MAC-V2: LONG-HORIZON MODEL-BASED VALUE EXPANSION FROM PIXELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We study whether *long-horizon model-based value expansion* can serve as an effective framework in pixel-based offline reinforcement learning (RL). Long-horizon model-based value expansion from pixels is substantially harder than state-based environments, as the agent must learn both the representation and its dynamics, and errors from either can accumulate recursively over imagined trajectories. We introduce MAC-v2, which enables long-horizon model-based expansion from pixels by training a reconstruction-free latent world model with action chunking. By learning representations from prediction and control objectives, MAC-v2 does not need to encode control-irrelevant visual details that can hinder the dynamics prediction. Action chunking further mitigates the prediction error by reducing the number of recursive model predictions required for value expansion. Across diverse visual benchmarks including visual OGBench, MM-Bench2, Robomimic, and LIBERO-10, MAC-v2 achieves strong performance against model-free and model-based offline RL baselines. Our analyses show that long-horizon model-based value expansion is critical for the performance of MAC-v2, and that both action chunking and latent learning objectives are essential for reliable long-horizon latent imagination.

1 INTRODUCTION

Many real-world decision-making problems, such as robotic manipulation, require agents to learn directly from high-dimensional visual observations. This is particularly challenging in offline reinforcement learning (RL), where policies must be learned entirely from a fixed dataset without additional environment interaction. Unlike online RL, offline RL cannot evaluate the policy through environment interaction and rely on off-policy value estimates learned from the fixed dataset. As the learned policy deviates from the behavior policy, these off-policy estimates can become inaccurate due to distribution shift. Model-based value expansion (Feinberg et al., 2018) offers an appealing alternative by learning a world model, where on-policy trajectories can be generated by using a world model without accessing the real environment (Yu et al., 2020; Kidambi et al., 2020; Sun et al., 2023; Hafner et al., 2025b; Park & Lee, 2025).

Prior works have shown that model-based value expansion can be applied to offline RL from pixels, but typically only over short imagination horizons. For example, methods such as LOMPO (Rafailov et al., 2021), OfflineDV2 (Lu et al., 2023), and SeMOPO (Wan et al., 2024) learn latent world models from visual observations and mitigate model exploitation through uncertainty estimation, conservative penalties, or representation decomposition. While these approaches demonstrate that model-based value expansion can be effective, their model rollouts are generally kept short (e.g., LOMPO, OfflineDV2, and SeMOPO use an imagination horizon of 5 steps). In contrast, recent state-based offline RL methods have shown that substantially longer model-based value expansion can be effective for long-horizon tasks (Park et al., 2026; Chung et al., 2026).

Long-horizon model-based value expansion from pixels is substantially harder, as the agent must learn both the representation and its dynamics from the fixed offline data. Since pixel observations contain substantial control-irrelevant information (Fu et al., 2026; Luo et al., 2026), training a world model directly on pixels becomes unnecessarily challenging. This motivates learning a compact latent representation for world modeling. Here, the key challenge is to learn a representation that preserves the information needed for control, and also be predictable by the learned dynamics (Islam

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.

et al., 2022; Sun et al., 2024). Otherwise, this can lead to inaccurate value estimation, as errors from either the representation or the dynamics can accumulate recursively over imagined trajectories, and this issue is further exacerbated in the offline setting, where the agent cannot collect corrective data when those trajectories reach poorly modeled regions.

We introduce MAC-v2, which enables long-horizon model-based value expansion from pixels by training a reconstruction-free latent world model with action chunking. By learning representations from prediction and control objectives, MAC-v2 does not need to encode control-irrelevant visual details that can hinder the dynamics prediction, while also preserving the information needed for control. Moreover, we utilize action chunking (Zhao et al., 2023) for the policy and the world model, reducing the number of recursive model predictions required over long horizons (Park et al., 2026). Together, these design choices mitigate the compounding error during imagined rollouts and enable long-horizon model-based value expansion from visual observations.

We evaluate MAC-v2 across a broad suite of visual benchmarks and show that it matches or substantially outperforms prior model-free and model-based RL methods that rely on shorter-horizon value estimates. MAC-v2 substantially outperforms prior model-free and model-based baselines on 25 visual OGBench (Park et al., 2025a), and matches the performance of the best prior methods in 10 MMBench2 (Hansen & Wang, 2026) tasks. We further evaluate MAC-v2 on robotic manipulation benchmarks, including Robomimic (Mandlekar et al., 2021) and LIBERO-10 (Liu et al., 2023a), where the offline datasets consist primarily of expert demonstrations and behavior cloning policy serves as a strong baseline. On Robomimic, MAC-v2 is the only RL method that matches or exceeds behavior cloning across all evaluated tasks, while remaining competitive with strong behavioral baselines on LIBERO-10. Our analyses show that long-horizon model-based value expansion is critical for this performance, and learning a reconstruction-free encoder with both world model and control objectives is important for reliable long-horizon latent rollouts. Overall, our results suggest that, with the design choices of MAC-v2, latent world models can support effective long-horizon model-based value expansion from pixels.

2 Related Works

Offline model-free RL. Offline RL aims to learn policies solely from pre-collected datasets without further environment interaction (Lange et al., 2012; Levine et al., 2020). A central challenge is distribution shift, as learned policies may select out-of-distribution states or actions for which value estimates are unreliable. Prior work addresses this issue through conservative objectives (Kumar et al., 2020), behavior regularization (Wu et al., 2019; Peng et al., 2019; Nair et al., 2020; Fujimoto & Gu, 2021; Tarasov et al., 2023; Park et al., 2025c), in-sample value maximization (Kostrikov et al., 2022; Xu et al., 2023; Garg et al., 2023), and sample-selection methods (Chen et al., 2023; Hansen-Estruch et al., 2023). More recently, action chunking has shown strong performance on long-horizon offline RL tasks (Li et al., 2025b;a; Song et al., 2026) by improving temporal consistency and reducing the effective decision horizon. However, these model-free approaches rely on short-horizon off-policy targets for value learning, which can limit performance due to inaccurate value propagation (Park et al., 2025b). In contrast, MAC-v2 learns values from long-horizon on-policy targets generated by a model through action-chunked latent rollouts, reducing the error from the recursive bootstrapping.

Offline model-based RL. MAC-v2 is a model-based offline RL method. Such methods learn a dynamics model from the fixed dataset and use it to generate synthetic trajectories for policy evaluation and optimization. By evaluating the policy through the learned model, they can utilize on-policy value estimation instead of off-policy estimates. However, policies may exploit inaccuracies in the learned dynamics model, leading to unreliable value estimates. Prior work addresses this issue through conservative objectives (Kidambi et al., 2020; Yu et al., 2021), uncertainty-aware penalties (Yu et al., 2020; Rigter et al., 2022; Sun et al., 2023; Sims et al., 2024; Park & Lee, 2025), and in-distribution policy optimization (Park et al., 2026). In particular, MAC (Park et al., 2026) combines action-chunked dynamics with a rejection sampling policy, enabling long-horizon model-based value expansion on state-based tasks. MAC-v2 builds on this framework, but addresses the additional challenge of learning a representation that can support accurate recursive model predictions directly from visual observations.

Online RL from pixels. Learning from pixels requires representations that retain information relevant for control while remaining compact and predictable. Prior work has studied this problem extensively in online RL, including model-free approaches (Mnih et al., 2013; Kostrikov et al., 2021; Lee et al., 2025) and model-based approaches (Hafner et al., 2020b; 2025a; Hansen et al., 2022; 2024). Representation objectives include image reconstruction (Hafner et al., 2020b; 2025a;b), contrastive learning (Laskin et al., 2020; Stooke et al., 2021), and reconstruction-free latent prediction (Schrittwieser et al., 2020; Hansen et al., 2024; Maes et al., 2026). MAC-v2 adopts the reconstruction-free latent prediction paradigm established by the prior works, and we do not propose a entirely new representation learning objective. Instead, we study how to utilize such representations for long-horizon model rollouts, required by offline model-based value expansion. We show that, when combined with action chunking, predictive latent representations can support long-horizon imagination and value estimation from a fixed visual dataset.

Offline model-based RL from pixels. Several prior works have specifically studied model-based offline RL from visual observations using reconstruction-based latent representations. LOMPO (Rafailov et al., 2021) learns a latent dynamics model and calculates model uncertainty in latent space, while OfflineDV2 (Lu et al., 2023) adapts DreamerV2 (Hafner et al., 2020a) to offline visual control with uncertainty-aware penalties. However, model imagination is typically kept relatively short (e.g., LOMPO and OfflineDV2 use imagination of 5 steps). We instead study whether predictive latent representations can support *long-horizon model-based value expansion* from pixels. By combining latent prediction with action chunking, MAC-v2 enables stable value expansion over more than 100 environment steps.

3 Preliminaries

Problem setting. We consider a block Markov decision process (MDP) defined as $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathcal{A}, r, \mu, p, q)$, where $\mathcal{S}$ is the latent environment state space, $\mathcal{O}$ is the observation space, and $\mathcal{A} = \mathbb{R}^d$ is the action space. The reward function is $r(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, $\mu(s) \in \Delta(\mathcal{S})$ is the initial state distribution, $p(s' \mid s,a) : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition dynamics, and $q(o \mid s) : \mathcal{S} \to \Delta(\mathcal{O})$ is the observation model. The agent does not observe s_t directly and instead receives $o_t \sim q(\cdot \mid s_t)$. We assume that the current observation is sufficient to identify the underlying state for decision making, so that reactive policies $\pi(a_t \mid o_t)$ are sufficient. For a policy π, we define $V^\pi(s) = \mathbb{E}_{\tau \sim p^\pi(\tau \mid s_0 = s)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$, where $\gamma \in (0,1)$ is the discount factor. The goal of offline RL is to learn a policy maximizing $\mathbb{E}_{s_0 \sim \mu}[V^\pi(s_0)]$ using only a fixed dataset $\mathcal{D} = \{\tau^{(i)}\}$ of previously collected trajectories, without additional environment interaction.

Flow Matching Policy. A flow matching policy (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Liu et al., 2023b) represents a noise-conditioned deterministic policy $a = \pi_\theta(z, \epsilon)$ by transporting samples from a simple reference distribution to the action distribution conditioned on a state. Let $v_\theta : [0,1] \times \mathcal{Z} \times \mathcal{A} \to \mathcal{A}$ denote a state-conditioned, time-dependent velocity field, where $u \in [0,1]$ denotes the flow time. Given an initial noise sample $\epsilon \sim \mathcal{N}(0, I_d)$, the corresponding flow $\psi_\theta(u; z, \epsilon)$ is defined by the ordinary differential equation (ODE) (Lee, 2012)

$$\frac{\mathrm{d}}{\mathrm{d}u} \psi_\theta(u; z, \epsilon) = v_\theta(u, z, \psi_\theta(u; z, \epsilon)), \qquad \psi_\theta(0; z, \epsilon) = \epsilon. \tag{1}$$

The policy is defined as the endpoint of this flow,

$$\pi_\theta(z, \epsilon) = \psi_\theta(1; z, \epsilon), \tag{2}$$

which induces a conditional action distribution $\pi_\theta(a \mid z)$ through $\epsilon \sim \mathcal{N}(0, I_d)$.

Given state-action samples $(s,a) \sim \mathcal{D}$, flow matching trains the velocity field along linear interpolation paths between Gaussian noise ϵ and the target action a. Specifically, for

$$a_u = (1-u)\epsilon + ua, \tag{3}$$

the velocity field is trained by minimizing

$$L^{\text{flow}}(\theta) = \mathbb{E}_{\substack{(s,a) \sim \mathcal{D},\, \epsilon \sim \mathcal{N}(0, I_d), \\ u \sim \text{Unif}([0,1])}} \left[\| v_\theta(u, s, a_u) - (a - \epsilon) \|_2^2 \right]. \tag{4}$$

After training, actions are sampled by drawing $\epsilon \sim \mathcal{N}(0, I_d)$ and numerically integrating the learned velocity field from $u = 0$ to $u = 1$.

4 MAC-v2

We present MAC-v2, an offline model-based RL framework that enables *long-horizon model-based value expansion* from visual observations. MAC-v2 builds on two lines of work: model-based value expansion with action chunking from MAC (Park et al., 2026), and latent world models trained with prediction and control objectives akin to reconstruction-free online model-based RL literatures (Schrittwieser et al., 2020; Hansen et al., 2022; 2024). Neither line of work alone enables long-horizon model-based value expansion in pixel-based offline RL, as MAC-v2 assumes low-dimensional state observations, while prior reconstruction-free latent world models have been limited to short-horizon rollouts.

Our key design choice is to learn a reconstruction-free latent dynamics with action chunking. With action chunking size of n, an imagination horizon of Hn environment steps requires only H recursive model predictions, reducing the compounding of prediction errors. Moreover, we learn the representation with world modeling and behavior cloning objective, encouraging the representation to preserve the information for reward prediction and control, while being predictable under the learned dynamics model. We describe our world model objectives in Section 4.1, and the long-horizon model-based value expansion with the learned latent space in Section 4.2.

4.1 Learning Representations for Long-Horizon Imagination

We encode each image observation into a latent state using a deterministic encoder $f_\xi : \mathcal{O} \to \mathcal{Z}$. Rather than reconstructing the input image, MAC-v2 learns the encoder through objectives relevant to latent prediction and control. In particular, gradients from the dynamics, reward, and behavior cloning objectives are propagated into the encoder, learning a compact representation that retains information for predicting future latent states, rewards, and control.

Concretely, given a latent state and an action chunk, we learn a deterministic dynamics model $p_\psi : \mathcal{Z} \times \mathcal{A}^n \to \mathcal{Z}$ to predict the latent state after n environment steps, by minimizing

$$L^{\text{dyn}}(\psi, \xi) = \mathbb{E}_{(o_t, \boldsymbol{a}_t, \cdots, o_{t+n}) \sim \mathcal{D}} \left[\| p_\psi(f_\xi(o_t), \boldsymbol{a}_t) - \text{sg}(f_\xi(o_{t+n})) \|_2^2 \right], \tag{5}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator.

We also learn a distributional action-chunked reward model $r_\psi : \mathcal{Z} \times \mathcal{A}^n \to \Delta(\mathbb{R})$ with two-hot encoding (Hafner et al., 2025a). We regress the target return $\boldsymbol{r}_t = \sum_{i=0}^{n-1} \gamma^i r_{t+i}$ by minimizing

$$L^{\text{rew}}(\psi, \xi) = \mathbb{E}_{(o_t, \boldsymbol{a}_t) \sim \mathcal{D}} \left[-\log r_\psi(\boldsymbol{r}_t | f_\xi(o_t), \boldsymbol{a}_t) \right]. \tag{6}$$

We parameterize the behavioral policy as a conditional flow model, and train a state-conditioned velocity field $v_\theta : \mathbb{R} \times \mathcal{Z} \times \mathcal{A}^n \to \mathcal{A}^n$ using the flow-matching objective

$$L^{\text{flow}}(\theta, \xi) = \mathbb{E}_{\substack{\epsilon \sim \mathcal{N}(0, I_{nd}),\ (o_t, \boldsymbol{a}_t) \sim \mathcal{D}, \\ u \sim \text{Unif}([0,1]),\ \boldsymbol{a}_\epsilon = (1-u)\epsilon + u\boldsymbol{a}_t}} \left[\| v_\theta(u, f_\xi(o_t), \boldsymbol{a}_\epsilon) - (\boldsymbol{a}_t - \epsilon) \|_2^2 \right]. \tag{7}$$

We denote by $\pi_\theta(z_t, \epsilon)$ the endpoint at $u = 1$ obtained by integrating the learned velocity field from the initial noise $\epsilon \sim \mathcal{N}(0, I_{nd})$.

4.2 Long-Horizon Latent Value Expansion with Action Chunking

With the learned latent world model, we apply the action-chunked value-learning framework of MAC (Park et al., 2026) in the latent space. First, we extract the policy from the behavioral policy by sampling behavioral action candidates $\hat{a}^{(i)} = \pi_\theta(z, \epsilon^{(i)})$ from multiple noises $\epsilon^{(i)} \sim \mathcal{N}(0, I_{nd})$, and extracting the best action among the candidates via rejection sampling (Chen et al., 2023):

$$\pi_\theta^*(z) = \text{argmax}_{\hat{a}^{(1)}, \cdots, \hat{a}^{(M)}} Q_\varphi(z, \hat{a}^{(i)}). \tag{8}$$

We train the value function purely from latent on-policy model rollout, generated as

$$\mathcal{T} = (z_t, \hat{\boldsymbol{a}}_t, \hat{\boldsymbol{r}}_t, \hat{z}_{t+n}, \hat{\boldsymbol{a}}_{t+n}, \hat{\boldsymbol{r}}_{t+n}, \ldots, \hat{z}_{t+Hn}), \tag{9}$$

where $\hat{\boldsymbol{a}}_t$ denotes the action chunk $\hat{a}_{t:t+n-1}$ generated from the rejection sampling policy $\pi_\theta^*(z_t)$, and $\hat{\boldsymbol{r}}_t$ denotes the discounted sum of rewards $\sum_{i=0}^{n-1} \gamma^i r_{t+i}$ predicted from the reward model

Algorithm 1 MAC-v2

Input: Dataset $\mathcal{D}$, rollout length H, action chunking size n, rejection sampling size M, encoder f_ξ, dynamics / reward model p_ψ / r_ψ, value V_φ, Q-function Q_φ, flow BC policy π_θ

// Training loop
while not converged **do**
  ▷ Sample action-chunked batch from the dataset ($\boldsymbol{a}_t = a_{t:t+n-1}$, $\boldsymbol{r}_t = \sum_{i=0}^{n-1} \gamma^i r_{t+i}$)
  Sample batch $\{(o_t, \boldsymbol{a}_t, \boldsymbol{r}_t, o_{t+n})\} \sim \mathcal{D}$
  ▷ Train **dynamics**, **reward** model (and **encoder**) using dataset transitions
  Update p_ψ, f_ξ to minimize $\mathbb{E}[\|p_\psi(f_\xi(o_t), \boldsymbol{a}_t) - \mathrm{sg}(f_\xi(o_{t+n}))\|_2^2]$ (Equation (5))
  Update r_ψ, f_ξ to maximize $\mathbb{E}[\log(\boldsymbol{r}_t \mid r_\psi(f_\xi(o_t), \boldsymbol{a}_t))]$ (Equation (6))
  ▷ Train **BC policy** (and **encoder**) using dataset transitions
  Update π_θ, f_ξ with flow-matching loss (Equation (7))
  ▷ Generate **model rollouts** in latent space ($\hat{z}_t = f_\xi(o_t)$)
  for $k = 0, 1, \ldots, H-1$ **do**
    $\hat{\boldsymbol{a}}_{t+kn} \leftarrow \text{POLICY}(\hat{z}_{t+kn})$
    $\hat{z}_{t+(k+1)n}, \hat{\boldsymbol{r}}_{t+kn} \sim p_\psi(\cdot \mid \hat{z}_{t+kn}, \hat{\boldsymbol{a}}_{t+kn}), r_\psi(\cdot \mid \hat{z}_{t+kn}, \hat{\boldsymbol{a}}_{t+kn})$
  ▷ Update **value** and **critic** using model rollouts
  Update V_φ with nH-step targets from the rollout (Equation (10))
  Update Q_φ with the learned value function V_φ (Equation (11))
// Extract action from flow BC policy π_θ with rejection sampling
function POLICY(z)
  $\epsilon^{(i)} \sim \mathcal{N}(0, I)$
  $\hat{\boldsymbol{a}}^{(i)} = \pi_\theta(z, \epsilon^{(i)})$
  return $\mathrm{argmax}_{\hat{\boldsymbol{a}}^{(1)}, \cdots, \hat{\boldsymbol{a}}^{(M)}} Q_\varphi(z, \hat{\boldsymbol{a}}^{(i)})$

$r_\psi(\cdot|z_t, \boldsymbol{a}_t)$. Here, initial states z_t is uniformly sampled from the dataset and encoded with f_ξ, and subsequent actions, rewards, and next latent states are synthesized by our rejection-sampling policy, reward model, and dynamics model, respectively, where we denote with a hat notation.

After collecting the trajectory, we learn a distributional value function $V_\varphi(z_t) : \mathcal{Z} \to \Delta(\mathbb{R})$ with the following loss:

$$L^V(\varphi) = \mathbb{E}\left[-\log V_\varphi(\hat{v}_{t+kn}|\hat{z}_{t+kn})\right];\ \hat{v}_{t+kn} = \sum_{i=k}^{H-1} \gamma^{(i-k)n} \hat{\boldsymbol{r}}_{t+in} + \gamma^{(H-k)n} v_{\bar{\varphi}}(\hat{z}_{t+Hn}) \quad (10)$$

where $\bar{\varphi}$ denotes exponentially averaged target parameters (Mnih et al., 2013), $v_\varphi(\cdot) = \mathbb{E}_{v \sim V_\varphi(\cdot)}[v]$, and the expectations are over $\mathcal{T} = (z_t = \hat{z}_t, \hat{\boldsymbol{a}}_t, \hat{\boldsymbol{r}}_t, \ldots, \hat{z}_{t+Hn})$ and k uniformly sampled from $\{0, 1, \ldots, H-1\}$.

Finally, we train the action-chunked Q function $Q_\varphi(z_t, \boldsymbol{a}_t) : \mathcal{Z} \times \mathcal{A}^n \to \Delta(\mathbb{R})$ for the rejection sampling with the following loss:

$$L^Q(\varphi) = \mathbb{E}_{o_t \sim \mathcal{D}}\left[-\log Q_\varphi(\hat{\boldsymbol{r}}_t + \gamma^n v_\varphi(\hat{z}_{t+n})|z_t, \hat{\boldsymbol{a}}_t)\right]. \quad (11)$$

For both value learning, we use two-hot encoding for the distribution. We provide a pseudocode for MAC-v2 in Algorithm 1.

5 Experiments

We now empirically evaluate MAC-v2 and examine whether long-horizon model-based value expansion is effective for offline RL from pixels. First, we show that MAC-v2 can serve as an effective general-purpose offline RL algorithm from visual observations in Section 5.1, by evaluating on offline RL benchmarks with diverse control problems and challenging long-horizon tasks where accurate value propagation is important. Second, we examine whether MAC-v2 can be also applied in robotic manipulation settings, where the offline datasets are closer to expert demonstrations and

Table 1: **Results on standard offline RL visual benchmark tasks.** We report the success rate for OGBench tasks and normalized scores (0-100) for MMBench2 tasks across 4 seeds, averaged over 15 episodes. Best scores for each task are highlighted in magenta.

	Model-Free					Model-Based		
Environment	IQL	ReBRAC	IFQL	FQL	QC	LOMPO	OffDV2	MAC-v2
OGBench (25 tasks)								
visual-cube-single (5 tasks)	65 ±4	74 ±1	36 ±5	62 ±4	78 ±2	0 ±0	0 ±0	74 ±1
visual-cube-double (5 tasks)	13 ±1	4 ±3	2 ±1	5 ±3	3 ±1	0 ±0	0 ±0	53 ±3
visual-scene (5 tasks)	18 ±8	33 ±4	21 ±2	41 ±2	57 ±2	2 ±1	4 ±2	72 ±12
visual-puzzle-3x3 (5 tasks)	2 ±3	19 ±1	21 ±1	22 ±1	20 ±0	3 ±2	2 ±1	95 ±4
visual-puzzle-4x4 (5 tasks)	0 ±0	13 ±1	4 ±5	10 ±1	20 ±2	0 ±0	1 ±0	46 ±5
MMBench2 (10 tasks)								
dmcontrol (2 tasks)	19 ±2	39 ±2	17 ±2	53 ±1	22 ±2	14 ±4	44 ±12	31 ±4
maniskill (1 task)	5 ±2	8 ±3	11 ±4	3 ±2	3 ±2	23 ±3	0 ±0	4 ±2
ogbench (2 tasks)	7 ±4	15 ±2	10 ±3	30 ±4	6 ±2	0 ±0	1 ±0	33 ±4
pygame (5 tasks)	41 ±2	50 ±5	47 ±1	58 ±1	50 ±1	31 ±3	43 ±1	66 ±2

strong behavioral cloning policies already provide competitive performance, in Section 5.2. Finally, we analyze the importance of long-horizon model-based value expansion of MAC-v2 and ablate the training objectives used for world model learning in Section 5.3. Unless otherwise specified, all results are averaged over four random seeds.

5.1 Visual offline RL benchmarks

We first evaluate whether MAC-v2 can serve as a general-purpose offline RL method from pixels. We consider two complementary benchmark suites. Visual OGBench (Park et al., 2025a) contains structured manipulation problems with long temporal horizons, suited for evaluating whether model-based value expansion can propagate useful information over long sequences. MMBench2 (Hansen & Wang, 2026) provides a more diverse visual control tasks and diverse source of data collection, testing whether it can be used for beyond the structured manipulation setting.

5.1.1 Experimental setting

Tasks and datasets. We evaluate on 25 visual single-task environments from OGBench (Park et al., 2025a) and report average success rates within each task family, following the evaluation protocol of Park et al. (2025c). We additionally evaluate on 10 "unseen" tasks from MMBench2 (Hansen & Wang, 2026), which cover a diverse set of visual control problems. We resize observations to 64×64 and combine all the datasets (e.g., collected from curiosity-driven policy, expert policy, random policy) assigned to each task. For MMBench2, we report normalized return on a 0-100 scale using the minimum and maximum trajectory returns from the benchmark.

Methods. We compare MAC-v2 against strong model-free and model-based offline RL baselines. For model-free methods, we consider IQL (Kostrikov et al., 2022), ReBRAC (Tarasov et al., 2023), IFQL, the flow-based variant of IDQL (Hansen-Estruch et al., 2023), and FQL (Park et al., 2025c). For model-based methods, we focus specifically on prior *offline* RL methods that learn from visual observations. This literature is considerably smaller than the broader literature on online visual model-based RL methods, which are not directly comparable in our setting. For offline model-based methods, we include LOMPO (Rafailov et al., 2021), which learns a latent world model and apply uncertainty-aware policy optimization on top of the frozen representation, OfflineDV2 (Lu et al., 2023) (denoted as OffDV2), which combines uncertainty penalty (Yu et al., 2020) on top of the world model from DreamerV2 (Hafner et al., 2020a).

5.1.2 Results

Table 1 summarizes the results on the visual OGBench and MMBench2 tasks. On OGBench, MAC-v2 outperforms all baselines in 4 out of 5 task families, and matches QC on the remaining one (`visual-cube-single`). Notably, MAC-v2 shows 95% score in `puzzle-3x3`, where the best prior method scores 22%, while prior model-based RL methods score near-zero performance across all tasks. On MMBench2, MAC-v2 achieves the best performance on the `ogbench` and `pygame` domains, and remains competitive with strong model-free baselines such as FQL on the remaining

Table 2: **Results on visual robotic benchmarks.** We report the success rate across 4 seeds, averaged over 15 episodes. Best scores for each task are highlighted in magenta.

	Model-Free			Model-Based		
Environment	BC	QC	QC-IFQL	LOMPO	OffDV2	MAC-v2
Robomimic (3 tasks)						
lift	100 ± 0	100 ± 0	100 ± 0	27 ± 42	51 ± 19	100 ± 0
can	97 ± 3	97 ± 3	92 ± 3	0 ± 0	0 ± 0	95 ± 3
square	68 ± 21	55 ± 13	55 ± 10	0 ± 0	0 ± 0	72 ± 10
LIBERO (10 tasks)						
LIBERO-10 (10 tasks)	80 ± 1	77 ± 1	78 ± 3	1 ± 2	0 ± 0	80 ± 1

tasks. Together, these results shows the effectiveness of long-horizon model-based value expansion across diverse visual offline RL problems.

5.2 Robotic manipulation benchmarks

We next study the case where offline datasets are composed primarily of successful or near-expert demonstrations. In these datasets, behavioral cloning is already a strong baseline, which is challenging for offline RL algorithms to improve upon. This evaluation tests whether model-based value expansion can also be effective in these datasets, and whether it outperforms behavioral cloning policies.

5.2.1 Experimental setting

Tasks and datasets. We evaluate on three visual manipulation tasks from Robomimic (Mandlekar et al., 2021) and the ten long-horizon tasks from LIBERO-10 (Liu et al., 2023a). For Robomimic, we use the multi-human (`mh`) datasets and 84×84 observations from both the agent-view and wrist cameras. For LIBERO-10, we use the original 128×128 observations and only use the agent-view camera. We report average task success rates on both benchmarks.

Methods. Because temporally coherent action sequences are particularly important in these robotic manipulation tasks, we compare against strong model-free baselines that utilizes action chunking. These include action-chunked flow-matching BC (Lipman et al., 2023), QC (Li et al., 2025b), which learns an action-chunked value function and performs rejection sampling over a flow-based BC policy, and QC-IFQL, an action-chunked variant of IFQL (Hansen-Estruch et al., 2023). We also compare with model-based baselines, including LOMPO (Rafailov et al., 2021) and OfflineDV2 (Lu et al., 2023) (denoted as OffDV2).

5.2.2 Results

Table 2 summarizes the results on visual robotic manipulation. On Robomimic, MAC-v2 matches or exceeds BC on all three tasks and improves performance on the challenging `square` task. Notably, MAC-v2 is the only RL method that consistently matches or outperforms BC across the full benchmark. On LIBERO-10, MAC-v2 remains competitive with BC and strong action-chunked model-free methods. We found that LIBERO-10 consists largely of unimodal demonstration data, leaving relatively limited room for rejection sampling to improve the policy over the behavioral policy. These results suggest that MAC-v2 can retain the robustness of behavior cloning in robotic settings, while providing meaningful improvements when value-based policy selection has sufficient room to improve upon the dataset policy.

5.3 Analysis

In this section, we discuss and analyze the components of MAC-v2 through the following Q&As. Specifically, we examine whether long-horizon model-based value expansion drives its performance gains, and which components make such expansion effective.

Q: Is long-horizon model-based value expansion important for MAC-v2?

A: To answer this question, we compare the performance of MAC-v2 with various horizons, by varying the rollout horizon H and action chunk length n, summarized in Figure 2. First, we evaluate rollout horizons of $\{10, 50, 100\}$ environment steps with an action chunk size of 10 on `puzzle-4x4-task5`. Performance improves as the horizon increases, demonstrating the benefit of long-horizon value expansion. We also vary the action chunk size over $\{1, 5, 10\}$ on `puzzle-4x4-task1`, and the performance improves as the chunk length increases, consistent with the result with the rollout length. Together, these results support that long-horizon model-based value expansion with action chunking is important for the performance of MAC-v2.

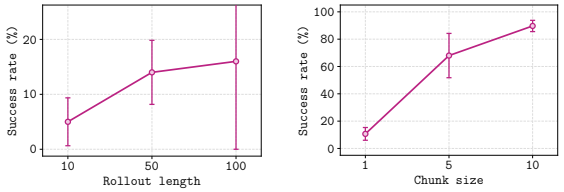

Figure 1: **Importance of long-horizon rollout in MAC-v2.**

Q: Can prior methods also perform long-horizon value estimation?

A: To answer this question, we evaluate QC (Li et al., 2025b) and OfflineDV2 (Lu et al., 2023) on `puzzle-4x4-task5` with different value estimation horizons. Specifically, we vary the action chunk size in QC and the model rollout length in OfflineDV2 to control the horizon of its value estimation. While MAC-v2 benefits from extending the value estimation horizon, neither baseline shows comparable improvements as its horizon increases. These results suggest that simply extending the value estimation horizon is insufficient for the baselines, highlighting the importance of MAC-v2's design choices in facilitating long-horizon model-based value expansion.

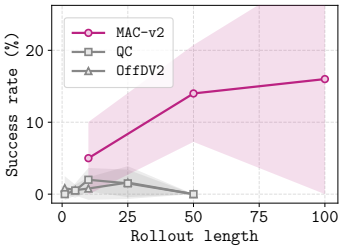

Figure 2: **Prior works do not scale with longer horizons.**

Q: Is each MAC-v2's objective essential for long-horizon model rollouts?

A: We find that the both BC policy objective and WM objective plays an important role in retaining control-relevant and predictable information. Specifically, we remove the gradient from the BC objective and the world modeling objective to the encoder while keeping all other components unchanged on `cube-double-task3`. Figure 3 shows that removing the actor gradient or the world modeling gradient substantially degrades performance, suggesting that both dynamics prediction and policy objective alone are insufficient to learn a representation suitable for long-horizon latent rollouts. This highlights that MAC-v2's design choice to incorporate policy objective and world modeling objective for representation learning is crucial for reliable long-horizon latent value expansion.

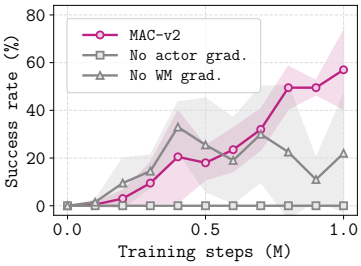

Figure 3: **Objectives are essential.**

Q: How about using other representation objectives instead?

A: Here, we replace the encoder learning objective of MAC-v2 with other representation objectives and examine whether the learned representation can support long-horizon value expansion. We consider pixel reconstruction (Wahlström et al., 2015), SimCLR (Chen et al., 2020), LeWorldModel (Maes et al., 2026) and compare those methods in `cube-double-task3`. As shown in Figure 4, prior representation learning methods substantially degrades the performance. This highlights that MAC-v2's objective is crucial for reliable long-horizon latent value expansion, and cannot be replaced with conventional representation learning objectives.

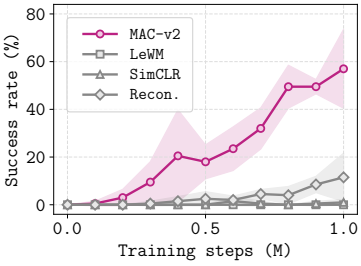

Figure 4: **Other losses falls short.**

6 CLOSING REMARKS

In this work, we introduced MAC-v2, an offline model-based RL framework for long-horizon model-based value expansion from visual observations. MAC-v2 combines reconstruction-free latent dynamics with action chunking, enabling long-horizon imagination with substantially fewer recursive world model predictions. Across 35 standard visual offline RL tasks and 13 robotic manipulation tasks, MAC-v2 achieves strong performance against both model-free and model-based baselines, with substantial gains on challenging OGBench and Robomimic tasks while remaining competitive with strong behavior-cloning baselines on LIBERO-10.

Several open questions remain for MAC-v2, suggesting directions for future work. First, while having larger action chunks reduces the number of recursive model predictions, but excessively long chunks reduce the policy's reactivity. Developing adaptive or hierarchical action chunking could provide a better trade-off between long-horizon prediction and closed-loop control. Moreover, our experiments are primarily limited to simulated, single-task environments with relatively limited visual variation, leaving a substantial gap to realistic multi-task settings. Extending MAC-v2 to large-scale vision-language-action models (VLAs) or world action models (WAMs) could be a promising direction toward long-horizon model-based value expansion in more diverse and realistic domains.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the writing, and implementing the methods. We have not used generative AI tools for designing or providing feedback on research methodology or experiments, proposing or refining hypotheses, assisting with translation. Supporting qualitative and thematic data analysis and interpreting the results, and generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, cleaning and reformatting dataset, interpreting results, are not applicable to this work. The authors have manually reviewed the paper to prevent plagiarism, and read the LLM-generated code and verified and tested its correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We include implementation details needed for reproduction, such as hyperparameters (including the baselines), datasets, and evaluation protocols in Section A. We provide the full results of each tasks in Section B for better future reproducibility.

REFERENCES

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations (ICLR)*, 2023.

Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020.

Hojun Chung, Junseo Lee, and Songhwai Oh. Offline reinforcement learning with universal horizon models. *arXiv preprint arXiv:2605.15603*, 2026.

Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning (ICML)*, 2018.

Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I Jordan, Joseph E Gonzalez, and Sergey Levine. Model-based value expansion for efficient model-free reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2018.

Minghao Fu, Fan Feng, Nicklas Hansen, and Biwei Huang. Back to parsimonious latents: Learning task-centric world models from visual foundations. *arXiv preprint arXiv:2605.25620*, 2026.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2021.

Divyansh Garg, Joey Hejna, Matthieu Geist, and Stefano Ermon. Extreme q-learning: Maxent rl without entropy. In *International Conference on Learning Representations (ICLR)*, 2023.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020a.

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020b.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025a.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025b.

Nicklas Hansen and Xiaolong Wang. Hallucination in world models is predictable and preventable, 2026.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning (ICML)*, 2022.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024.

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *ArXiv*, abs/2304.10573, 2023.

Riashat Islam, Manan Tomar, Alex Lamb, Yonathan Efroni, Hongyu Zang, Aniket Didolkar, Dipendra Misra, Xin Li, Harm Van Seijen, Remi Tachet des Combes, et al. Agent-controller representations: Principled offline rl with rich exogenous information. *arXiv preprint arXiv:2211.00164*, 2022.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel : Model-based offline reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations (ICLR)*, 2021.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Aviral Kumar, Aurick Zhou, G. Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning: State-of-the-art*, pp. 45–73. Springer, 2012.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*, pp. 5639–5650. PMLR, 2020.

Hojoon Lee, Dongyoon Hwang, Donghu Kim, Hyunseung Kim, Jun Jet Tai, Kaushik Subramanian, Peter R Wurman, Jaegul Choo, Peter Stone, and Takuma Seno. Simba: Simplicity bias for scaling up parameters in deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025.

John M Lee. *Introduction to Smooth Manifolds*. Springer, 2012.

Sergey Levine, Aviral Kumar, G. Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *ArXiv*, abs/2005.01643, 2020.

Qiyang Li, Seohong Park, and Sergey Levine. Decoupled q-chunking. *arXiv preprint arXiv:2512.10926*, 2025a.

Qiyang Li, Zhiyuan Zhou, and Sergey Levine. Reinforcement learning with action chunking. *arXiv preprint arXiv:2507.07969*, 2025b.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023b.

Cong Lu, Philip J Ball, Tim GJ Rudner, Jack Parker-Holder, Michael A Osborne, and Yee Whye Teh. Challenges and opportunities in offline reinforcement learning from visual observations. *Transactions on Machine Learning Research (TMLR)*, 2023.

Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-h0. 7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Ajay Mandlekar, Danfei Xu, J. Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2021.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *ArXiv*, abs/1312.5602, 2013.

Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. Accelerating online reinforcement learning with offline datasets. *ArXiv*, abs/2006.09359, 2020.

Kwanyoung Park and Youngwoon Lee. Model-based offline reinforcement learning with lower expectile q-learning. *International Conference on Learning Representations (ICLR)*, 2025.

Kwanyoung Park, Seohong Park, Youngwoon Lee, and Sergey Levine. Scalable offline model-based rl with action chunks. *International Conference on Learning Representations (ICLR)*, 2026.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations (ICLR)*, 2025a.

Seohong Park, Kevin Frans, Deepinder Mann, Benjamin Eysenbach, Aviral Kumar, and Sergey Levine. Horizon reduction makes rl scalable. *arXiv*, abs/2506.04168, 2025b.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In *International Conference on Machine Learning (ICML)*, 2025c.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *ArXiv*, abs/1910.00177, 2019.

Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Offline reinforcement learning from images with latent space models. In *Learning for dynamics and control*, pp. 1154–1168. PMLR, 2021.

Marc Rigter, Bruno Lacerda, and Nick Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *Neural Information Processing Systems (NeurIPS)*, 2022.

Julian Schrittwieser, Ioannis Antonoglou, T. Hubert, K. Simonyan, L. Sifre, Simon Schmitt, A. Guez, Edward Lockhart, Demis Hassabis, T. Graepel, T. Lillicrap, and D. Silver. Mastering Atari, Go, Chess and Shogi by planning with a learned model. *Nature*, 588 7839:604–609, 2020.

Anya Sims, Cong Lu, Jakob N Foerster, and Yee W Teh. The edge-of-reach problem in offline model-based reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Gwanwoo Song, Kwanyoung Park, and Youngwoon Lee. Chunk-guided q-learning. *arXiv preprint arXiv:2603.13971*, 2026.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International conference on machine learning*, pp. 9870–9879. PMLR, 2021.

Ruixiang Sun, Hongyu Zang, Xin Li, and Riashat Islam. Learning latent dynamic robust representations for world models. *arXiv preprint arXiv:2405.06263*, 2024.

Yihao Sun, Jiaji Zhang, Chengxing Jia, Haoxin Lin, Junyin Ye, and Yang Yu. Model-bellman inconsistency for model-based offline reinforcement learning. In *International Conference on Machine Learning*, pp. 33177–33194. PMLR, 2023.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2023.

Niklas Wahlström, Thomas B Schön, and Marc Peter Deisenroth. From pixels to torques: Policy learning with deep dynamical models. *arXiv preprint arXiv:1502.02251*, 2015.

Shenghua Wan, Ziyuan Chen, Le Gan, Shuai Feng, and De-Chuan Zhan. Semopo: learning high-quality model and policy from low-quality offline visual datasets. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 49867–49889, 2024.

Yifan Wu, G. Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *ArXiv*, abs/1911.11361, 2019.

Haoran Xu, Li Jiang, Jianxiong Li, Zhuoran Yang, Zhaoran Wang, Victor Chan, and Xianyuan Zhan. Offline rl with no ood actions: In-sample learning via implicit value regularization. In *International Conference on Learning Representations (ICLR)*, 2023.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. In *Neural Information Processing Systems (NeurIPS)*, 2021.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.

A Experimental Details

We implement MAC-v2 on top of the codebase of Park et al. (2025b). Each experiment takes approximately 9 hours for OGBench and MMBench2 benchmarks, 12 hours for Robomimic and 14 hours for LIBERO-10 benchmark on a single A5000 GPU.

A.1 Implementation Details

Network architecture for MAC-v2. For OGBench experiments, we use the `impala_small` encoder from Espeholt et al. (2018), following the implementation of FQL (Park et al., 2025c) to match the baselines. For LIBERO-10 experiments, we use the Resnet-18 backbone (trained from scratch) as the image is more complicated and higher resolution (128px). For other experiments, we use the convolutional encoder architecture (12M version) from DreamerV3 (Hafner et al., 2025a). On top of the encoder, we use 4-layer MLP with hidden size 256 for actor, critic, and dynamics head. For critic and reward head, we use the two-hot encoding (Hafner et al., 2025a).

Data augmentation. We apply random crop augmentation of size 4 for OGBench and 12 for LIBERO-10, following the implementation of Park et al. (2025c). For model-free approaches, we use the same augmentation scheme as they follow the implementation of Park et al. (2025c). For model-based RL baselines, we follow the setup of original papers.

Accelerating rejection sampling during training. Following Park et al. (2026), we use different numbers of samples for rejection sampling during training and evaluation (which we call N_{train} and N_{test}) to reduce the training time. Specifically, we use $N_{\text{train}} = 8$ during training and $N_{\text{test}} = 32$ at test time.

Implementation details for the compared methods. For the model-free baselines, we follow the implementation by Park et al. (2025a) and Li et al. (2025b). For LOMPO (Rafailov et al., 2021) and OfflineDV2 (Lu et al., 2023), we use the official codebase released by the authors.

MAC-v2 with terminals. We predict the terminal signal $\hat{\mathbf{d}}_t = 1 - \prod_{i=0}^{n-1}(1 - \hat{d}_{t+i})$ from the terminal model $d_\psi(z_t, \boldsymbol{a}_t) : \mathcal{Z} \times \mathcal{A}^n \to \mathbb{R}$, and calculate the discounted reward target $\boldsymbol{r}_t = \sum_{i=0}^{n-1} \gamma^i (\prod_{j=1}^{i-1}(1 - d_{t+j}))r_t$. We use a hard threshold 0.5 and calculate $\hat{d}_t = \mathbb{1}(d_\psi(z_t, a_t) > 0.5)$. We train the value function from latent on-policy model rollout (i.e., imaginary trajectories), generated as

$$\mathcal{T} = \{(z_t^{(i)}, \hat{\boldsymbol{a}}_t^{(i)}, \hat{\boldsymbol{r}}_t^{(i)}, \hat{\boldsymbol{d}}_t^{(i)}, \hat{z}_{t+n}^{(i)}, \hat{\boldsymbol{a}}_{t+n}^{(i)}, \hat{\boldsymbol{r}}_{t+n}^{(i)}, \hat{\boldsymbol{d}}_{t+n}^{(i)}, \ldots, \hat{z}_{t+Hn}^{(i)})\}_{i=1}^{M}. \tag{12}$$

We calculate the value target as:

$$\hat{v}_{t+kn} = \sum_{i=k}^{H-1} \gamma^{(i-k)n} (\prod_{j=k}^{i-1} \hat{d}_{t+jn}) \hat{\boldsymbol{r}}_{t+in} + \gamma^{(H-k)n} (\prod_{j=k}^{H-1} \hat{d}_{t+jn}) v_{\bar{\varphi}}(\hat{s}_{t+Hn}), \tag{13}$$

Finally, we train the action-chunked Q function $Q_\varphi(s_t, \boldsymbol{a}_t) : \mathcal{Z} \times \mathcal{A}^n \to \Delta(\mathbb{R})$ for the rejection sampling with the following loss:

$$L^Q(\varphi) = \mathbb{E}_{o_t \sim \mathcal{D}} \left[-\log Q_\varphi(\hat{\boldsymbol{r}}_t + \gamma^n (1 - \hat{\boldsymbol{d}}_{t+jn}) v_\varphi(\hat{z}_{t+n}) \mid z_t, \hat{\boldsymbol{a}}_t) \right]. \tag{14}$$

A.2 HYPERPARAMETERS

We group the configurations into three families: model-free baselines (Table 4), model-based baselines (Table 5), and MAC-v2 (Table 6). Image resolution, encoder and augmentation are set per benchmark, and are given in Table 3. Evaluation uses 15 episodes for each seed. We match the encoder for MAC-v2 and model-free methods, but we use the original codebase's architecture for model-based baselines as their architecture (history-conditioned RSSM) significantly differs from our method (markovian CNN encoder).

Table 3: **Per-benchmark input and encoder settings.** These apply to MAC-v2 and the model-free baselines, so every comparison uses the same encoder within a benchmark. Please refer to Section A.1 for implementation details of the encoder.

Benchmark	Resolution	Encoder	Aug. pad
OGBench	64×64	`impala_small`	4
MMBench2	64×64	`12M Conv.`	4
Robomimic	84×84	`12M Conv.`	4
LIBERO-10	128×128	`ResNet-18`	12

Shared hyperparameters for model-free baseline. IQL, ReBRAC, IFQL, FQL and QC are run from the reference FQL codebase (Park et al., 2025c) at its visual-task defaults. QC additionally uses best-of-32 sampling over 5-step action chunks, so it is chunked where the other four are single-step. We use $p_{\text{aug}} = 0.5$ following the original codebase.

Table 4: **Hyperparameters of the model-free baselines.**

Hyperparameters	Value
Gradient steps	500K
Learning rate	3×10^{-4}
Optimizer	Adam (Kingma & Ba, 2015)
Batch size	256
MLP size	$[512, 512, 512, 512]$
Frame stack	3
Image augmentation	DrQ random shift, $p_{\text{aug}} = 0.5$ (tied)
Discount factor γ	0.99

Shared hyperparameters for model-based baseline. Both are run at their authors' defaults, changing only the uncertainty coefficient λ. Both methods train their world model for 25K steps before the policy phase. Both methods are trained for 100k steps, as we found the performance starts to degrade after 100k steps. We note that the computational budget for 100k steps for these methods is larger than MAC-v2, taking around 16 hours in a single A5000 GPU.

Table 5: **Shared hyperparameters of model-based baselines.**

Hyperparameters	Value
Gradient steps	100K
Learning rate	3×10^{-4}
Optimizer	Adam (Kingma & Ba, 2015)
Batch size	256
MLP size	$[512, 512, 512, 512]$
Image resolution	64×64
Frame stack	1

MAC-v2 hyperparameters. We report the hyperparameters for our method in Table 6. MAC-v2 uses the same $(n, H, N_{\text{train}}, N_{\text{test}}) = (10, 10, 8, 32)$ across all tasks and all four benchmarks, no per-task tuning is performed. We apply DrQ-style random-shift augmentation (Kostrikov et al.,

2021), using the same shift for an observation and the next observation. We normalize the value with the maximum return of the dataset, and train the value and reward heads with symexp two-hot categorical loss of Hafner et al. (2025a).

Table 6: **Hyperparameters of MAC-v2.**

Hyperparameters	**Value**
Learning rate	3×10^{-4}
Optimizer	Adam
Layer normalization	True
Target network update rate τ	0.02
Discount factor γ	0.999
λ-return coefficient	1.0
Flow steps	5
N_{train}	8
N_{test}	32
Rollout length H	10
Action chunk size n	10
Latent dimension	512
Image augmentation	DrQ random shift (tied)
Value / reward loss	symexp two-hot, 255 bins on $[-5, 5]$
Weight decay	1×10^{-4}
Gradient clipping	1.0
Frame stack	1

Hyperparameters for baselines. For LOMPO and Offline DV2, we sweep the uncertainty penalty coefficient $\lambda \in \{1, 3, 10, 30\}$. For FQL we search over the behavior-cloning coefficient $\alpha \in \{0.1, 0.3, 1.0\}$, and for ReBRAC over $(\alpha_{\text{actor}}, \alpha_{\text{critic}}) \in \{0.01, 0.1, 1.0\} \times \{0, 0.01\}$. For FQL, IQL, ReBRAC, we use the reported hyperparameter from Park et al. (2025c) in OGBench. For QC and QC-IFQL, we search over the action chunking size $\{5, 10\}$. IFQL is run at a single fixed configuration. The hyperparameters are reported in Table 7, 8, 9, 10 for OGBench, MMBench2, Robomimic, LIBERO-10, repectively. We denote "N/A" in the tables if a method achieves zero performance across all hyperparameters tested in our sweep. If not specified, all other hyperparameters follow the defaults provided in the original papers.

Table 7: **Baseline hyperparameters on the OGBench visual tasks.** Uncertainty penalty λ for the model-based baselines, behavior-cloning coefficient α for FQL, and $(\alpha_{\text{actor}}, \alpha_{\text{critic}})$ for ReBRAC.

`Environment`	`LOMPO` (λ)	`OffDV2` (λ)	`FQL` (α)	`ReBRAC` ($\alpha_{\text{actor}}, \alpha_{\text{critic}}$)	`IQL` (α)
`visual-cube-single-play-v0`	5	5	300	$(1.0, 0)$	1
`visual-cube-double-play-v0`	5	10	100	$(0.1, 0)$	0.3
`visual-scene-play-v0`	3	10	100	$(0.1, 0.01)$	10
`visual-puzzle-3x3-play-v0`	3	10	300	$(0.3, 0.01)$	10
`visual-puzzle-4x4-play-v0`	5	3	300	$(0.3, 0.01)$	3

Table 8: **Baseline hyperparameters on MMBench2.** Uncertainty penalty λ for the model-based baselines, behavior-cloning coefficient α for FQL, and $(\alpha_{\text{actor}}, \alpha_{\text{critic}})$ for ReBRAC, α for IQL.

Environment	LOMPO (λ)	OffDV2 (λ)	FQL (α)	ReBRAC $(\alpha_{\text{actor}}, \alpha_{\text{critic}})$	IQL (α)
cup-catch-var1	1	10	0.3	$(0.01, 0)$	1
finger-turn-easy-var1	30	1	0.3	$(0.01, 0.01)$	1
ms-push-banana	30	30	1.0	$(1.0, 0)$	1
og-point-var1	N/A	10	0.3	$(0.01, 0.01)$	1
og-point-var2	1	10	0.3	$(0.01, 0.01)$	1
pygame-point-maze-var4	3	1	0.1	$(0.01, 0)$	1
pygame-reacher-easy	30	1	1.0	$(0.1, 0.01)$	1
pygame-dungeon-explorer1	10	3	0.3	$(0.1, 0)$	1
pygame-foraging	30	10	0.1	$(1.0, 0)$	1
pygame-whirlpool	3	10	0.1	$(0.1, 0)$	1

Table 9: **Baseline hyperparameter on Robomimic.** Uncertainty penalty λ for the model-based baselines, action chunk size n for BC, QC and QC-IFQL.

Task	LOMPO (λ)	OffDV2 (λ)	BC (n)	QC (n)	QC-IFQL (n)
lift	3	10	10	5	5
can	N/A	N/A	10	5	10
square	N/A	N/A	10	5	5

Table 10: **Per-task uncertainty coefficient λ of the model-based baselines on LIBERO-10.** We sweep over $\lambda \in \{1, 3, 10, 30\}$ on each tasks for model-based baselines. N/A marks tasks on which every λ scored zero.

Task	LOMPO (λ)	OffDV2 (λ)
put the white mug on the left plate and put the yellow and white mug on the right plate	N/A	N/A
put the white mug on the plate and put the chocolate pudding to the right of the plate	N/A	N/A
put the yellow and white mug in the microwave and close it	3	N/A
turn on the stove and put the moka pot on it	N/A	N/A
put both the alphabet soup and the cream cheese box in the basket	N/A	N/A
put both the alphabet soup and the tomato sauce in the basket	10	N/A
put both moka pots on the stove	N/A	N/A
put both the cream cheese box and the butter in the basket	N/A	N/A
put the black bowl in the bottom drawer of the cabinet and close it	N/A	N/A
pick up the book and place it in the back compartment of the caddy	N/A	N/A

B Complete Numerical Results

For completeness, we provide the full per-task results for OGBench tasks in Table 11 (corresponding to Table 1). The results are averaged over 4 seeds and we report the standard deviations for each tasks. We highlight the numbers with the best performance.

Table 11: **Complete results for OGBench visual benchmark tasks.**

	Model-Free					Model-Based		
Environment	IQL	ReBRAC	IFQL	FQL	QC	LOMPO	OffDV2	MAC-v2
visual-cube-single-task1	73 ±7	84 ±4	44 ±6	67 ±3	88 ±5	0 ±0	0 ±0	87 ±2
visual-cube-single-task2	57 ±6	71 ±6	29 ±9	70 ±4	80 ±5	0 ±0	0 ±0	72 ±3
visual-cube-single-task3	79 ±4	85 ±1	59 ±10	68 ±13	90 ±4	0 ±0	0 ±0	86 ±3
visual-cube-single-task4	62 ±18	69 ±4	27 ±7	51 ±11	68 ±5	0 ±0	0 ±0	64 ±3
visual-cube-single-task5	54 ±4	62 ±4	22 ±4	57 ±8	62 ±6	0 ±0	0 ±0	62 ±3
visual-cube-double-task1	25 ±7	5 ±3	6 ±2	15 ±12	6 ±2	0 ±0	1 ±1	67 ±4
visual-cube-double-task2	4 ±1	1 ±0	0 ±0	0 ±0	2 ±2	0 ±0	0 ±0	47 ±14
visual-cube-double-task3	13 ±4	4 ±3	1 ±1	2 ±2	2 ±1	0 ±0	0 ±0	52 ±6
visual-cube-double-task4	1 ±1	0 ±0	0 ±1	0 ±0	0 ±1	0 ±0	0 ±0	10 ±5
visual-cube-double-task5	21 ±3	10 ±14	2 ±1	4 ±2	4 ±1	0 ±0	0 ±0	90 ±4
visual-scene-task1	69 ±40	97 ±2	88 ±4	99 ±1	100 ±0	7 ±5	9 ±6	100 ±0
visual-scene-task2	0 ±0	65 ±16	0 ±1	88 ±5	96 ±3	5 ±6	9 ±5	100 ±0
visual-scene-task3	17 ±2	4 ±3	15 ±6	19 ±5	89 ±9	0 ±0	0 ±0	90 ±2
visual-scene-task4	1 ±0	0 ±1	0 ±0	1 ±1	1 ±1	0 ±0	0 ±0	42 ±29
visual-scene-task5	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	29 ±29
visual-puzzle-3x3-task1	7 ±12	90 ±4	100 ±1	97 ±1	100 ±0	13 ±9	9 ±5	100 ±0
visual-puzzle-3x3-task2	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	100 ±0
visual-puzzle-3x3-task3	0 ±1	2 ±1	1 ±1	1 ±0	0 ±0	0 ±0	0 ±0	83 ±22
visual-puzzle-3x3-task4	0 ±0	1 ±1	1 ±0	6 ±2	0 ±0	0 ±0	0 ±0	98 ±2
visual-puzzle-3x3-task5	1 ±1	0 ±1	1 ±2	7 ±3	0 ±0	0 ±0	0 ±0	96 ±6
visual-puzzle-4x4-task1	0 ±0	22 ±5	11 ±19	28 ±5	57 ±7	0 ±0	2 ±1	90 ±4
visual-puzzle-4x4-task2	0 ±0	9 ±2	0 ±1	1 ±0	2 ±1	0 ±0	1 ±0	21 ±8
visual-puzzle-4x4-task3	0 ±0	15 ±2	2 ±3	14 ±3	31 ±11	0 ±0	1 ±1	82 ±18
visual-puzzle-4x4-task4	0 ±0	10 ±1	4 ±4	7 ±2	8 ±8	0 ±0	0 ±1	20 ±7
visual-puzzle-4x4-task5	0 ±0	10 ±1	0 ±0	1 ±0	1 ±1	0 ±0	1 ±1	16 ±16

Table 12: **Complete results for MMBench2 tasks.**

	Model-Free					Model-Based		
Environment	IQL	ReBRAC	IFQL	FQL	QC	LOMPO	OffDV2	MAC-v2
cup-catch-var1	13 ±2	47 ±6	11 ±1	71 ±1	22 ±4	4 ±3	45 ±26	33 ±4
finger-turn-easy-var1	24 ±5	32 ±4	23 ±4	35 ±3	23 ±2	24 ±9	43 ±1	29 ±5
ms-push-banana	5 ±2	8 ±3	11 ±4	3 ±2	3 ±2	23 ±3	0 ±0	4 ±2
og-point-var1	8 ±5	17 ±2	11 ±3	32 ±6	7 ±2	0 ±0	1 ±0	42 ±5
og-point-var2	7 ±4	14 ±3	9 ±3	28 ±5	6 ±2	0 ±0	1 ±0	24 ±4
pygame-point-maze-var4	10 ±6	25 ±14	26 ±7	45 ±4	16 ±1	2 ±3	5 ±1	50 ±8
pygame-reacher-easy	11 ±2	11 ±1	10 ±1	12 ±2	9 ±1	5 ±1	28 ±4	22 ±3
pygame-dungeon-explorer1	43 ±2	57 ±4	46 ±3	65 ±5	64 ±1	27 ±8	28 ±7	78 ±3
pygame-foraging	65 ±0	65 ±2	65 ±1	76 ±1	68 ±1	44 ±8	52 ±3	82 ±1
pygame-whirlpool	74 ±7	89 ±8	86 ±4	91 ±1	92 ±3	79 ±18	99 ±1	99 ±1

Table 13: **Complete results for LIBERO tasks.**

	Model-Free			Model-Based		
Environment	BC	QC	QC-IFQL	LOMPO	OffDV2	MAC-v2
LIBERO (10 tasks)						
turn on the stove and put the moka pot on it	96 ± 3	89 ± 10	85 ± 9	0 ± 0	0 ± 0	97 ± 2
put the black bowl in the bottom drawer of the cabinet and close it	98 ± 1	86 ± 9	94 ± 3	0 ± 0	0 ± 0	95 ± 3
put the yellow and white mug in the microwave and close it	84 ± 3	77 ± 7	83 ± 8	9 ± 12	0 ± 0	81 ± 8
put both moka pots on the stove	76 ± 4	69 ± 2	64 ± 10	0 ± 0	0 ± 0	75 ± 16
put both the alphabet soup and the cream cheese box in the basket	69 ± 4	67 ± 18	68 ± 8	0 ± 0	0 ± 0	75 ± 6
put both the alphabet soup and the tomato sauce in the basket	77 ± 5	80 ± 8	75 ± 8	2 ± 4	0 ± 0	82 ± 7
put both the cream cheese box and the butter in the basket	84 ± 10	90 ± 6	85 ± 3	0 ± 0	0 ± 0	86 ± 5
put the white mug on the left plate and put the yellow and white mug on the right plate	83 ± 6	71 ± 11	83 ± 3	0 ± 0	0 ± 0	74 ± 9
put the white mug on the plate and put the chocolate pudding to the right of the plate	49 ± 7	59 ± 5	68 ± 14	0 ± 0	0 ± 0	48 ± 9
pick up the book and place it in the back compartment of the caddy	84 ± 5	80 ± 5	78 ± 7	0 ± 0	0 ± 0	82 ± 5

C TRAINING CURVES

We provide the training curve of MAC-v2 for the OGBench, MMBench, robomimic, LIBERO-10 result of Tables 1 and 2, in Figures 5 to 8. We plot the mean and the standard deviation (across 4 seeds) by covering [mean - std, mean + std] area with a lighter color.

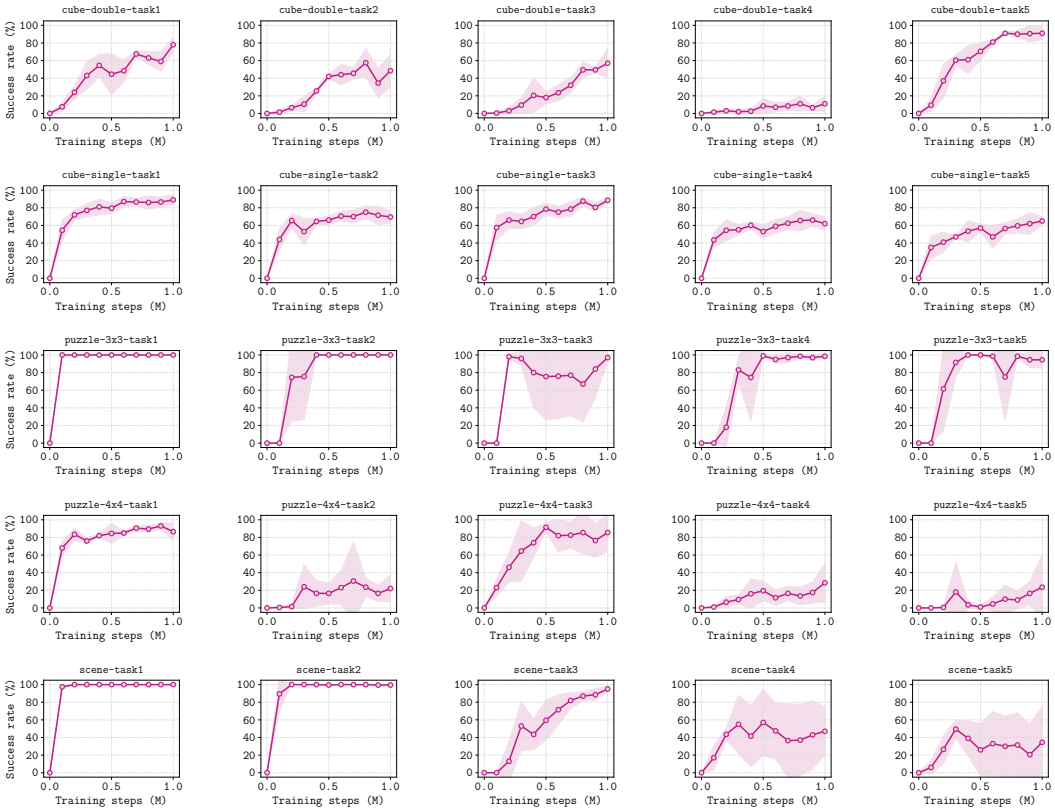

Figure 5: **Training curve of MAC-v2 in OGBench environments.** We report the success rate for 15 evaluation episodes across 4 seeds (total 60 episodes). Shaded region represents the [mean - std, mean + std].

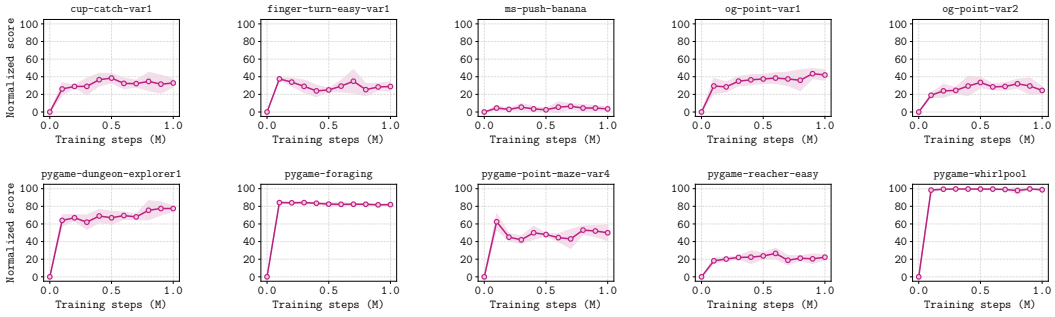

Figure 6: **Training curve of MAC-v2 in MMBench2 environments.** We report the normalized return for 15 evaluation episodes across 4 seeds (total 60 episodes). Shaded region represents the [mean - std, mean + std].

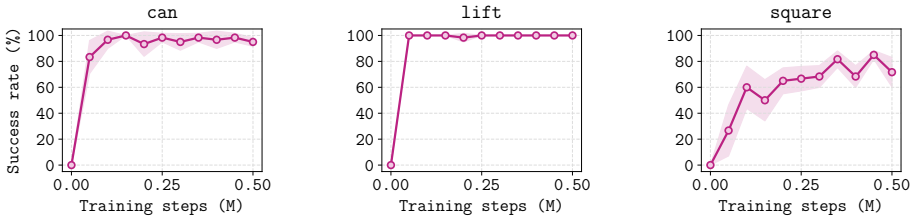

Figure 7: **Training curve of MAC-v2 in Robomimic environments.** We report the success rate for 15 evaluation episodes across 4 seeds (total 60 episodes). Shaded region represents the [mean - std, mean + std].

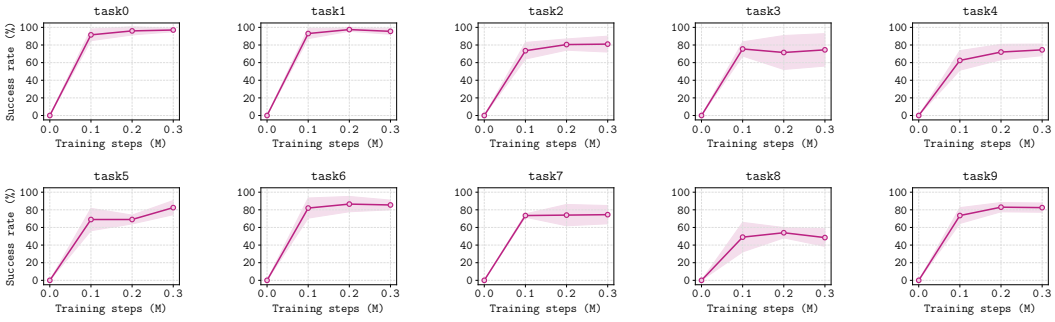

Figure 8: **Training curve of MAC-v2 in LIBERO-10 environments.** We report the success rate for 15 evaluation episodes across 4 seeds (total 60 episodes). Shaded region represents the [mean - std, mean + std]. For the task instruction of each tasks for this graph, please refer to Table 2.