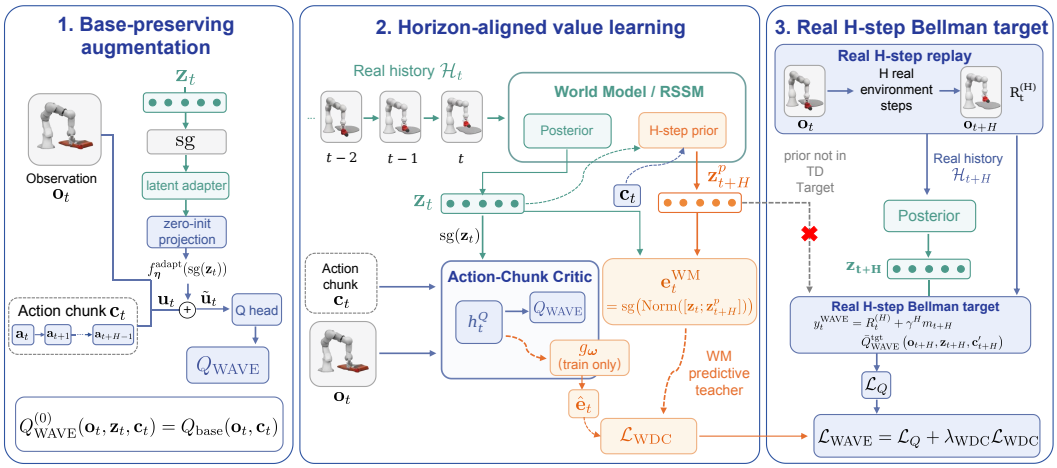


Figure 2: **WAVE architecture.** **(1)** A zero-initialized adapter adds the causal posterior to the pretrained critic features. **(2)** A training-only head predicts the WDC teacher built from the posterior and the H-step prior. **(3)** Bellman targets use real H-step replay; the prior never enters the TD target.

4.2 Causal World-Model Conditioning

The base critic sees only $\mathbf{o}_t$, whereas the posterior $\mathbf{z}_t$ summarizes real history in a representation trained to predict chunk outcomes. Concatenating $\mathbf{z}_t$ to the critic input would change the pretrained critic's outputs before any online data are collected, so we add it through a residual adapter:

$$\mathbf{u}_t = f^{\text{base}}_{\boldsymbol{\theta}}\big(\mathbf{o}_t, \mathbf{c}_t\big), \quad \tilde{\mathbf{u}}_t = \mathbf{u}_t + f^{\text{adapt}}_{\boldsymbol{\eta}}\big(\text{sg}(\mathbf{z}_t)\big), \tag{3}$$

where $f^{\text{adapt}}_{\boldsymbol{\eta}}$ is a bottleneck adapter whose output dimension matches the base critic feature $\mathbf{u}_t$, $\text{sg}(\cdot)$ denotes stop-gradient, and the remaining critic layers map $\tilde{\mathbf{u}}_t$ to $Q_{\text{WAVE}}(\mathbf{o}_t, \mathbf{z}_t, \mathbf{c}_t)$.

At the offline-to-online transition, we keep the pretrained critic parameters and zero-initialize the adapter's output projection and bias, as zero-initialized branches do when adding conditions to pretrained generative models (Zhang et al., 2023). The residual therefore vanishes, and for every $\mathbf{z}_t$,

$$Q^{(0)}_{\text{WAVE}}(\mathbf{o}_t, \mathbf{z}_t, \mathbf{c}_t) = Q_{\text{base}}(\mathbf{o}_t, \mathbf{c}_t). \tag{4}$$

Posterior conditioning thus starts from the offline value function instead of perturbing it, and the stop-gradient keeps value-learning updates out of the RSSM parameters ϕ.

WAVE keeps QC's best-of-N rule but ranks candidates with the posterior-conditioned critic: the observation-conditioned actor, which does not receive $\mathbf{z}_t$, proposes $\{\mathbf{c}_t^{(n)}\}_{n=1}^N$, and the agent selects

$$n_t^* = \arg\max_{1\le n\le N} \bar{Q}_{\mathrm{WAVE}}(\mathbf{o}_t, \mathbf{z}_t, \mathbf{c}_t^{(n)}), \quad \mathbf{c}_t = \mathbf{c}_t^{(n_t^*)}, \tag{5}$$

where $\bar{Q}_{\mathrm{WAVE}}$ aggregates the current critic ensemble.

4.3 World-Model Dynamics Consistency

The posterior $\mathbf{z}_t$ does not depend on the candidate chunk $\mathbf{c}_t$, so it cannot tell the critic how different chunks from the same history differ in outcome; without further supervision, that distinction is learned from TD targets alone. The prior $\mathbf{z}_{t+H}^p$ (Equation 2) does depend on $\mathbf{c}_t$, so we use it as an auxiliary target for the critic features $\mathbf{h}_t^Q$ shared by the value readout and a prediction head:

$$\mathbf{e}_t^{\mathrm{WM}} = \mathrm{sg}\big(\mathrm{Norm}([\mathbf{z}_t; \mathbf{z}_{t+H}^p])\big), \quad \hat{\mathbf{e}}_t = g_{\boldsymbol{\omega}}(\mathbf{h}_t^Q), \tag{6}$$

where $\mathbf{h}_t^Q$ is the final 512-dimensional hidden representation before the scalar value readout, $g_{\boldsymbol{\omega}}$ is a training-only prediction head, and Norm standardizes the deterministic RSSM components with fixed training-set statistics. The teacher pairs the current posterior with the chunk's predicted consequence at the same horizon as the chunk value, so matching it requires features that distinguish chunks by their predicted outcomes.

We call this objective *world-model dynamics consistency* (WDC). For both the current (cur) and predicted (prior) parts, deterministic components are matched by mean squared error and categorical components by the forward KL $D_{\mathrm{KL}}(p_{\mathrm{teacher}} \| p_{\mathrm{critic}})$:

$$\mathcal{L}_{\mathrm{WDC}} = \mathbb{E}_{\mathcal{D}}\left[\frac{1}{4}\sum_{k\in\{\mathrm{cur},\mathrm{prior}\}} \left(\ell_{\mathrm{MSE}}^k + \ell_{\mathrm{KL}}^k\right)\right]. \tag{7}$$

The critic is optimized with $\mathcal{L}_{\mathrm{WAVE}} = \mathcal{L}_Q + \lambda_{\mathrm{WDC}}\mathcal{L}_{\mathrm{WDC}}$, where $\mathcal{L}_Q$ is the QC loss (Section 3) and λ_{WDC} is calibrated once before online learning so that the WDC gradient on the shared critic representation is approximately 5% of the value gradient (Appendix B.4). The prior and the prediction head are used only in this loss, not for action selection or bootstrapping.

4.4 Real-Replay Value Learning

A predicted endpoint can make an unproductive chunk appear to reach a high-value state, and a Bellman backup through that prediction can then favor the wrong chunk even with exact rewards and exact value optimization. We make this concrete with a minimal construction, where f_H denotes the true H-step environment transition and $\hat{f}_H$ the world model's endpoint prediction; the predictor is kept fixed to isolate transition error from optimization error.

Endpoint-substitution construction. Consider two chunk-boundary states $\mathbf{o}_0 = \mathbf{0}$ and $\mathbf{o}_1 = \varepsilon_H \mathbf{v}$ with $\|\mathbf{v}\|_2 = 1$, and two candidate chunks $\mathbf{c}^+$ and $\mathbf{c}^-$, both of which leave the true boundary state unchanged. At $\mathbf{o}_0$, $\mathbf{c}^+$ earns a discounted chunk return $r_0 > 0$ and $\mathbf{c}^-$ earns nothing; at $\mathbf{o}_1$, both chunks earn $r_1 > r_0$. The predictor is exact except that $\hat{f}_H(\mathbf{o}_0, \mathbf{c}^-) = \mathbf{o}_1$: it wrongly predicts that the zero-reward chunk reaches the more valuable state.

Lemma 1 (Policy loss from predicted endpoints)**.** *For any $0 < \gamma < 1$, integer $H \ge 1$, $l_v > 0$, and $0 < \varepsilon_H \le 1/l_v$, set $\beta = \gamma^H$, $d = l_v\varepsilon_H$, $r_0 = \beta d/2$, and $r_1 = (1-\beta/2)d$ in the construction above. Then the discounted chunk returns lie in $[0,1]$, the true optimal value V^* is l_v-Lipschitz on the boundary states, and the predictor's worst-case endpoint error is $\sup_{\mathbf{o},\mathbf{c}} \|\hat{f}_H(\mathbf{o},\mathbf{c}) - f_H(\mathbf{o},\mathbf{c})\|_2 = \varepsilon_H$. Let $\widehat{Q}$ be the exact fixed point of the Bellman backup that substitutes $\hat{f}_H$ for f_H while retaining the real chunk returns, and let π_{D} be its greedy policy. Then π_{D} selects $\mathbf{c}^-$ at $\mathbf{o}_0$ and, evaluated in the real environment, incurs the strictly positive loss*

$$V^*(\mathbf{o}_0) - V^{\pi_{\mathrm{D}}}(\mathbf{o}_0) = \frac{\gamma^H l_v \varepsilon_H}{2(1-\gamma^H)} > 0. \tag{8}$$

Unlike upper bounds on the effect of model error (Kearns & Singh, 2002), this loss is attained: π_{D} repeatedly selects $\mathbf{c}^-$, stays at $\mathbf{o}_0$, and receives no reward. For instance, with $\gamma^H = 0.9$, $l_v = 10$, and $\varepsilon_H = 0.1$, substitution values $\mathbf{c}^-$ above $\mathbf{c}^+$ at $\mathbf{o}_0$ (4.95 vs. 4.905), whereas the real values are 4.05 and 4.5 (Appendix A.1). Direct-WM (Section 5.3) applies the same substitution with the learned world model.

WAVE instead keeps the observed endpoint in every value target:

$$y_t^{\mathrm{WAVE}} = R_t^{(H)} + \gamma^H m_{t+H} \bar{Q}^{\mathrm{tgt}}_{\mathrm{WAVE}}\big(\mathbf{o}_{t+H}, \mathbf{z}_{t+H}, \mathbf{c}'_{t+H}\big), \tag{9}$$

where $\mathbf{o}_{t+H}$ is the real replay endpoint, $\mathbf{z}_{t+H}$ is its posterior inferred from real history, and $\mathbf{c}'_{t+H}$ is selected as in Equation 1.

Proposition 2 (Real-replay targets do not depend on the world model)**.** *Let the chunk-boundary observation be a Markov state, and let the posterior $\mathbf{z}_t$ come from an arbitrary, possibly inaccurate world model ϕ. Let $\mathcal{T}_{\mathrm{W}}$ be the backup with sample target Equation 9 and exact maximization over next chunks, acting on bounded functions of the real history and chunk (these include all critics $Q(\mathbf{o}, \mathbf{z}, \mathbf{c})$). Then, for every ϕ, (i) the unique fixed point of $\mathcal{T}_{\mathrm{W}}$ is the optimal chunk value $Q^*(\mathbf{o}, \mathbf{c})$ of the real environment, whose greedy policy is optimal; and (ii) the WAVE critic class contains every base critic: with the zero-initialized adapter output, $Q_{\mathrm{WAVE}}(\mathbf{o}, \mathbf{z}, \mathbf{c}) = Q_{\mathrm{base}}(\mathbf{o}, \mathbf{c})$ for all $\mathbf{z}$.*

In the construction of Lemma 1, the observed endpoint thus restores $Q^*(\mathbf{o}_0, \mathbf{c}^+) = 4.5 > Q^*(\mathbf{o}_0, \mathbf{c}^-) = 4.05$ whatever the predictor (proofs in Appendix A). The world model therefore acts on WAVE only through the critic's inputs and the WDC signal, which shape the approximation the critic reaches and how quickly it gets there; Section 5 measures this effect.

4.5 Offline-to-Online Optimization

Offline stage. QC and the chunk-level RSSM are pretrained separately on $\mathcal{D}_{\mathrm{off}}$. At the offline-to-online transition, we attach the zero-initialized adapter and the WDC head while keeping the pretrained QC parameters and optimizer states, so that $Q^{(0)}_{\mathrm{WAVE}} = Q_{\mathrm{base}}$ (Equation 4). Before any interaction, we warm up only the WDC head on $\mathcal{D}_{\mathrm{off}}$ for 10,000 updates with all other parameters frozen and then calibrate λ_{WDC} once (Appendix B.4).

Online stage. At each decision boundary, the RSSM infers $\mathbf{z}_t$ from real history and the posterior-conditioned critic selects among the actor's candidates by Equation 5. The executed transitions are added to $\mathcal{D}$, and minibatches from $\mathcal{D}$ update the critic, adapter, and WDC head with $\mathcal{L}_{\mathrm{WAVE}}$ and the target in Equation 9; actor and target-critic updates follow QC. The RSSM is updated separately on offline and online sequences with its own objective and optimizer; critic gradients never reach it, no imagined transitions enter replay, and all budgets count real environment steps (Algorithm 1 in Appendix B).

5 Experiments

Our experiments answer three questions. **Q1:** Does WAVE improve online sample efficiency over QC and other chunk-based learners (Section 5.2)? **Q2:** Does it matter whether world-model predictions enter the critic as features or as Bellman endpoints (Section 5.3)? **Q3:** Which WAVE components and predictive targets matter, and how does their benefit change with the amount of offline data (Section 5.4)?

5.1 Experimental Setup

Tasks. We evaluate on two OGBench manipulation environments (Park et al., 2025a) that require temporally extended, coordinated action sequences (Appendix D.1). *Cube* (`cube-triple-play-singletask-task5-v0`) requires stacking three cubes in sequence and has a 10M-transition offline data set; *Scene* (`scene-play-singletask-task5-v0`, *rearrange-hard*) requires manipulating a cube, drawer, window, and two locks in sequence and has 1M transitions.

Baselines. We compare with Flow Q-Learning (FQL) (Park et al., 2025b), the underlying QC learner (Li et al., 2025), Decoupled Q-Chunking (DQC) (Li et al., 2026), and Chunk-Guided Q-Learning (CGQ) (Song et al., 2026). The released DQC implementation is offline-only, so we adapt it to our reward-based offline-to-online protocol (Appendix D.6). Direct-WM shares WAVE's RSSM checkpoint, replay, and online world-model updates, but replaces the observed endpoint with the decoded H-step prediction throughout QC's bootstrap computation, as in Lemma 1; it has no adapter or WDC, and unlike STEVE (Buckman et al., 2018) it does not weight model-based targets by uncertainty (Appendix E). Published baselines retain their released method configurations where applicable, with only the adaptations required for the common offline-to-online protocol (Appendix D.7), and are not tuned on the evaluation tasks.

Protocol and metrics. All methods receive 10^6 offline updates followed by 10^6 real online interactions; QC, Direct-WM, and WAVE use $H = 5$ and $\gamma = 0.99$. RSSM pretraining and WDC-head warm-up are additional to this budget, and evaluation episodes are excluded from replay and the interaction budget. We report AUC_T, the normalized trapezoidal area under the success curve up to T real interactions; S_T, the success rate at T; Return, the mean episode return of the final evaluation; and T_{80}, the first evaluation checkpoint reaching at least 80% success, without interpolation. Evaluations occur every 5k interactions through 50k, every 50k through 200k, and every 100k thereafter; intermediate evaluations use 50 episodes and the final evaluation uses 200. The Q3 ablations are separate runs with a 200k-interaction budget, so their values need not match Figures 3 and 4 at the same interaction count; predictive-target and data-scale studies use Cube, and the component ablation is also run on Scene. All results are mean ± sample standard deviation over three seeds; for comparisons with WAVE, we also report two-sided Welch's t-tests.

5.2 Q1: Does WAVE Improve Online Sample Efficiency?

Table 1: **Main offline-to-online results.** Mean ± sample standard deviation over three seeds after 10^6 offline updates and 10^6 real online interactions. Direct-WM and WAVE build on QC and use the same world model. Bold marks the best mean in each column. †Adapted to the offline-to-online protocol (Appendix D.6).

	Cube			Scene		
Method	$\mathrm{AUC}_{1\mathrm{M}}\uparrow$	$\mathrm{S}_{1\mathrm{M}}\uparrow$	Return ↑	$\mathrm{AUC}_{1\mathrm{M}}\uparrow$	$\mathrm{S}_{1\mathrm{M}}\uparrow$	Return ↑
FQL	0.000±0.000	0.000±0.000	−2874±213	0.869±0.013	0.888±0.088	−795±89
DQC†	0.001±0.001	0.000±0.000	−2984±5	0.008±0.007	0.007±0.012	−2161±75
CGQ	0.873±0.018	0.798±0.099	−628±200	0.072±0.030	0.000±0.000	−875±6
QC	0.832±0.019	0.935±0.007	−438±30	0.922±0.003	0.957±0.013	**−609**±44
Direct-WM	0.000±0.000	0.000±0.000	−2993±0	0.043±0.016	0.003±0.004	−1586±36
WAVE (ours)	**0.929**±0.042	**0.968**±0.011	**−342**±35	**0.956**±0.009	**0.965**±0.028	−616±59

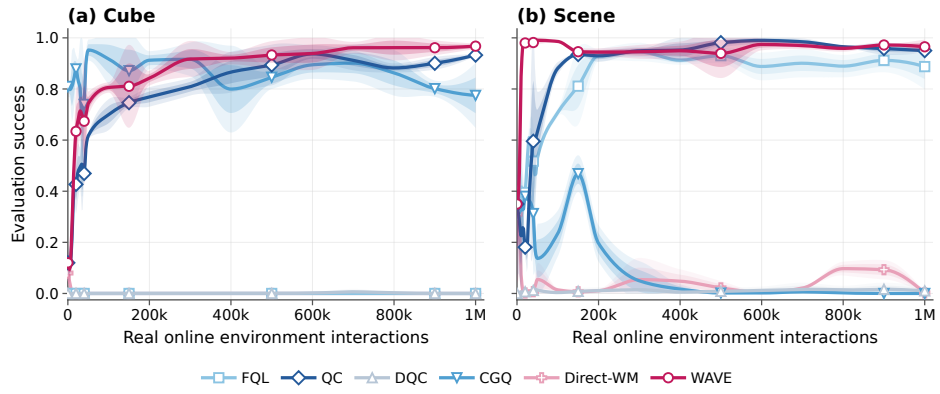


Figure 3: **Evaluation success during online learning.** Lines show the mean over three seeds as a function of real online interactions; shaded regions show the variability across seeds.

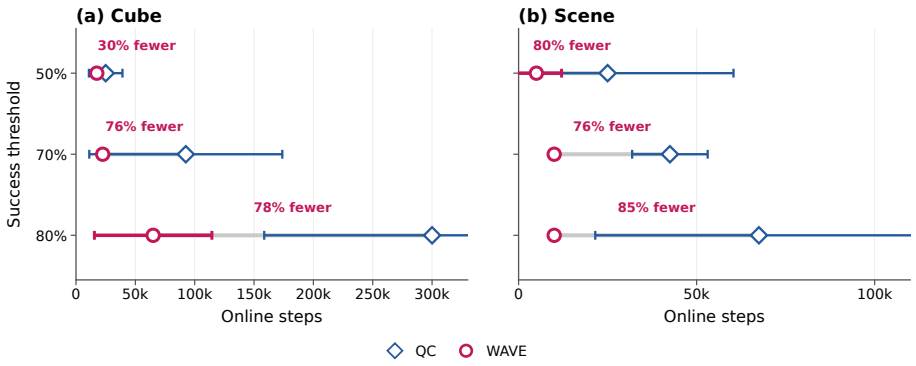


Figure 4: **Real interactions needed to first reach 50%, 70%, and 80% success.** Markers show three-seed means and bars sample standard deviations; annotations give WAVE's reduction relative to QC. Thresholds use evaluation checkpoints without interpolation; panels use different x-axis ranges.

WAVE improves over QC on both tasks. WAVE raises $\mathrm{AUC}_{1\mathrm{M}}$ from 0.832 to 0.929 on Cube and from 0.922 to 0.956 on Scene (Table 1; $p = 0.04$ and $p = 0.015$). On Cube, it also attains the highest final success and return. On Scene, where QC already reaches 0.957 final success, the gain is in learning speed: WAVE's success curve rises much earlier (Figure 3), and its final success and return are on par with QC's (0.965 vs. 0.957 and -616 vs. -609).

WAVE reaches high success with fewer interactions. Averaged over seeds, WAVE needs 76–85% fewer real interactions than QC to first reach 70% and 80% success on both tasks, and 30% (Cube) and 80% (Scene) fewer to reach 50% (Figure 4). The largest absolute saving is at the 80% threshold on Cube, where QC's crossing also varies most across seeds.

No other baseline is strong on both tasks. CGQ is competitive with WAVE in $\mathrm{AUC}_{1\mathrm{M}}$ on Cube (0.873; $p = 0.13$), but its success declines late in training to 0.798, and on Scene it peaks below 0.5 and then collapses (Figure 3). FQL, which does not use action chunks, makes no progress on Cube, in line with its weak cube-triple results in Li et al. (2025), but reaches 0.888 final success on Scene. Our offline-to-online adaptation of DQC, which replaces its goal-conditioned reward pipeline with the task reward, fails on both tasks (Appendix D.6).

5.3 Q2: Predictions as Features or as Bellman Endpoints?

Direct-WM differs from QC only in the bootstrap path, where the decoded prediction replaces the observed endpoint for next-chunk proposal, candidate selection, and target evaluation (Appendix E); WAVE differs from QC only in how the critic uses the world model, so comparing each with QC separates the two uses of the same model. Used as a Bellman endpoint, the model lowers $\mathrm{AUC}_{1\mathrm{M}}$ from 0.832 to 0.000 on Cube and from 0.922 to 0.043 on Scene, with near-zero final success; used as critic features, it raises $\mathrm{AUC}_{1\mathrm{M}}$ to 0.929 and 0.956 (Table 1). The substituted endpoints have mean replay-batch MSEs of 0.102 on Cube and 0.046 on Scene (Appendix E.4), and substituting them into the bootstrap path is enough to make QC fail, the failure mode that Lemma 1 isolates in a minimal construction. In Direct-WM, the actor and critics are also queried at the decoded observations, so the endpoint error acts together with the resulting shift in their inputs.

5.4 Q3: Which Components and Predictive Targets Matter, and When?

Future-consequence supervision accounts for most of the WDC gain. With 10M offline transitions, posterior conditioning raises $\mathrm{AUC}_{200\mathrm{k}}$ from 0.612 to 0.687 (Table 2a). Future WDC raises it further to 0.736 ± 0.013 (one-step) and 0.735 ± 0.003 (H-step), and full WAVE reaches 0.742 ± 0.054, whereas current-state WDC adds nothing (0.675 ± 0.028), which we attribute to the critic already receiving $\mathbf{z}_t$ through the adapter. T_{80} follows the same ordering: QC reaches 80% success on no seed within 200k interactions, posterior conditioning takes 115.0k, current-state WDC succeeds on two

Table 2: **Ablations on Cube** (200k online interactions; mean ± sample standard deviation over three seeds). T_{80}: parentheses give the number of seeds reaching 80% when fewer than three do, and the mean is over those seeds; NR: reached by no seed. Δ: seed-paired difference between full WAVE and QC.

(a) Components and predictive targets, 10M offline transitions

Variant	$\mathrm{AUC}_{50k}\uparrow$	$\mathrm{AUC}_{200k}\uparrow$	$\mathrm{S}_{200k}\uparrow$	T_{80} (k) $\downarrow$
QC	0.416±0.019	0.612±0.048	0.730±0.077	NR
Posterior only	0.475±0.011	0.687±0.024	0.830±0.026	115.0±60.6
Posterior + current WDC	0.459±0.066	0.675±0.028	0.833±0.043	150.0±0.0 (2/3)
Posterior + 1-step future WDC	0.448±0.032	0.736±0.013	0.870±0.010	65.0±30.4
Posterior + H-step future WDC	0.504±0.058	0.735±0.003	0.855±0.036	65.0±30.4
Full WAVE	**0.519**±0.047	**0.742**±0.054	**0.888**±0.042	**56.7**±37.9

(b) Offline-data scale, $\mathrm{AUC}_{200k}\uparrow$

Offline data	QC	Posterior only	Full WAVE	Δ (WAVE − QC)
10M	0.612±0.048	0.687±0.024	**0.742**±0.054	+0.130±0.059
30M	0.746±0.012	0.760±0.060	**0.815**±0.017	+0.069±0.022

of three seeds, and both future targets take 65.0k (full WAVE: 56.7k). The components compound on Scene as well: QC, posterior conditioning, and full WAVE reach $\mathrm{AUC}_{200k} = 0.699 \pm 0.045$, 0.732 ± 0.088, and 0.943 ± 0.023.

The H-step target matches the one-step target with a single world model. The H-step target reaches the same AUC_{200k} as the one-step target (0.735 vs. 0.736) and a higher mean AUC_{50k} (0.504 ± 0.058 vs. 0.448 ± 0.032). It comes from the chunk-level RSSM that already provides the posterior, whereas the one-step target requires training a second, primitive-step RSSM (Appendix D.4).

WAVE improves over QC at both offline-data scales. With 30M offline transitions, QC itself is stronger ($\mathrm{AUC}_{200k} = 0.746$ vs. 0.612 at 10M), and WAVE still improves on it on every seed, by $+0.069 \pm 0.022$ compared with $+0.130 \pm 0.059$ at 10M (Table 2b). In mean AUC_{200k}, the smaller gain comes from posterior conditioning, whose increment over QC falls from 0.075 to 0.014, whereas WDC adds 0.055 on top of it at both scales; with 30M transitions, consequence supervision thus provides most of WAVE's gain.

6 Conclusion

We studied how a world model can improve action-chunk value learning without replacing real transitions in the Bellman backup. A small endpoint error can reverse the ranking of chunks when predictions serve as bootstrap endpoints, whereas real-replay targets keep the optimal fixed point for any world model. WAVE accordingly uses the model to condition and supervise the critic's representation while keeping every target on real replay. On two OGBench tasks, it reaches 70% and 80% success with 76–85% fewer real interactions than Q-Chunking, whereas substituting endpoints from the same world model yields near-zero success, and its ablations trace the auxiliary gain to predicting future latent states. These results support a simple principle: predict with the model, but bootstrap from reality.

Limitations. Our experiments cover two manipulation tasks with low-dimensional states and instantiate WAVE on QC; extending it to image observations, a broader task suite, and other chunk critics such as DQC and CGQ is a natural next step. The RSSM is trained with reward and task-progress heads in addition to its dynamics, and separating how dynamics, reward, and history information each contribute to the posterior's benefit would guide the design of these world-model signals. WAVE also adds the cost of training and updating a world model (Appendix D.8).

Ethics Statement

This work studies reinforcement learning on simulated robotic manipulation tasks from the OGBench benchmark. It involves no human subjects, personal data, or newly collected data sets. More sample-efficient robot learning could eventually support systems deployed in the physical world, which would require their own safety evaluation; we are not aware of ethical concerns specific to the proposed method.

Reproducibility Statement

Section 4 and Appendices B and C specify the critic augmentation, the WDC objective and its calibration, the training procedure (Algorithm 1), and the chunk-level world model. Section 5.1 and Appendix D give the tasks, data sets, protocol, evaluation schedule, metrics, and hyperparameters, and Appendices D.6 and E describe the adapted DQC baseline and Direct-WM. Complete statements and proofs of the analysis are in Appendix A. We will release the code and experiment configurations upon publication.

AI Use Statement

In this work, we used generative AI tools to formulate mathematical claims and assist in writing their proofs (Proposition 2 and its proof), to provide feedback on research methodology and experiments, to assist with implementation and debugging, and to interpret results. We also used generative AI tools to assist in drafting portions of the manuscript and to edit the paper for readability and consistency, as well as to format references. We did not use generative AI tools to generate synthetic data sets; translation, data set cleaning and reformatting, and qualitative or thematic data analysis are not applicable to this work. We have reviewed all AI-assisted work: the authors checked every mathematical statement and proof, verified each reported number against the experiment logs, and checked each cited reference against the original publication. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

References

Philip J. Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1577–1594. PMLR, 2023.

Jacob Buckman, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee. Sample-efficient reinforcement learning with stochastic ensemble value expansion. In *Advances in Neural Information Processing Systems*, volume 31, pp. 8234–8244, 2018.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin C. M. Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.026.

Amir-Massoud Farahmand, André Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1486–1494. PMLR, 2017.

Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I. Jordan, Joseph E. Gonzalez, and Sergey Levine. Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*, 2018.

Scott Fujimoto, Wei-Di Chang, Edward Smith, Shixiang (Shane) Gu, Doina Precup, and David Meger. For SALE: State-action representation learning for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 61573–61624, 2023. doi: 10.52202/075280-2691.

Scott Fujimoto, Pierluca D'Oro, Amy Zhang, Yuandong Tian, and Michael Rabbat. Towards general-purpose model-free reinforcement learning. In *International Conference on Learning Representations*, pp. 83509–83552, 2025.

Christopher Grimm, André Barreto, Satinder P. Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5541–5552, 2020.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, pp. 47376–47405, 2024.

Nicklas A. Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8387–8406. PMLR, 2022.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2–3):209–232, 2002. doi: 10.1023/A:1017984413808.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pp. 761–770. PMLR, 2020.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic Q-ensemble. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1702–1712. PMLR, 2022.

Qiyang Li, Zhiyuan (Paul) Zhou, and Sergey Levine. Reinforcement learning with action chunking. In *Advances in Neural Information Processing Systems*, volume 38, pp. 55518–55553, 2025. doi: 10.52202/085713-1856.

Qiyang Li, Seohong Park, and Sergey Levine. Decoupled Q-chunking. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=aqGNdZQL9l.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-QL: Calibrated offline RL pre-training for efficient online fine-tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 62244–62269, 2023. doi: 10.52202/075280-2719.

Tianwei Ni, Benjamin Eysenbach, and Ruslan Salakhutdinov. Recurrent model-free RL can be a strong baseline for many POMDPs. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16691–16723. PMLR, 2022.

Junhyuk Oh, Satinder Singh, and Honglak Lee. Value prediction network. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6118–6128, 2017.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations*, pp. 94937–94982, 2025a.

Seohong Park, Qiyang Li, and Sergey Levine. Flow Q-learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48104–48127. PMLR, 2025b.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588:604–609, 2020. doi: 10.1038/s41586-020-03051-4.

Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021.

David Silver, Hado van Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David Reichert, Neil Rabinowitz, André Barreto, and Thomas Degris. The Predictron: End-to-end learning and planning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3191–3199. PMLR, 2017.

Gwanwoo Song, Kwanyoung Park, and Youngwoon Lee. Chunk-guided Q-learning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=yojuz84NLb.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1–2):181–211, 1999. doi: 10.1016/S0004-3702(99)00052-1.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023. doi: 10.1109/ICCV51070.2023.00355.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.016.

APPENDIX CONTENTS

A Proofs for Section 4.4 14
B Additional Method Details 16
C World-Model Architecture and Training 17
D Experimental Setup and Hyperparameters 19
E Direct-WM Baseline 22

A Proofs for Section 4.4

A.1 Policy Loss from Endpoint Substitution

Proof of Lemma 1. We use the endpoint-substitution construction of Section 4.4 with $\beta = \gamma^H$, $d = l_v \varepsilon_H \le 1$, and chunk returns

$$\begin{aligned} r_H(\mathbf{o}_0, \mathbf{c}^+) &= r_0 = \frac{\beta d}{2}, \\ r_H(\mathbf{o}_0, \mathbf{c}^-) &= 0, \\ r_H(\mathbf{o}_1, \mathbf{c}^+) = r_H(\mathbf{o}_1, \mathbf{c}^-) &= r_1 = \left(1 - \frac{\beta}{2}\right) d, \end{aligned} \tag{10}$$

where r_H denotes the deterministic discounted chunk return. Since $d \le 1$, we have $0 < r_0 \le 1/2$ and $0 < r_1 \le 1$, so all returns lie in $[0, 1]$, and $r_1 - r_0 = (1-\beta)d > 0$. This chunk process can be realized at the primitive timescale by awarding the chunk return on its first step and inserting $H - 1$ forced, zero-reward steps before the next boundary.

Endpoint error. Both chunks leave the true boundary state unchanged, so $f_H(\mathbf{o}, \mathbf{c}) = \mathbf{o}$ for every pair. The predictor agrees with f_H except at $(\mathbf{o}_0, \mathbf{c}^-)$, where it returns $\mathbf{o}_1$; its worst-case error is therefore $\|\mathbf{o}_1 - \mathbf{o}_0\|_2 = \varepsilon_H$. Both states are valid real states, and replay may cover both states and both chunks, so the failure below is not caused by missing data.

True values. Because the true state never changes, the optimal policy selects $\mathbf{c}^+$ at $\mathbf{o}_0$ and either chunk at $\mathbf{o}_1$, giving

$$V^*(\mathbf{o}_0) = \frac{r_0}{1-\beta}, \qquad V^*(\mathbf{o}_1) = \frac{r_1}{1-\beta}. \tag{11}$$

Their difference is $(r_1 - r_0)/(1-\beta) = d = l_v \|\mathbf{o}_1 - \mathbf{o}_0\|_2$, so V^* is l_v-Lipschitz on the boundary states.

Substituted fixed point. The model-substituted Bellman operator is

$$\begin{aligned} (\widehat{\mathcal{T}}_H Q)(\mathbf{o}, \mathbf{c}) &= r_H(\mathbf{o}, \mathbf{c}) \\ &\quad + \beta \max_{\mathbf{c}'} Q\big(\hat{f}_H(\mathbf{o}, \mathbf{c}), \mathbf{c}'\big). \end{aligned} \tag{12}$$

This operator is a β-contraction, so it has a unique fixed point $\widehat{Q}$; write $\widehat{V}(\mathbf{o}) = \max_{\mathbf{c}} \widehat{Q}(\mathbf{o}, \mathbf{c})$. We verify that the following values satisfy $Q = \widehat{\mathcal{T}}_H Q$. The predictor is exact at $\mathbf{o}_1$, so $\widehat{Q}(\mathbf{o}_1, \mathbf{c}^\pm) = \widehat{V}(\mathbf{o}_1) = r_1/(1-\beta)$. At $\mathbf{o}_0$, set $\widehat{V}(\mathbf{o}_0) = \beta r_1/(1-\beta)$, so that

$$\widehat{Q}(\mathbf{o}_0, \mathbf{c}^-) = \beta \widehat{V}(\mathbf{o}_1) = \frac{\beta r_1}{1-\beta}, \qquad \widehat{Q}(\mathbf{o}_0, \mathbf{c}^+) = r_0 + \beta \widehat{V}(\mathbf{o}_0) = r_0 + \frac{\beta^2 r_1}{1-\beta}.$$

Then

$$\begin{aligned} &\widehat{Q}(\mathbf{o}_0, \mathbf{c}^-) - \widehat{Q}(\mathbf{o}_0, \mathbf{c}^+) \\ &\quad = \beta r_1 - r_0 = \frac{\beta(1-\beta)d}{2} > 0, \end{aligned} \tag{13}$$

so $\max_{\mathbf{c}} \widehat{Q}(\mathbf{o}_0, \mathbf{c}) = \widehat{Q}(\mathbf{o}_0, \mathbf{c}^-) = \widehat{V}(\mathbf{o}_0)$, consistent with the assumed value. These values therefore satisfy the Bellman equation at every state–chunk pair and, by uniqueness, equal $\widehat{Q}$. Thus, π_{D} always selects $\mathbf{c}^-$ at $\mathbf{o}_0$.

Real return of π_{D}. In the real environment, $\mathbf{c}^-$ keeps the state at $\mathbf{o}_0$ with zero reward, so $V^{\pi_{\mathrm{D}}}(\mathbf{o}_0) = 0$ and

$$V^*(\mathbf{o}_0) - V^{\pi_{\mathrm{D}}}(\mathbf{o}_0) = \frac{r_0}{1-\beta} = \frac{\gamma^H l_v \varepsilon_H}{2(1-\gamma^H)}. \tag{14}$$

□

Table 3: Chunk values at $\mathbf{o}_0$ in the numerical illustration.

Chunk	Observed endpoints (WAVE target): Q^*	Predicted endpoints: $\widehat{Q}$
$\mathbf{c}^+$	$0.45 + 0.9 \times 4.5 = 4.5$	$0.45 + 0.9 \times 4.95 = 4.905$
$\mathbf{c}^-$	$0 + 0.9 \times 4.5 = 4.05$	$0 + 0.9 \times 5.5 = 4.95$
Preferred chunk	$\mathbf{c}^+$	$\mathbf{c}^-$

Numerical illustration: before and after endpoint substitution. Take $\beta = \gamma^H = 0.9$, $l_v = 10$, and $\varepsilon_H = 0.1$, so that $d = 1$, $r_0 = 0.45$, and $r_1 = 0.55$. With real endpoints, $V^*(\mathbf{o}_0) = 4.5$ and $V^*(\mathbf{o}_1) = 5.5$; after endpoint substitution, the exact fixed point gives $\widehat{V}(\mathbf{o}_0) = 4.95$ and $\widehat{V}(\mathbf{o}_1) = 5.5$. Table 3 lists the resulting chunk values at $\mathbf{o}_0$.

Endpoint substitution reverses the chunk ranking: the model values $\mathbf{c}^-$ as a route to the higher-value state $\mathbf{o}_1$, even though this transition never occurs in the real environment. The induced policy therefore repeatedly selects $\mathbf{c}^-$, remains at $\mathbf{o}_0$, and receives zero reward, whereas the optimal policy repeatedly selects $\mathbf{c}^+$ and receives 0.45 per chunk:

$$V^*(\mathbf{o}_0) = \frac{0.45}{1 - 0.9} = 4.5, \qquad V^{\pi_{\mathrm{D}}}(\mathbf{o}_0) = 0. \tag{15}$$

The resulting policy loss is 4.5, although the rewards are exact and the substituted Bellman equation is solved exactly. Here $Q^*(\mathbf{o}_0, \mathbf{c}^-) = 4.05$ is the value of taking $\mathbf{c}^-$ once and then acting optimally; it differs from the zero return of π_{D}, which takes $\mathbf{c}^-$ at every visit.

A.2 Real-Replay Targets and Model Error

Proof of Proposition 2. Assume bounded chunk returns. Let h_t denote the real history up to the chunk boundary t, and let $\mathcal{B}$ be the space of bounded functions $Q(h, \mathbf{c})$ with the supremum norm. Every critic of the form $Q(\mathbf{o}_t, \mathbf{z}_t, \mathbf{c})$ with $\mathbf{z}_t = e_{\boldsymbol{\phi}}(h_t)$ is an element of $\mathcal{B}$. With exact maximization, the backup whose sample target is Equation 9 is

$$(\mathcal{T}_{\mathrm{W}} Q)(h_t, \mathbf{c}) = \mathbb{E}\Big[R_t^{(H)} + \gamma^H m_{t+H} \max_{\mathbf{c}'} Q(h_{t+H}, \mathbf{c}') \;\Big|\; h_t, \mathbf{c}_t = \mathbf{c} \Big],$$

where the expectation is over the real environment and h_{t+H} extends h_t by the real chunk outcome. The world model enters only through the argument of Q; it does not change the distribution of $(R_t^{(H)}, m_{t+H}, h_{t+H})$.

Contraction. For $Q_1, Q_2 \in \mathcal{B}$, $|\max_{\mathbf{c}'} Q_1(h, \mathbf{c}') - \max_{\mathbf{c}'} Q_2(h, \mathbf{c}')| \le \|Q_1 - Q_2\|_\infty$ and $0 \le m_{t+H} \le 1$, so $\|\mathcal{T}_{\mathrm{W}} Q_1 - \mathcal{T}_{\mathrm{W}} Q_2\|_\infty \le \gamma^H \|Q_1 - Q_2\|_\infty$. By the Banach fixed-point theorem, $\mathcal{T}_{\mathrm{W}}$ has a unique fixed point in $\mathcal{B}$.

Fixed point. Because the boundary observation is Markov, the conditional distribution of $(R_t^{(H)}, m_{t+H}, \mathbf{o}_{t+H})$ given $(h_t, \mathbf{c})$ equals its distribution given $(\mathbf{o}_t, \mathbf{c})$. Regarding $Q^*(\mathbf{o}, \mathbf{c})$ as an element of $\mathcal{B}$ that ignores all of the history except the last observation,

$$(\mathcal{T}_{\mathrm{W}} Q^*)(h_t, \mathbf{c}) = \mathbb{E}\Big[R_t^{(H)} + \gamma^H m_{t+H} \max_{\mathbf{c}'} Q^*(\mathbf{o}_{t+H}, \mathbf{c}') \;\Big|\; \mathbf{o}_t, \mathbf{c} \Big] = Q^*(\mathbf{o}_t, \mathbf{c}),$$

where the last equality is the chunk-level Bellman optimality equation of the real environment. Hence Q^* is the unique fixed point, independently of $\boldsymbol{\phi}$, and its greedy policy is optimal. Since Q^* depends only on $(\mathbf{o}, \mathbf{c})$, it is also expressible as a function of the critic inputs $(\mathbf{o}, \mathbf{z}, \mathbf{c})$. Without the Markov assumption, the same argument shows that the fixed point is the optimal history-dependent value, which is again independent of $\boldsymbol{\phi}$.

Class containment. Write the adapter as $f_{\boldsymbol{\eta}}^{\mathrm{adapt}}(\mathbf{z}) = \mathbf{W}_2\, \sigma(\mathbf{W}_1 \mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2$. At the initialization used by WAVE, $\mathbf{W}_2 = \mathbf{0}$ and $\mathbf{b}_2 = \mathbf{0}$, so $\tilde{\mathbf{u}}_t = \mathbf{u}_t$ in Equation 3 and $Q_{\mathrm{WAVE}}(\mathbf{o}, \mathbf{z}, \mathbf{c}) = Q_{\mathrm{base}}(\mathbf{o}, \mathbf{c})$ for every $\mathbf{z}$, for any value of the base-critic parameters. Thus every base critic is a WAVE critic. □

Contrast with endpoint substitution. In the construction of Appendix A.1, $\mathcal{T}_{\mathrm{W}}$ uses the true endpoint $f_H(\mathbf{o}_0, \mathbf{c}^-) = \mathbf{o}_0$ instead of the predicted $\mathbf{o}_1$, so its fixed point takes the values in the first column of Table 3 and selects $\mathbf{c}^+$, whereas the substituted backup selects $\mathbf{c}^-$. The predictor still enters WAVE through $\mathbf{z}_t$ and the WDC teacher, but neither changes the fixed point.

Scope. Proposition 2 characterizes the exact backup, whose fixed point is determined by the real environment alone. With function approximation and sampled targets, the world model acts through the critic's inputs and the WDC loss, which determine the approximation the critic reaches and how quickly; Section 5 measures this effect. Part (ii) guarantees that augmentation keeps every base critic in the hypothesis class and that online learning starts from the pretrained critic.

B Additional Method Details

This section adds implementation details to Section 4 and summarizes the training procedure in Algorithm 1.

B.1 Causal Posterior and Predictive Prior

At each decision boundary t, the RSSM summarizes the real history $\mathcal{H}_t = (\mathbf{o}_{\leq t}, \mathbf{a}_{<t})$ into the posterior $\mathbf{z}_t = e_{\boldsymbol{\phi}}(\mathcal{H}_t)$; no observation from the current chunk enters it. For a replayed chunk $\mathbf{c}_t$, the prior $\mathbf{z}^p_{t+H} = f_{\boldsymbol{\phi}}(\mathbf{z}_t, \mathbf{c}_t)$ is obtained by one chunk-level transition (Appendix C.2), so the predictive target is conditioned on the same H-step action sequence whose value the critic evaluates. For replayed sequences, posteriors are recomputed from the corresponding real interaction history with the current RSSM whenever a minibatch is sampled for a learner update; posterior latents are not stored as fixed cached values in replay.

B.2 Residual Posterior Conditioning

The adapter in Equation 3 is a bottleneck MLP,

$$f^{\text{adapt}}_{\boldsymbol{\eta}}(\mathbf{z}) = \mathbf{W}_2\, \sigma(\mathbf{W}_1 \mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2,$$

where the 512-dimensional posterior is mapped to a 64-dimensional bottleneck, $\sigma(\cdot) = \mathrm{LayerNorm}(\mathrm{GELU}(\cdot))$, and the output projection maps the bottleneck back to 512 dimensions. The projected residual is added immediately after the critic's first 512-dimensional Dense layer and before that layer's GELU activation and LayerNorm. $\mathbf{W}_2$ and $\mathbf{b}_2$ are initialized to zero, whereas the bottleneck layer uses the standard initialization. This gives Equation 4. The posterior enters as a detached input, so critic optimization updates the base critic and the adapter, while the RSSM parameters are updated only by the world-model objective.

B.3 World-Model Dynamics Consistency Target

A single training-only prediction head $g_{\boldsymbol{\omega}}$ is shared across the critic ensemble and applied to the final hidden representation $\mathbf{h}^Q_t$ of each critic member. The head is a linear map from the 512-dimensional critic representation to the 1024-dimensional prediction $\hat{\mathbf{e}}_t$, with no additional hidden layer or activation. The hidden representation is taken immediately before the scalar value readout. The prediction is matched to the stopped teacher $\mathbf{e}^{\text{WM}}_t = \mathrm{sg}(\mathrm{Norm}([\mathbf{z}_t; \mathbf{z}^p_{t+H}]))$.

Norm standardizes the 256-dimensional deterministic component of each of $\mathbf{z}_t$ and $\mathbf{z}^p_{t+H}$ using fixed statistics computed from the offline training data, while leaving the categorical components unstandardized. For each part $k \in \{\text{cur}, \text{prior}\}$, ℓ^k_{MSE} is the mean squared error on the deterministic 256-dimensional block. For the categorical block, the implementation reshapes its 256 entries into 32 groups of 8 entries and computes the group-wise forward KL divergence $D_{\text{KL}}(p_{\text{teacher}} \| p_{\text{critic}})$, averaged over the groups, where p_{critic} is obtained by applying softmax to the corresponding logits of $\hat{\mathbf{e}}_t$. This grouping is specific to the WDC auxiliary loss and does not change the RSSM's underlying 16×16 categorical latent parameterization. The factor $1/4$ in Equation 7 averages the current deterministic, current categorical, prior deterministic, and prior categorical terms.

Because the current part asks the critic to predict $\mathbf{z}_t$, which it already receives through the adapter, it adds little information beyond the adapter input; this is consistent with the lack of gain from current-only WDC in Table 2.

B.4 Auxiliary-Loss Calibration

Before online interaction, only the WDC head is optimized for 10,000 warm-up updates while all other learner parameters are frozen. For each run, we then compute the gradient-norm ratio between $\mathcal{L}_{\mathrm{WDC}}$ and $\mathcal{L}_Q$ with respect to the shared critic representation over 16 training minibatches. We automatically set λ_{WDC} from the median ratio so that the weighted WDC gradient norm is approximately 5% of the value-gradient norm. The resulting coefficient is fixed for the entire online phase and is not recalibrated online, which keeps WDC a regularizer rather than a competing objective.

B.5 Training Procedure

Algorithm 1 summarizes the offline-to-online procedure. The prediction head $g_{\boldsymbol{\omega}}$ and the prior $\mathbf{z}^p_{t+H}$ are used only inside $\mathcal{L}_{\mathrm{WDC}}$; action selection needs only the actor, the RSSM posterior, and the critic.

Algorithm 1 WAVE offline-to-online training

Require: offline data $\mathcal{D}_{\mathrm{off}}$, chunk horizon H, number of candidates N, online budget T

1: Pretrain QC (actor, critic ensemble) on $\mathcal{D}_{\mathrm{off}}$; pretrain the chunk-level RSSM $\boldsymbol{\phi}$ on $\mathcal{D}_{\mathrm{off}}$ with $\mathcal{L}_{\mathrm{RSSM}}$
2: Attach the adapter $\boldsymbol{\eta}$ with zero output layer and the WDC head $\boldsymbol{\omega}$ ▷ $Q^{(0)}_{\mathrm{WAVE}} = Q_{\mathrm{base}}$
3: Warm up $\boldsymbol{\omega}$ on $\mathcal{D}_{\mathrm{off}}$ for 10,000 updates with all other parameters frozen; calibrate λ_{WDC}
4: $\mathcal{D} \leftarrow \mathcal{D}_{\mathrm{off}}$
5: **while** fewer than T real environment steps have been taken **do**
6: Infer $\mathbf{z}_t$ from the real history; sample $\{\mathbf{c}_t^{(n)}\}_{n=1}^N \sim \pi(\cdot \mid \mathbf{o}_t)$
7: Select $\mathbf{c}_t$ by Equation 5; execute it for up to H steps; add the transitions to $\mathcal{D}$
8: **for** each learner update, at QC's update-to-data ratio **do**
9: Sample a minibatch from $\mathcal{D}$; compute $\mathbf{z}_t$, $\mathbf{z}_{t+H}$, and $\mathbf{z}^p_{t+H}$ with the RSSM (no gradient)
10: Update critic, adapter, and $\boldsymbol{\omega}$ on $\mathcal{L}_{\mathrm{WAVE}} = \mathcal{L}_Q + \lambda_{\mathrm{WDC}}\mathcal{L}_{\mathrm{WDC}}$ with the target in Equation 9
11: Update the actor and the target critics as in QC
12: **end for**
13: Every 5 online main-loop steps, update $\boldsymbol{\phi}$ on mixed offline and online sequences with $\mathcal{L}_{\mathrm{RSSM}}$ using its own optimizer
14: **end while**

C World-Model Architecture and Training

WAVE uses a chunk-level recurrent state-space model (RSSM) to summarize causal interaction history and predict the latent consequence of an H-step action chunk. This section describes the RSSM state representation, chunk-level dynamics, observation prediction, and optimization protocol.

C.1 RSSM State Representation

The recurrent world model maintains an internal latent state

$$\boldsymbol{\xi}_t = (\mathbf{d}_t, \mathbf{s}_t)\,, \tag{16}$$

where $\mathbf{d}_t \in \mathbb{R}^{256}$ is the deterministic recurrent state and $\mathbf{s}_t$ contains 16 categorical stochastic variables with 16 classes each. The representation exposed to WAVE concatenates the deterministic state with the flattened stochastic state:

$$\mathbf{z}_t = [\mathbf{d}_t; \mathrm{vec}(\mathbf{s}_t)] \in \mathbb{R}^{512}. \tag{17}$$

The posterior state is inferred from real observations and causal interaction history, whereas the predictive prior is obtained without observing the future endpoint.

C.2 Chunk-Level Dynamics

Unlike a primitive-action world model that applies one transition per environment step, our RSSM operates directly at the action-chunk level. Given the current posterior state and the complete action

chunk $\mathbf{c}_t$, the model performs one chunk-level transition:

$$\boldsymbol{\xi}^p_{t+H} = F_{\boldsymbol{\phi}}\left(\boldsymbol{\xi}_t, \mathbf{c}_t\right), \tag{18}$$

where

$$\mathbf{c}_t = [\mathbf{a}_t^\top, \ldots, \mathbf{a}_{t+H-1}^\top]^\top. \tag{19}$$

The corresponding predictive representation is

$$\mathbf{z}^p_{t+H} = \mathrm{Rep}\left(\boldsymbol{\xi}^p_{t+H}\right). \tag{20}$$

Thus, one RSSM transition predicts the consequence of the same complete H-step action chunk evaluated by the chunk critic. This provides the horizon alignment between predictive dynamics and action-chunk value learning used by WAVE.

C.3 Observation Prediction

The RSSM includes an observation prediction head applied to the feature of the predicted prior state. For numerical stability, observations are predicted in symlog space:

$$\hat{\mathbf{o}}^{\mathrm{symlog}}_{t+H} = D_{\boldsymbol{\phi}}\left(\mathrm{Rep}\left(\boldsymbol{\xi}^p_{t+H}\right)\right), \tag{21}$$

and converted back to the original observation scale by

$$\hat{\mathbf{o}}_{t+H} = \mathrm{symexp}\left(\hat{\mathbf{o}}^{\mathrm{symlog}}_{t+H}\right). \tag{22}$$

The observation prediction head is trained jointly with the RSSM. WAVE does not use $\hat{\mathbf{o}}_{t+H}$ inside its Bellman target. The decoded prediction is instead used by the Direct-WM baseline described in Appendix E.

C.4 World-Model Training Objective

The world model is optimized with a predictive objective that is separate from the critic objective. Its observation prediction component minimizes reconstruction error in symlog space:

$$\mathcal{L}_{\mathrm{obs}} = \mathbb{E}\left[\left\|\hat{\mathbf{o}}^{\mathrm{symlog}}_{t+H} - \mathrm{symlog}\left(\mathbf{o}_{t+H}\right)\right\|_2^2\right]. \tag{23}$$

The RSSM additionally predicts the discounted H-step chunk return using a 255-bin two-hot categorical target, with loss $\mathcal{L}_{\mathrm{rew}}$. A continuation head predicts whether the trajectory remains non-terminal at the next chunk boundary using binary cross-entropy, yielding $\mathcal{L}_{\mathrm{cont}}$. We also use a progress-prediction head supervised by the number of task-specific completion predicates satisfied at the chunk endpoint. This gives four progress classes for Cube and six for Scene, and is trained with categorical cross-entropy $\mathcal{L}_{\mathrm{prog}}$. The progress head is used only as world-model supervision.

The latent objective consists of a dynamics KL and a representation KL:

$$\mathcal{L}_{\mathrm{latent}} = \mathcal{L}_{\mathrm{dyn}} + 0.1\,\mathcal{L}_{\mathrm{rep}}, \tag{24}$$

where $\mathcal{L}_{\mathrm{dyn}}$ uses $D_{\mathrm{KL}}(\mathrm{sg}(q)\|p)$ and $\mathcal{L}_{\mathrm{rep}}$ uses $D_{\mathrm{KL}}(q\|\,\mathrm{sg}(p))$. Both KL terms use a free-nats threshold of 1.0.

The complete world-model objective is

$$\mathcal{L}_{\mathrm{RSSM}} = \mathcal{L}_{\mathrm{obs}} + \mathcal{L}_{\mathrm{rew}} + \mathcal{L}_{\mathrm{cont}} + 0.1\,\mathcal{L}_{\mathrm{prog}} + \mathcal{L}_{\mathrm{dyn}} + 0.1\,\mathcal{L}_{\mathrm{rep}}. \tag{25}$$

Thus, the observation, reward, continuation, and dynamics-KL terms have unit weight, while the progress and representation-KL terms use weight 0.1.

C.5 Offline Pretraining and Online Adaptation

Before online interaction, the RSSM is pretrained on $\mathcal{D}_{\text{off}}$ independently of the QC learner. During online adaptation, the world model continues to be optimized with its own predictive objective using replay sequences from the offline data and newly collected online experience.

The RSSM optimizer is separate from the critic optimizer. In particular, value-learning gradients do not update the world-model parameters:

$$\nabla_{\phi} \mathcal{L}_{\text{WAVE}}|_{\text{critic update}} = \mathbf{0}, \tag{26}$$

whereas ϕ is updated separately through $\mathcal{L}_{\text{RSSM}}$.

The same world-model checkpoint, predictive objective, and online update protocol are used for WAVE and Direct-WM. Their comparison therefore changes how the learned predictive model enters value learning rather than how the world model itself is trained.

D Experimental Setup and Hyperparameters

This section summarizes the experimental protocol and the hyperparameters used by WAVE and Direct-WM. Unless otherwise stated, WAVE and Direct-WM inherit the underlying QC optimization and candidate-selection procedure, and all comparisons within a task use the same offline initialization, interaction budget, and evaluation protocol.

D.1 OGBench Manipulation Environments

Figure 5 shows representative views of the two OGBench manipulation environments used in our experiments. Both are single-task variants derived from the corresponding OGBench manipulation domains.

For **Cube**, we use the triple-cube setting with the fifth evaluation goal, which corresponds to a stacking task. The robot must perform a sequence of pick-and-place operations to arrange three cube blocks into the target stacked configuration. This setting requires coordinated multi-object manipulation and temporally extended action sequences.

For **Scene**, we use the fifth evaluation goal, *rearrange-hard*. The environment contains a cube block, a drawer, a window, and two lock buttons associated with the drawer and window. The robot must sequentially manipulate these objects to reach the target configuration, requiring multi-stage manipulation and long-horizon reasoning.

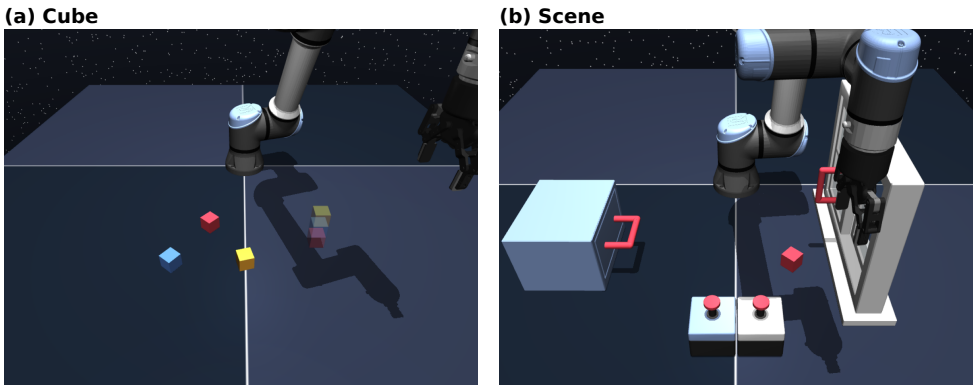


Figure 5: **OGBench manipulation environments used in our experiments.** (a) Cube: a triple-cube stacking task. (b) Scene: a hard multi-object rearrangement task involving the cube, drawer, window, and lock controls. Images are rendered directly from the corresponding OGBench environments.

D.2 Offline-to-Online Protocol

The primary experiments use an offline-to-online protocol consisting of 10^6 offline optimization updates followed by 10^6 real environment interactions. The action-chunk horizon is fixed to $H = 5$, and the discount factor is $\gamma = 0.99$ for QC, WAVE, and Direct-WM. Evaluation interactions are excluded from replay and do not count toward the online interaction budget.

Intermediate evaluations use 50 episodes, while the final evaluation after 10^6 real interactions uses 200 episodes. All reported aggregate results use three random seeds and are reported as mean $\pm$ sample standard deviation.

Table 4 summarizes the shared protocol.

Table 4: **Shared offline-to-online training and evaluation protocol.** These settings are used by QC, WAVE, and Direct-WM unless otherwise specified.

Setting	Value
Action-chunk horizon H	5
Discount factor γ	0.99
Offline optimization updates	10^6
Online interaction budget	10^6 real steps
Critic batch size	256
Intermediate evaluation episodes	50
Final evaluation episodes	200
Number of seeds	3
AUC200 horizon	200k real steps
AUC1M horizon	1M real steps
Evaluation transitions added to replay	No
Imagined transitions added to replay	No

For the controlled early-adaptation studies in Section 5.4, the online interaction budget is reduced to 200k real environment steps. The 10M and 30M dataset-scale experiments use nested offline datasets while keeping the optimization and online-adaptation protocol fixed.

D.3 WAVE-Specific Hyperparameters

Table 5 lists the settings that WAVE adds to the underlying QC learner; all other settings follow QC. The adapter initialization, WDC-head warm-up, and calibration of λ_{WDC} are described in Appendices B.2 and B.4.

Table 5: **WAVE-specific training settings.**

Setting	Value
Candidates per decision (best-of-N) N	32
Adapter input / bottleneck / output width	512 / 64 / 512
Adapter insertion point ($\mathbf{u}_t$)	after first Dense(512), before GELU/LayerNorm
Posterior adapter initialization	zero output projection and bias
WDC head ($\mathbf{h}_t^Q \in \mathbb{R}^{512} \to \hat{\mathbf{e}}_t$)	shared linear Dense(512→1024)
Critic, adapter, and WDC-head optimizer and learning rate	Adam, 3×10^{-4}
WDC-head warm-up	10,000 updates
WDC calibration minibatches	16
Target auxiliary/value gradient ratio	5%
λ_{WDC} after calibration	automatically calibrated per run; fixed online
World-model gradient from critic loss	stopped
Predicted prior used in Bellman target	No
WDC prediction head used at inference	No

D.4 PREDICTIVE-TARGET ABLATIONS

The predictive-target ablations retain the $H = 5$ QC controller, posterior adapter, replay protocol, and real-replay Bellman target. The current-only variant supervises only the current posterior, whereas the future-only variants supervise only a predictive future latent.

For the one-step variant, we train a matched primitive-step RSSM on the same Cube 10M offline data and use its transition $\mathbf{z}^p_{t+1} = f^{(1)}(\mathbf{z}_t, \mathbf{a}_t)$ only to construct the detached WDC future target. The original $H = 5$ RSSM remains the source of the posterior used by the critic, so the control horizon and Bellman backup remain unchanged. The one-step model matches the chunk-level RSSM in latent dimensions, prediction heads, losses, optimizer, and training budget; its action encoder necessarily receives one primitive action rather than the five-action chunk, and the resulting teacher is held fixed during online adaptation.

D.5 WORLD-MODEL CONFIGURATION

The chunk-level RSSM uses a 256-dimensional deterministic recurrent state and 16 categorical stochastic variables with 16 classes each. Flattening the stochastic component yields a 256-dimensional stochastic feature, so the representation exposed to WAVE has total dimension 512.

The RSSM receives the complete H-step action chunk and performs one chunk-level transition to predict the state at $t + H$. The observation prediction head operates in symlog space. For Direct-WM, the decoded observation prediction is mapped back to the original observation scale using symexp before being passed to the actor and critic.

Table 6: **Chunk-level RSSM configuration used by WAVE and Direct-WM.**

Setting	Value
Deterministic RSSM state dimension	256
Number of stochastic variables	16
Categories per stochastic variable	16
Flattened stochastic dimension	256
WAVE representation dimension	512
World-model transition unit	one H-step chunk
Chunk horizon	5
Observation prediction space	symlog
Observation reconstruction for Direct-WM	symexp
Reward prediction representation	255-bin two-hot
Reward-loss coefficient λ_{rew}	1.0
Progress-loss coefficient λ_{prog}	0.1
Optimizer, learning rate	AdamW, 3×10^{-4}
Batch size / sequence length	32 / 16 chunks
Offline pretraining updates	10,000
Online RSSM update cadence	every 5 online main-loop steps
Critic-to-RSSM gradient	stopped

The world model is pretrained independently of QC and continues to adapt during online learning using its own predictive objective. WAVE and Direct-WM use the same RSSM checkpoint, replay construction, and online RSSM update protocol within each task. Thus, the comparison contrasts two ways of integrating the same learned dynamics model into value learning.

D.6 DQC ADAPTATION TO THE OFFLINE-TO-ONLINE PROTOCOL

The publicly released DQC implementation is designed for offline training and does not provide a corresponding online fine-tuning pipeline. To enable a controlled comparison under our common reward-based offline-to-online protocol, we therefore construct an adapted DQC baseline that preserves the core algorithmic structure of the original method while matching the data, reward, and interaction interfaces used by the other baselines.

Specifically, the adapted implementation retains the main DQC components, including the full H-step chunk critic, the partial-action critic, implicit backup and distillation, the flow-matching actor,

best-of-N candidate selection, and Polyak target updates. We also preserve the principal chunk-length settings used by DQC, including the full-chunk and partial-chunk horizons.

The adaptation modifies only the interfaces required to place DQC in the same evaluation setting as QC and WAVE. In particular, the original goal-conditioned reward and relabeling pipeline is replaced by the reward function of the evaluated single-task environment, and goal relabeling, oracle representations, and other privileged inputs are removed. We additionally add online replay, real-environment interaction, checkpoint restoration, and the shared evaluation protocol used throughout our experiments.

We refer to the resulting baseline as *DQC adapted to the common reward-based offline-to-online protocol*. It preserves the core DQC update structure, and its training pipeline differs from the original offline code in the interfaces listed above. Its results in Table 1 therefore characterize this adaptation under our protocol, not the original offline DQC method.

D.7 Other Baseline Configurations

FQL retains the native single-step configuration used by the released ACFQL/FQL implementation, while CGQ uses a source-equivalence-checked port of the released implementation with its official domain-specific method settings. For both baselines, we modify only the interfaces required by the common offline-to-online protocol, including online replay, reward handling, and evaluation. All methods use the same task-specific training data, 10^6 offline optimization updates, 10^6 real online interaction steps, and the common evaluation protocol described in Appendix D.2. No result-driven tuning was performed on the evaluation tasks.

D.8 Compute Resources

All experiments were run on a single NVIDIA GeForce RTX 5090 GPU with 32 GB of GPU memory per run, with no multi-GPU training. Offline pretraining of the underlying RL learner required approximately 1.5–2.6 hours depending on the task. Typical primary runs required approximately 2–5 hours for the 1M-step online stage, depending on the method; WAVE and Direct-WM were toward the upper end because they additionally perform online RSSM inference and adaptation.

D.9 Direct-WM Matching Protocol

Direct-WM is matched to WAVE in chunk horizon, discount factor, QC initialization, world-model initialization, replay protocol, online interaction budget, and RSSM online adaptation. The two methods differ in how world-model information enters value learning.

For WAVE, the predicted prior provides auxiliary representation supervision, while the Bellman target uses the observed replay endpoint. For Direct-WM, the decoded predicted endpoint is instead used for actor candidate generation, current-critic candidate selection, and target-critic bootstrap evaluation, as detailed in Appendix E.

E Direct-WM Baseline

To evaluate a direct endpoint-substitution strategy, we construct a model-based baseline referred to as **Direct-WM**. Direct-WM uses the same underlying QC learner and the same recurrent world model as WAVE, but replaces the real replay endpoint used for TD bootstrapping with the world-model-predicted endpoint. It does not use WAVE's posterior residual adapter, WDC objective, or any other world-model auxiliary objective attached to the critic.

E.1 Predicted Endpoint Construction

Given the causal interaction history at decision boundary t, the RSSM first constructs the current posterior state

$$\boldsymbol{\xi}_t = \mathrm{Posterior}_{\boldsymbol{\phi}}\left(\mathcal{H}_t\right). \tag{27}$$

The complete action chunk

$$\mathbf{c}_t = [\mathbf{a}_t^\top, \ldots, \mathbf{a}_{t+H-1}^\top]^\top \tag{28}$$

is then passed through one chunk-level RSSM transition:

$$\boldsymbol{\xi}^{p}_{t+H} = F_{\boldsymbol{\phi}}\left(\boldsymbol{\xi}_t, \mathbf{c}_t\right). \tag{29}$$

The observation prediction head decodes the resulting prior state in symlog space, after which the prediction is mapped back to the original observation scale:

$$\hat{\mathbf{o}}_{t+H} = \mathrm{symexp}\left(D_{\boldsymbol{\phi}}\left(\mathrm{Rep}\left(\boldsymbol{\xi}^{p}_{t+H}\right)\right)\right). \tag{30}$$

The predicted observation $\hat{\mathbf{o}}_{t+H}$, rather than the replay observation $\mathbf{o}_{t+H}$, is used throughout the bootstrap decision at the next chunk boundary.

E.2 Direct Model-Based Bellman Backup

The policy first proposes candidate next chunks conditioned on the predicted endpoint:

$$\mathcal{C}_{t+H} = \mathcal{C}\left(\hat{\mathbf{o}}_{t+H}\right). \tag{31}$$

The current critic selects the highest-valued candidate,

$$\mathbf{c}'_{t+H} = \arg\max_{\mathbf{c}\in\mathcal{C}_{t+H}} \bar{Q}_{\boldsymbol{\theta}}\left(\hat{\mathbf{o}}_{t+H}, \mathbf{c}\right), \tag{32}$$

and the target critic evaluates the selected chunk at the same predicted endpoint.

The resulting Bellman target is

$$y_t^{\mathrm{DirectWM}} = \sum_{j=0}^{H-1} \gamma^j r_{t+j} + \gamma^H m_{t+H} \bar{Q}_{\bar{\boldsymbol{\theta}}}\left(\hat{\mathbf{o}}_{t+H}, \mathbf{c}'_{t+H}\right). \tag{33}$$

Thus, the real replay reward and bootstrap mask are retained, but candidate generation, candidate selection, and target-critic evaluation are all based on the model-predicted future observation. The real endpoint $\mathbf{o}_{t+H}$ is not used in this bootstrap path.

This construction applies the endpoint substitution analyzed in Lemma 1 to the learned model. Prediction error in $\hat{\mathbf{o}}_{t+H}$ can therefore affect both the next chunk selected by the current critic and the value assigned to that chunk by the target critic. Unlike the lemma, which fixes the candidate set, Direct-WM also queries the actor and critics at the decoded observation, which may lie outside the replay distribution.

E.3 Differences from WAVE-Specific Components

Direct-WM is intentionally kept minimal to provide a direct contrast between endpoint substitution and WAVE's representation-based integration strategy. In particular, Direct-WM does not use

- posterior-conditioned residual critic features;
- world-model dynamics consistency (WDC) supervision; or
- any other WAVE-specific auxiliary critic module.

The actor and critic otherwise follow the underlying QC procedure. The chunk horizon, discount factor, offline initialization, real-interaction budget, candidate-selection procedure, and target-critic update protocol are matched to WAVE.

E.4 World-Model Optimization

The world-model prediction used by Direct-WM is detached from the TD computation. Consequently,

$$\nabla_{\boldsymbol{\phi}} \mathcal{L}_Q^{\mathrm{DirectWM}} = \mathbf{0}, \tag{34}$$

and critic updates do not modify the RSSM.

The RSSM is instead optimized separately using the same predictive objective, offline initialization, replay construction, and online update protocol used by WAVE. Direct-WM and WAVE therefore

share the same learned dynamics model; their difference lies in how its predictions enter value learning.

For diagnostic analysis only, we additionally record the discrepancy between the predicted and observed replay endpoints. The resulting mean replay-batch endpoint MSE is 0.102 on Cube and 0.046 on Scene, measured in the decoded raw-observation space. These real future observations are not used for actor candidate generation, critic candidate selection, or target-critic evaluation in Direct-WM.