ADAPTIVE Q-CHUNKING FOR OFFLINE-TO-ONLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline-to-online reinforcement learning with action chunking eliminates multi-step off-policy bias and enables temporally coherent exploration, but all existing methods use a fixed chunk size across every state. This is suboptimal: near contact events the agent needs short chunks for reactive control, while during free-space motion long chunks provide better credit assignment. The natural solution is to train critics for several chunk sizes and select the best one at each state, but naive comparison of learned critic values systematically collapses to the shortest chunk due to discount-scale mismatch, and degrades to noise in low-value states. We propose **Adaptive Q-Chunking (AQC)**, which resolves both failures by comparing the advantage of each chunk size relative to a per-horizon baseline, normalized by the discount factor. This criterion converts biased wrong answers into unbiased near-random choices when no genuine signal exists, and becomes discriminative when a particular scale enables better planning. We prove theoretical bounds on the advantage selector's noise immunity and on the value dominance of adaptive chunking over any fixed chunk size. We demonstrate that **AQC** achieves state-of-the-art offline and online success rates on OGBench and Robomimic, and can be applied to enhance the performance of large-scale VLA models that predict action sequences, significantly boosting performance on RoboCasa-GR1 tasks.

1 INTRODUCTION

Reinforcement learning (RL) promises to learn any task from rewards alone, but starting from scratch is impractical for complex behaviors (Levine et al., 2020). Offline RL addresses this by learning a fixed policy from pre-collected data, but must contend with distributional shift and value overestimation (Kumar et al., 2020; Fujimoto et al., 2018; Ross et al., 2011). A practical approach is offline-to-online RL: first, learn from a fixed dataset of demonstrations, then continue improving through online interaction (Nair et al., 2020; Lee et al., 2022; Ball et al., 2023; Nakamoto et al., 2024; Zhou et al., 2025). This combination works well for many tasks, but struggles in long-horizon settings where rewards are sparse (Park et al., 2025a;b). The agent must execute hundreds of actions before receiving feedback, making learning slow. Even worse, single-step policies tend to behave inconsistently during exploration, rarely reaching the states where rewards occur (Li et al., 2025b).

Action chunking addresses both problems by committing to sequences of actions rather than single steps (Zhao et al., 2023; Chi et al., 2023). A growing body of work incorporates chunks into RL: transformer-based episodic critics (Li et al., 2025a), coarse-to-fine discrete Q-networks (Seo & Abbeel, 2025; Seo et al., 2024), detached value learning for offline RL (Kim et al., 2025), and receding-horizon exploitation (Nagy et al., 2026). Among these, Q-chunking (QC) (Li et al., 2025b) is the most directly relevant predecessor: it learns Q-values over chunks of h consecutive actions, making value estimation stable even when the dataset is imperfect (Fedus et al., 2020), and executing entire chunks open-loop produces more coherent exploration. However, QC uses a single fixed chunk size h for every state and every task. Decoupled Q-chunking (DQC) (Li et al., 2026) partially addresses this rigidity by allowing the policy and critic to operate at different chunk sizes, but it does not adapt the chunk sizes per state and requires goal-conditioned training that does not apply to standard reward-based tasks (Andrychowicz et al., 2017).

We argue that the deeper limitation is to use any fixed chunk size at all. Consider a robot arm moving through free space toward an object: a long action sequence works well here because the motion is predictable. But near a contact event like grasping or insertion, the same long sequence

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.

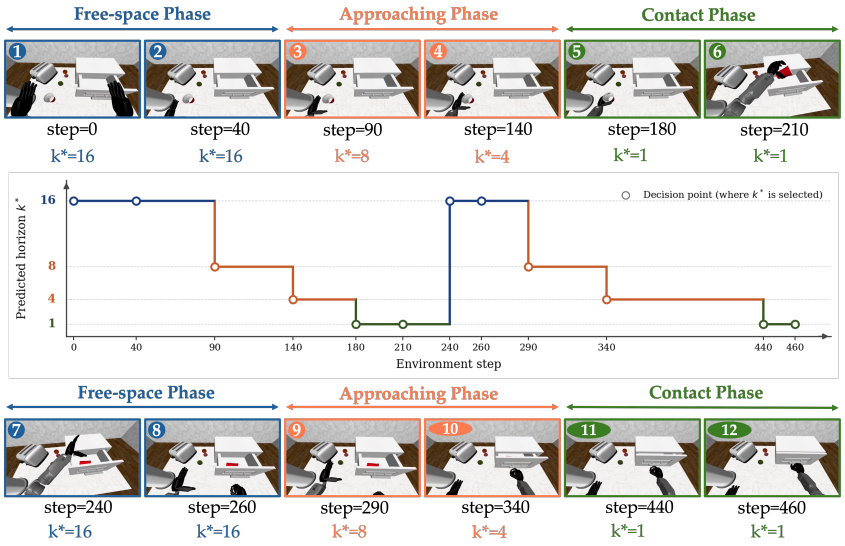


Figure 1: **Qualitative rollout of AQC.** Our method adaptively adjusts the commitment horizon k^* based on the task phase. It utilizes long chunks ($k^* = 16$) for efficient movement in free space and automatically switches to fine-grained control ($k^* = 1$) during complex contact-rich manipulation.

becomes harmful, small errors accumulate through contact dynamics, and the robot must react to new observations at every step. No single chunk size handles both situations well, and tuning the right size for each task requires extensive experimentation.

The natural solution is to train critics for several chunk sizes and pick the best one at each state. But this naive approach fails for two reasons. First, critics for shorter chunks produce larger values by construction, so the agent always picks the shortest chunk, even when a longer one would be better. Second, in states with low expected return, all chunk sizes give nearly identical values, so the selection becomes random noise rather than a meaningful choice.

We propose **Adaptive Q-Chunking (AQC)**, which resolves both problems by comparing the *advantage* of each chunk size relative to what the baseline behavior would achieve. The advantage measures how much better a particular chunk is compared to executing a chunk of that length from the demonstration policy. This comparison removes both sources of bias: short chunks no longer dominate artificially, and in low-value states where no chunk is clearly better, all scores collapse to near-zero and the selection becomes effectively random—avoiding the systematic bias toward shorter chunks that plagues the uncorrected selector. When a short chunk is uniquely helpful near contacts, or a long chunk benefits free-space motion, the advantage becomes large and guides the selection (Figure 1).

The same approach naturally extends to enhancing vision-language-action (VLA) models (Bjorck et al., 2025) with offline RL. We train critics on top of a GR00T N1.6 backbone on RoboCasa-GR1 tabletop manipulation tasks. The VLA natively outputs action chunks; **AQC** scores multiple candidate chunks at each scale and selects the best prefix via the per-scale advantage criterion. This gives the VLA reactive control near contacts while preserving smooth motion in free space, all without modifying the underlying model.

Our main contributions are:

- We present **AQC**, an offline-to-online RL method that adapts chunk sizes per state through a principled advantage-based criterion.
- We provide theoretical analysis establishing advantage separability, value dominance over fixed-chunk policies, and closed-loop optimality bounds.
- We demonstrate state-of-the-art results on OGBench (Park et al., 2025a) and Robomimic (Mandlekar et al., 2021), outperforming prior methods across diverse domains.
- We show that our approach enhances the performance of VLA policies on RoboCasa-GR1 tabletop tasks, exceeding prior chunked-critic methods on contact-rich manipulation.

2 Related Work

Offline-to-online reinforcement learning. Offline-to-online RL methods pretrain a policy on a fixed dataset and subsequently refine it through environment interaction. Prior work addresses the offline-to-online value shift via advantage-weighted regression (Nair et al., 2020), implicit Q-learning (Kostrikov et al., 2021), calibrated pessimism (Nakamoto et al., 2024), interleaved offline-online batches (Ball et al., 2023), warmup-based recalibration (Zhou et al., 2025), policy expansion (Zhang et al., 2023), or reincarnating RL (Agarwal et al., 2022). While these methods focus on managing the offline-to-online transition, our approach instead targets a complementary bottleneck: the inability of existing methods to adjust their planning horizon per state.

Action chunking in reinforcement learning. Action chunking predicts and executes sequences of future actions rather than a single action at each step. QC (Li et al., 2025b) extends this to TD-based RL, training a critic over full h-step chunks to eliminate the off-policy multi-step bias while producing temporally coherent exploration. TOP-ERL (Li et al., 2025a) introduces transformer-based off-policy episodic RL with multi-step returns over short action chunks. Tian et al. (2025) combine chunked critics with soft actor-critic, learning a transformer critic on n-step returns while keeping a single-step actor. CQN-AS (Seo & Abbeel, 2025) and CQN (Seo et al., 2024) use coarse-to-fine Q-networks with action sequences for data-efficient RL, factorizing the Q-function over discrete action bins. DEAS (Kim et al., 2025) detaches value learning from the action sequence for scalable offline RL. SEAR (Nagy et al., 2026) exploits the temporal structure of action chunks with a receding horizon to improve sample efficiency. DQC (Li et al., 2026) decouples the chunk size used for the critic from that used for the policy, but requires goal-conditioned hindsight relabeling (Andrychowicz et al., 2017) and an additional distillation hyperparameter. We instead learn partial critics via direct k-step TD losses bootstrapped from a long-horizon value estimate, achieving the same fixed point without distillation or goal conditioning, and extends to adaptive chunk size selection at inference.

Multi-step returns and hierarchical RL. Multi-step TD methods (Hessel et al., 2018) accelerate credit propagation but introduce off-policy bias when data is stale (Fedus et al., 2020; Kozuno et al., 2021). Retrace (Munos et al., 2016) corrects this bias with importance sampling truncation, while action chunking (Li et al., 2025b) eliminates it by conditioning on the exact behavior-policy trajectory. Park et al. (2025b) show formally that longer effective horizons amplify bootstrapping error, motivating our approach of using a long-horizon value function to provide bootstrap targets for shorter-horizon critics. Learning temporally extended actions has been widely studied in hierarchical RL (Dayan & Hinton, 1992; Dietterich, 2000; Kulkarni et al., 2016; Vezhnevets et al., 2016; 2017; Nachum et al., 2018; Ajay et al., 2021; Shankar & Gupta, 2020; Pertsch et al., 2021; Gehring et al., 2021; Xie et al., 2021) and the options framework (Sutton et al., 1999; Menache et al., 2002; Şimşek & Barto, 2004; Şimşek & Barto, 2007; Konidaris, 2011; Bacon et al., 2017; Bagaria & Konidaris, 2019; Bagaria et al., 2024), though bi-level optimization remains a persistent challenge (Nachum et al., 2018). We collapse this bi-level problem into a single-level objective in a temporally extended action space, where the "high-level" selection is a deterministic advantage-based criterion requiring no additional learned parameters.

3 Background

Offline-to-online RL. We consider an infinite-horizon, fully observable Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, r, \rho, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $T(s' \mid s, a) : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel, $r(s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\rho \in \Delta(\mathcal{S})$ is the initial state distribution, and $\gamma \in [0, 1)$ is the discount factor. We assume access to an offline dataset $\mathcal{D} = \{(s, a, r, s')\}$ collected by a behavior policy π_β. The goal of offline-to-online RL is to find a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ that maximizes the expected discounted return $\eta(\pi) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$, operating in two phases: an *offline phase* that pretrains on $\mathcal{D}$, followed by an *online phase* that fine-tunes with environment interactions added to a growing replay buffer.

TD learning and the bootstrapping bias problem. Actor-critic RL methods (Sutton et al., 1998; Haarnoja et al., 2018) learn a critic $Q_\phi(s, a)$ via the temporal-difference (TD) loss:

$$\mathcal{L}(\phi) = \mathbb{E}_{s_t, a_t, r_t, s_{t+1} \sim \mathcal{D}} \left[\left(Q_\phi(s_t, a_t) - r_t - \gamma \bar{V}(s_{t+1}) \right)^2 \right], \tag{1}$$

where $\bar{V}(s)$ is a target value estimate. In a long-horizon, sparse-reward task with effective horizon $H = 1/(1 - \gamma)$, the 1-step backup propagates reward signal backward by only one step per

gradient update, requiring $O(H)$ updates before the initial state receives any signal. Multi-step return methods (Hessel et al., 2018) speed up this process by using a length-n trajectory segment $(s_t, a_t, \ldots, s_{t+n})$ to construct an n-step backup:

$$\hat{V}_{n\text{-step}} := \sum_{j=0}^{n-1} \gamma^j r_{t+j} + \gamma^n Q_{\bar{\phi}}(s_{t+n}, a_{t+n}), \quad a_{t+n} \sim \pi(\cdot \mid s_{t+n}), \tag{2}$$

reducing the effective horizon by a factor of n. However, the n-step estimator is *biased* when data is off-policy (Fedus et al., 2020; Kozuno et al., 2021), meaning the cumulative rewards $\mathbf{r}_{t:t+n}$ in the replay buffer were collected under π_β, not the current policy π, so the sum is no longer an unbiased estimate of the on-policy n-step return.

Action chunking Q-learning. QC (Li et al., 2025b) propose to resolve this tension with *action chunking*: instead of training a single-action critic $Q(s_t, a_t)$, train a *chunked critic* $Q^h(s_t, \mathbf{a}_{t:t+h})$ that takes an entire chunk of h consecutive actions as input. The chunk TD backup is

$$\mathcal{L}_h(\phi) = \mathbb{E}\left[\left(Q^h(s_t, \mathbf{a}_{t:t+h}) - \sum_{j=0}^{h-1} \gamma^j r_{t+j} - \gamma^h \max_{i \le N} \bar{Q}^h(s_{t+h}, a^{(i)}_{t+h:t+2h})\right)^2\right], \tag{3}$$

where $\{a^{(i)}_{t+h:t+2h}\}_{i=1}^N$ are N candidate action chunks sampled from π_β at s_{t+h}. The max over N samples, known as EMAQ (Ghasemipour et al., 2021), approximates the best-of-N bootstrap and ensures the critic pushes above the behavior value. Crucially, because the full chunk $\mathbf{a}_{t:t+h}$ is taken directly from the data and held fixed during the backup, the n-step bias is *eliminated*: there is no policy mismatch between the actions that generated the rewards $\mathbf{r}_{t:t+h}$ and the actions that the critic conditions on (Li et al., 2025a;b).

Implicit value backup via expectile regression. To avoid explicit action maximization in the TD target, one can learn a value function $V_\xi(s)$ to implicitly approximate $\max_a Q(s, a)$ using the *expectile loss* (Kostrikov et al., 2021):

$$\mathcal{L}_V(\xi) = \mathbb{E}_{(s_t, \mathbf{a}_{t:t+h}) \sim \mathcal{D}}\left[f^{\kappa_V}_{\exp}\left(\bar{Q}^h(s_t, \mathbf{a}_{t:t+h}) - V_\xi(s_t)\right)\right], \tag{4}$$

where $f^{\kappa_V}_{\exp}(u) = |\kappa_V - \mathbf{1}_{[u<0]}|\, u^2$. At the optimum, $V_\xi(s)$ approximates the κ_V-expectile of $Q(s, \cdot)$ over the data distribution; with κ_V close to 1, this gives an upper envelope that approximates $V^*(s)$ under the EMAQ-boosted critic.

Flow-matching behavior cloning. We parameterize π_β as a flow-matching model (Lipman et al., 2023) following FQL (Park et al., 2025c), trained with the objective:

$$\mathcal{L}_{\text{BC}}(\theta) = \mathbb{E}_{(s_t, \mathbf{a}_{t:t+h}) \sim \mathcal{D},\, x_0 \sim \mathcal{N}(0,I),\, \tau \sim \mathcal{U}[0,1]}\left[\left\| v_\theta(s_t, x_\tau, \tau) - (\mathbf{a}_{t:t+h} - x_0)\right\|^2\right], \tag{5}$$

where $x_\tau = (1-\tau)x_0 + \tau\, \mathbf{a}_{t:t+h}$ is the linear interpolation along the flow path. At inference, N candidate action chunks are drawn by solving the flow ODE from $x_0 \sim \mathcal{N}(0, I)$, and the best is selected by the critic. This expressive policy class captures the multimodal, temporally correlated structure of manipulation datasets far better than Gaussian policies (Chi et al., 2023; Li et al., 2025b).

4 The Case for Adaptive Action Chunking

Both **QC** (Li et al., 2025b) and **DQC** (Li et al., 2026) fix the policy chunk size uniformly across all states. We argue that this is suboptimal: the beneficial chunk size varies across states within a single trajectory, and a globally fixed size is always a compromise. We further show that the natural approach to adapting the chunk size per state i.e. selecting by comparing Q^k values directly is fundamentally broken.

4.1 State-Dependent Commitment Length

Different phases of a manipulation task require qualitatively different chunk sizes. During free-space motion e.g., reaching toward a target cube, executing a long action sequence open-loop is both safe and effective. The BC flow policy produces temporally coherent trajectories, and longer open-loop execution accelerates credit assignment by propagating value over more timesteps per update. Near a

contact event such as grasping or placement, the same long-sequence execution becomes harmful. Small perturbations at a contact boundary propagate irreversibly through the subsequent physics, so the policy must react to state feedback at each step. A short chunk size is appropriate here.

A globally fixed chunk size cannot simultaneously serve both phases. A large chunk degrades performance near contacts by ignoring state feedback. A small chunk wastes the credit-assignment benefit of multi-step backups during free-space motion and yields incoherent online exploration. This motivates maintaining a discrete set $\mathcal{K} = \{k_1, \ldots, k_{|\mathcal{K}|}\}$ of candidate chunk sizes and selecting the commitment length $k^*(s_t) \in \mathcal{K}$ adaptively at each state.

4.2 Why Direct Q^k Comparison Fails

The natural adaptive selection rule compares the best action chunk across all candidate sizes using the learned critics:

$$k^*(s_t),\ a^* \ = \ \underset{k\in\mathcal{K},\ \mathbf{a}_{t:t+k}\sim\pi_\beta^N}{\arg\max}\ Q^k(s_t, \mathbf{a}_{t:t+k}). \tag{6}$$

This selector fails due to two compounding problems:

Discount-scale mismatch. In sparse-reward tasks, intermediate rewards are nearly zero for the vast majority of transitions. The Bellman equation for Q^k therefore simplifies to

$$Q^k(s_t, \mathbf{a}_{t:t+k}) \ = \ \underbrace{\sum_{j=0}^{k-1} \gamma^j r_{t+j}}_{\approx 0} + \gamma^k V^h(s_{t+k}) \ \approx \ \gamma^k V^h(s_{t+k}). \tag{7}$$

Since $\gamma < 1$, the factor γ^k is strictly decreasing in k. Consequently, $Q^{k_1} > Q^{k_2} > \cdots > Q^h$ for nearly every state, regardless of which chunk size actually yields a better policy. The selector degenerates to always choosing the smallest k in $\mathcal{K}$.

State-dependent baseline mismatch. Dividing by γ^k to remove the discount factor gives:

$$\underset{k}{\arg\max}\ \frac{Q^k(s_t, \mathbf{a}_{t:t+k})}{\gamma^k} \ \approx \ \underset{k}{\arg\max}\ V^h(s_{t+k}). \tag{8}$$

This corrects the discount-scale issue but exposes a subtler one. In states far from any reward, which constitute the majority of states in sparse-reward tasks, $V^h(s_{t+k}) \approx \epsilon$ for all $k \in \mathcal{K}$, where ϵ is small. The differences across chunk sizes are then dominated by function approximation errors in the Q^k networks rather than by genuine differences in planning quality. The argmax reflects network noise, not which chunk size is actually better.

Both failure modes stem from the same root cause: comparing raw Q^k values lacks a *per-scale reference* that accounts for the baseline value achievable at that chunk size under the data distribution. We resolve this in Section 5 by subtracting a per-scale value baseline $V^k(s_t)$ and normalizing by γ^k, yielding an advantage criterion that avoids both collapse and noise amplification.

5 Adaptive Q-Chunking

AQC selects the commitment length $k^*(s_t) \in \mathcal{K}$ adaptively at each state, re-querying the policy every $k^*(s_t)$ steps rather than uniformly every h steps. We first derive the selection criterion from first principles, then describe how its components are trained jointly.

5.1 The Per-Scale Advantage Criterion

Suppose we maintain critics $\{Q^k\}_{k\in\mathcal{K}}$ for a discrete set of chunk sizes $\mathcal{K} = \{k_1, \ldots, h\}$, and a behavior policy π_β that can generate candidate action chunks of length up to h. The ideal selection rule would pick the chunk size and action sequence with the best true future return:

$$k^*(s_t),\ a^* \ \in \ \underset{k\in\mathcal{K},\ \mathbf{a}_{t:t+k}\sim\pi_\beta}{\arg\max}\ Q^{k,*}(s_t, \mathbf{a}_{t:t+k}), \tag{9}$$

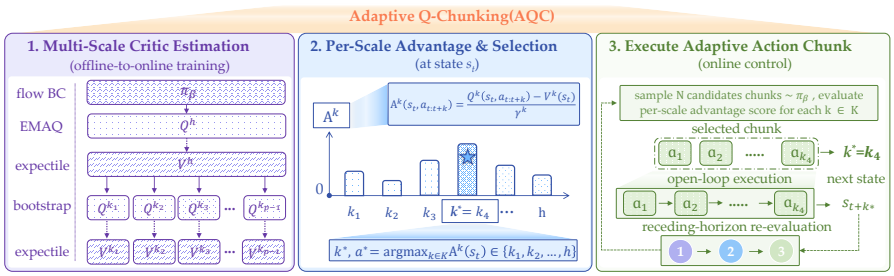


Figure 2: **Overview of AQC.** (1) Train a flow-BC policy π_β alongside a long-horizon critic Q^h and per-scale partial critics $\{Q^{k_i}\}$ bootstrapped from V^h. (2) Score N candidate chunks at every horizon k via the discount-normalized advantage. (3) Select the best (k^*, a^*) and execute open-loop for k^* steps in a receding-horizon loop. Training and inference details are given in Sections 5.2 and 5.3. Here p denotes $|\mathcal{K}|$

where $Q^{k,*}$ is the optimal k-step action-value function. Since we only have learned approximations Q^k and comparing their raw values across different k systematically fails (Section 4.2), we instead compare the *discount-normalized advantage* relative to a per-horizon baseline:

$$\text{score}(k, \mathbf{a}_{t:t+k}) \;:=\; \frac{Q^k(s_t, \mathbf{a}_{t:t+k}) - V^k(s_t)}{\gamma^k}. \tag{10}$$

Here $V^k(s_t)$ estimates the expected k-step return of the behavior policy from s_t. The subtraction removes state-dependent scale at horizon k, and division by γ^k removes the horizon-dependent discount. We establish the noise-immunity of this criterion formally:

Proposition 5.1 (Noise immunity of the advantage selector). *Let $\delta_k(s) := Q^k(s, \mathbf{a}_{t:t+k})/\gamma^k - V^k(s)/\gamma^k$ denote the discount-normalized advantage. In a region where $V^h(s) \le \epsilon$ for all reachable states, and assuming function approximation errors are bounded by σ,*

$$|\delta_k(s)| \;\le\; \epsilon + 2\sigma. \tag{11}$$

When $\epsilon \ll \sigma$, the advantage is dominated by approximation noise and all chunk sizes score near-zero. By contrast, the uncorrected selector $\arg\max_k Q^k/\gamma^k$ lacks this safeguard: when all scores are $\approx \epsilon + \sigma_k$, the argmax picks the scale with the largest positive noise σ_k, producing a deterministic but systematically biased choice.

The key implication: the advantage criterion converts a biased wrong answer into an unbiased near-random choice when no genuine signal exists, which is strictly preferable. A full proof appears in Appendix I.12.

5.2 Training

To instantiate Equation (10), we need Q^k and V^k for each $k \in \mathcal{K}$, trained jointly alongside π_β.

Long-horizon critic and value. The backbone of the pipeline is the long-horizon critic $Q^h_\phi(s_t, \mathbf{a}_{t:t+h})$, trained with the EMAQ (Ghasemipour et al., 2021) h-step TD loss:

$$\mathcal{L}_h(\phi) = \mathbb{E}_{\mathcal{D}}\left[\left(Q^h_\phi(s_t, \mathbf{a}_{t:t+h}) - \sum_{j=0}^{h-1}\gamma^j r_{t+j} - \gamma^h \max_{i\le N} \bar{Q}^h_\phi(s_{t+h}, a^{(i)}_{t+h:t+2h})\right)^2\right], \tag{12}$$

where $\{a^{(i)}_{t+h:t+2h}\}_{i=1}^N \sim \pi_\beta(\cdot \mid s_{t+h})$ and $\bar{Q}^h$ is the EMA target network. Correspondingly, we fit $V^h_\xi(s)$ via expectile regression on Q^h:

$$\mathcal{L}_{V^h}(\xi) = \mathbb{E}_{\mathcal{D}}\left[f^{\kappa_V}_{\text{exp}}\left(\bar{Q}^h_\phi(s_t, \mathbf{a}_{t:t+h}) - V^h_\xi(s_t)\right)\right]. \tag{13}$$

With κ_V close to 1, V^h_ξ approximates the upper envelope of h-step returns, serving both as the baseline for $k = h$ and as the bootstrap target for all partial critics.

		puzzle-3x3 (5 tasks)	scene (5 tasks)	cube-double (5 tasks)	cube-triple (5 tasks)	cube-quad. (5 tasks)	Overall (25 tasks)
1-step TD	IQL (Kostrikov et al., 2021)	0 → 20	0 → 39	0 → 0	0 → 0	0 → 0	0 → 12
	RLPD (Ball et al., 2023)	– → **100**	– → 94	– → **99**	– → 41	– → 0	– → 67
	FQL (Park et al., 2025c)	99 → **100**	52 → **95**	32 → 76	2 → 18	0 → 3	37 → 58
n-step TD	FQL-n	99 → **100**	21 → 70	9 → 77	1 → 1	7 → 36	27 → 57
Q-chunking	QC-FQL (Li et al., 2025b)	64 → **100**	86 → **99**	42 → **100**	3 → 53	2 → 77	38 → 86
	QC (Li et al., 2025b)	**100** → **100**	82 → **99**	66 → **98**	5 → 64	3 → 73	52 → 86
	DQC (Li et al., 2026)	86 → **97**	56 → 89	15 → 55	1 → 0	2 → 20	32 → 52
	SEAR (Nagy et al., 2026)	86 → **97**	72 → 92	45 → 78	5 → 22	4 → 38	42 → 65
Ours	AQC	**100** → **100**	**98** → **100**	**90** → **100**	**13** → **90**	**9** → **88**	**62** → **96**

Table 1: **Offline-to-online RL results on OGBench.** Each cell reports *offline* → *online* success rate (%) averaged over 5 tasks and 4 seeds. Numbers in bold are within 95% of the best in each column. Full task-level breakdowns with mean and standard deviations are in Appendix E.

Partial critics and per-scale baselines. For each $k \in \mathcal{K}\backslash\{h\}$, we train a partial critic $Q^k_\psi(s_t, \mathbf{a}_{t:t+k})$ that takes only the k-step prefix as input:

$$\mathcal{L}_k(\psi) = \mathbb{E}_{\mathcal{D}}\left[\left(Q^k_\psi(s_t, \mathbf{a}_{t:t+k}) - \sum_{j=0}^{k-1} \gamma^j r_{t+j} - \gamma^k \bar{V}^h_\xi(s_{t+k})\right)^2\right]. \tag{14}$$

Using V^h as the bootstrap target is a deliberate design choice. The long-range value from s_{t+k} onward is already encoded in V^h, so Q^k only needs to fit the k-step residual. A 1-step value function would require chaining $O(H)$ TD updates to propagate credit, accumulating error at each step. A formal bound showing that V^h bootstrap yields tighter sub-optimality guarantees appears in Proposition H.21 (Proof I.13).

Each Q^k is a separate network; it cannot be recovered by truncating Q^h, since Q^h is conditioned on the full h-step chunk. We then fit per-scale baselines $V^k_\zeta(s)$ via expectile regression on the corresponding partial critics:

$$\mathcal{L}_{V^k}(\zeta) = \mathbb{E}_{\mathcal{D}}\left[f^{\kappa_V}_{\exp}\left(\bar{Q}^k_\psi(s_t, \mathbf{a}_{t:t+k}) - V^k_\zeta(s_t)\right)\right]. \tag{15}$$

These networks share the same architecture as V^h_ξ and add negligible training overhead.

Behavior policy. We train π_β with the flow-matching BC objective (Equation (5)) on the full h-step chunk distribution.

5.3 Inference

At each state s_t, AQC samples N full-length candidate chunks from π_β and for each chunk, evaluates the per-scale advantage score (Eq.10) across all horizons $k \in \mathcal{K}$. The selected chunk and commitment horizon are:

$$k^*(s_t),\ a^* \ = \ \underset{k\in\mathcal{K},\ i\le N}{\arg\max}\ \text{score}(k, a^{(i)}_{t:t+k}). \tag{16}$$

The selected chunk a^* is executed open-loop for k^* steps, and the policy is re-queried at s_{t+k^*}. During online fine-tuning, new transitions are added to a replay buffer and all losses continue with mixed offline-online batches.

6 Experimental Results

We conduct experiments to analyze the empirical effectiveness of AQC on a range of long-horizon, sparse-reward domains. In particular, we aim to answer the following questions:

(Q1) *How well does AQC compare to prior offline-to-online RL methods?*
(Q2) *Can AQC enhance large-scale VLAs with offline RL?*
(Q3) *Which components of AQC are most critical?*

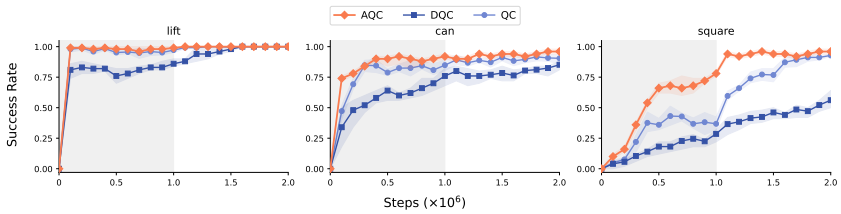


Figure 3: **Robomimic results.** Success rate vs. environment steps on three tasks. The first 1M steps are offline and the next 1M steps are online. (5 seeds)

6.1 Environments and Datasets

We consider long-horizon, sparse-reward manipulation domains across three benchmark suites. For OGBench (Park et al., 2025a), we use a total of five domains. Each domain provides five tasks of increasing difficulty; we report success rates averaged across all five tasks and four seeds, with 95% stratified bootstrap confidence intervals. We evaluate on `scene-sparse`, `puzzle-3x3-sparse`, `cube-double`, `cube-triple`, and `cube-quadruple`. We use datasets ranging from 1M to 100M transitions based on task difficulty. For Robomimic (Mandlekar et al., 2021), we use three tasks: `lift`, `can`, `square`; with multi-human datasets, evaluated over five seeds. Additionally, we evaluate **AQC** by finetuning GR00T N1.6 (Bjorck et al., 2025) using offline RL on the 24 RoboCasa-GR1 tabletop manipulation tasks (Nasiriany et al., 2024). See Appendix C for more details about experiments setup and dataset.

6.2 Comparisons

We compare against prior methods that speed up value backup as well as the previous best offline-to-online RL methods: **IQL** (Kostrikov et al., 2021), **RLPD** (Ball et al., 2023), **FQL** (Park et al., 2025c), **FQL-n**, **QC**, **QC-FQL** (Li et al., 2025b), **DQC** (Li et al., 2026), and **SEAR** (Nagy et al., 2026).

6.3 How well does AQC compare to prior methods?

Table 1 reports offline and online success rates across all five domains of OGBench. **AQC** achieves the highest online success rates on the three hardest domains (`cube-double`, `cube-triple`, `cube-quadruple`). On the easier domains (`scene-sparse`, `puzzle-3x3`), all chunking methods converge to near-perfect online performance. Across all domains, **AQC** consistently outperforms **QC**, **DQC**, and **SEAR**, with the gap widening as horizon length increases. Figure 3 reports training curves on `lift`, `can`, and `square` from Robomimic. **AQC** matches or exceeds **QC** on all three tasks. The gains are most pronounced on `square`, which involves a contact-rich peg-insertion step and directly illustrates the core motivation of our method: during the pre-grasp reaching phase, the motion is predictable and **AQC** selects longer chunks for temporally coherent trajectory; but once the nut contacts the peg, small misalignments require rapid corrective feedback, and **AQC** adaptively switches to shorter chunks to react to state changes. On `lift` and `can`, which involve simpler pick-and-place motions with fewer contact transitions, the advantage of adaptive selection is smaller but still non-negative, suggesting that even in these tasks there are phases where committing to shorter or longer chunks is beneficial.

6.4 Can AQC enhance large-scale VLAs with offline RL?

Figure 4 evaluates **AQC** on RoboCasa-GR1 tabletop manipulation tasks (Nasiriany et al., 2024), a challenging multi-task setting that tests whether our method generalizes to large-scale vision-language-action models. The 24 tasks span diverse contact-rich manipulation skills including picking, placing, opening containers, and operating household appliances. These tasks are particularly demanding for RL fine-tuning: VLAs are heavily overparameterized, the state-action space is high-dimensional, and collecting additional online data is impractical, making effective offline RL from limited suboptimal rollouts essential.

We use GR00T N1.6 (Bjorck et al., 2025) as the base VLA, which autoregressively generates 16-step action chunks conditioned on the current observation. Our protocol has two stages. First, we finetune GR00T N1.6 via supervised learning on 24k expert demonstrations spanning all 24 tasks, yielding a multi-task behavior policy. We deploy this policy to collect 300 offline rollouts per

task. Then, we fine-tune the base policy using behavior cloning on the combined expert + rollout dataset and use the model as an actor for training critic functions when necessary (Filtered BC).

For **AQC**, we set $h = 16$ to match GR00T N1.6's native chunk size and use $\mathcal{K} = \{1, 4, 8, 16\}$ as candidate commitment horizons. As shown in Figure 4, **AQC** outperforms both **QC** and **DQC** across all 24 tasks, with the largest margins on tasks involving multi-stage contact transitions (e.g., opening a cabinet then manipulating its contents). Across 4 seeds, **AQC** achieves $63.4 \pm 2.8\%$ average success rate compared to the base VLA at 47.6%. Filtered BC provides modest improvements over the base VLA but is fundamentally limited by its inability to leverage value information from suboptimal transitions. These results suggest that **AQC** scales effectively to large policies and high-dimensional observations, and that adaptive chunk selection remains beneficial even when the underlying policy is a multi-task VLA rather than a single-task expert.

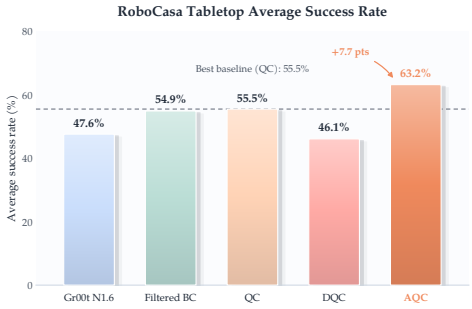


Figure 4: **Enhancing GR00T N1.6 on RoboCasa-GR1 tasks.** Success rates across 24 tabletop manipulation tasks (4 seeds for **AQC**). We compare the base VLA against Filtered BC, **QC**, **DQC**, and **AQC**.

6.5 What makes AQC work?

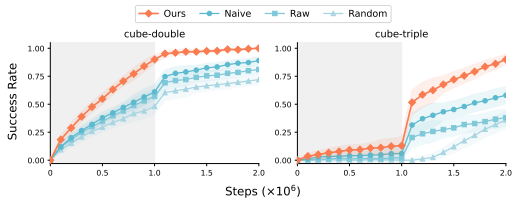


Figure 5: **Advantage criterion ablation** of selection criteria on `cube-double` (left) and `cube-triple` (right). Shaded regions are 95% confidence intervals over 4 seeds.

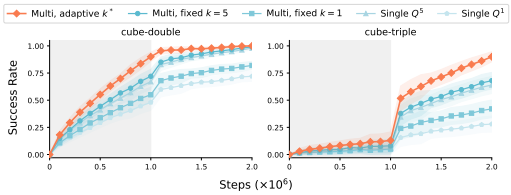


Figure 6: **Multi-scale critics and adaptive selection ablation** on `cube-double` (left), and `cube-triple` (right). Shaded regions are 95% confidence intervals over 4 seeds.

We run ablations on `cube-double` and `cube-triple` with 4 seeds each. The core design in **AQC** is the per-scale advantage selector. Figure 5 compares it against discount-corrected Q values without baselines, raw Q values, and random selection. The raw-Q variant collapses to always selecting $k = 1$, confirming the discount-scale collapse from Section 4. The discount-corrected variant partially recovers on easier tasks but degrades sharply on longer-horizon ones where function approximation noise dominates without a per-scale baseline. Figure 6 decomposes gains from (A) training multiple per-scale critics Q^k and (B) adaptive k^* selection at inference. Multi-critic training alone improves performance over single-critic baselines, confirming that richer value representation helps even without adaptation. Adaptive selection adds further gains on top, with the effect compounding on longer-horizon tasks where contact phases dominate. Additional ablations are in Appendix G.

7 Conclusion

We proposed **Adaptive Q-Chunking (AQC)**, an offline-to-online RL method that adapts action chunk sizes per state via a discount-normalized advantage criterion. By comparing each candidate chunk size to the baseline policy at that horizon, **AQC** resolves the bias toward shorter chunks and the random selection in low-value states that plague naive multi-chunk methods. Our empirical results demonstrate state-of-the-art performance across OGBench, Robomimic and RoboCasa-GR1. Beyond its immediate results, **AQC** suggests a broader principle for offline RL: when multiple learned value functions exist, their raw values are incomparable without a per-scale reference. The advantage-based criterion provides a principled way to convert these incomparable quantities into a shared decision space. We hope this perspective inspires future work on adaptive commitment horizons and multi-scale value-based decision making.

AI USE STATEMENT

We only used Large Language Models (LLMs) to polish the writing and improve the clarity and flow of the text in this paper. No LLMs were used for generating content, ideas, or experimental results.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Reincarnating reinforcement learning: Reusing prior computation to accelerate progress. *Advances in neural information processing systems*, 35:28955–28971, 2022.

Anurag Ajay, Aviral Kumar, Pulkit Agrawal, Sergey Levine, and Ofir Nachum. OPAL: Offline primitive discovery for accelerating offline reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=V69LGwJ0lIN.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in Neural Information Processing Systems*, 30, 2017.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Akhil Bagaria and George Konidaris. Option discovery using deep skill chaining. In *International Conference on Learning Representations*, 2019.

Akhil Bagaria, Ben Abbatematteo, Omer Gottesman, Matt Corsaro, Sreehari Rammohan, and George Konidaris. Effectively learning initiation sets in hierarchical reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

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

Johan Bjorck, Fernando Castaneda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, pp. 02783649241273668, 2023.

Peter Dayan and Geoffrey E Hinton. Feudal reinforcement learning. *Advances in neural information processing systems*, 5, 1992.

Thomas G Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of artificial intelligence research*, 13:227–303, 2000.

William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. Revisiting fundamentals of experience replay. In *International conference on machine learning*, pp. 3061–3071. PMLR, 2020.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1587–1596. PMLR, 2018.

Jonas Gehring, Gabriel Synnaeve, Andreas Krause, and Nicolas Usunier. Hierarchical skills for efficient exploration. *Advances in Neural Information Processing Systems*, 34:11553–11564, 2021.

Seyed Kamyar Seyed Ghasemipour, Dale Schuurmans, and Shixiang Shane Gu. EMaQ: Expected-max q-learning operator for simple yet effective offline and online RL. In *International Conference on Machine Learning*, pp. 3682–3691. PMLR, 2021.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pp. 267–274, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. ISBN 1558608737.

Changyeon Kim, Haone Lee, Younggyo Seo, Kimin Lee, and Yuke Zhu. DEAS: Detached value learning with action sequence for scalable offline RL. *arXiv preprint arXiv:2510.07730*, 2025.

George Dimitri Konidaris. *Autonomous robot skill acquisition*. University of Massachusetts Amherst, 2011.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Tadashi Kozuno, Yunhao Tang, Mark Rowland, Rémi Munos, Steven Kapturowski, Will Dabney, Michal Valko, and David Abel. Revisiting Peng's Q(λ) for modern reinforcement learning. In *International Conference on Machine Learning*, pp. 5794–5804. PMLR, 2021.

Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 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 *Conference on Robot Learning*, pp. 1702–1712. PMLR, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Ge Li, Dong Tian, Hongyi Zhou, Xinkai Jiang, Rudolf Lioutikov, and Gerhard Neumann. TOP-ERL: Transformer-based off-policy episodic reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=N4NhVN3Oph.

Qiyang Li, Zhiyuan Zhou, and Sergey Levine. Reinforcement learning with action chunking. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=XUks1Y96NR.

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.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=AlGy9vbc1B1.

Ajay Mandlekar, Danfei Xu, Josiah 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 *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=JrsfBJtDFdI.

Ishai Menache, Shie Mannor, and Nahum Shimkin. Q-cut—dynamic discovery of sub-goals in reinforcement learning. In *Machine Learning: ECML 2002: 13th European Conference on Machine Learning Helsinki, Finland, August 19–23, 2002 Proceedings 13*, pp. 295–306. Springer, 2002.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

C. F. Maximilian Nagy, Onur Celik, Emiliyan Gospodinov, Florian Seligmann, Weiran Liao, Aryan Kaushik, and Gerhard Neumann. SEAR: Sample efficient action chunking reinforcement learning. *arXiv preprint arXiv:2603.01891*, 2026.

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. *Advances in Neural Information Processing Systems*, 36, 2024.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems*, 2024. URL https://arxiv.org/abs/2406.02523.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=M992mjgKzI.

Seohong Park, Kevin Frans, Deepinder Mann, Benjamin Eysenbach, Aviral Kumar, and Sergey Levine. Horizon reduction makes RL scalable. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=hguaupzLCU.

Seohong Park, Qiyang Li, and Sergey Levine. Flow Q-learning. In *International Conference on Machine Learning*, 2025c. URL https://arxiv.org/abs/2502.02538.

Karl Pertsch, Youngwoon Lee, and Joseph Lim. Accelerating reinforcement learning with learned skill priors. In *Conference on robot learning*, pp. 188–204. PMLR, 2021.

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

Younggyo Seo and Pieter Abbeel. Coarse-to-fine q-network with action sequence for data-efficient reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=VoFXUNc9Zh.

Younggyo Seo, Jafar Uruç, and Stephen James. Continuous control with coarse-to-fine reinforcement learning. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=WjDR48cL3O.

Tanmay Shankar and Abhinav Gupta. Learning robot skills with temporal variational inference. In *International Conference on Machine Learning*, pp. 8624–8633. PMLR, 2020.

Özgür Şimşek and Andrew G Barto. Using relative novelty to identify useful temporal abstractions in reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 95, 2004.

Özgür Şimşek and Andrew G. Barto. Betweenness centrality as a basis for forming skills. Working-paper, University of Massachusetts Amherst, April 2007.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

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.

Dong Tian, Ge Li, Hongyi Zhou, Onur Celik, and Gerhard Neumann. Chunking the critic: A transformer-based soft actor-critic with n-step returns. *arXiv preprint arXiv:2503.03660*, 2025.

Alexander Vezhnevets, Volodymyr Mnih, Simon Osindero, Alex Graves, Oriol Vinyals, John Agapiou, et al. Strategic attentive writer for learning macro-actions. *Advances in neural information processing systems*, 29, 2016.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *International conference on machine learning*, pp. 3540–3549. PMLR, 2017.

Kevin Xie, Homanga Bharadhwaj, Danijar Hafner, Animesh Garg, and Florian Shkurti. Latent skill planning for exploration and transfer. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jXe91kq3jAq.

Haichao Zhang, Wei Xu, and Haonan Yu. Policy expansion for bridging offline-to-online reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=-Y34L45JR6z.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Zhiyuan Zhou, Andy Peng, Qiyang Li, Sergey Levine, and Aviral Kumar. Efficient online reinforcement learning fine-tuning need not retain offline data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HN0CYZbAPw.

A Limitations and Future Work

Despite its empirical effectiveness, our method has several limitations that suggest directions for future work. First, the method selects from a fixed discrete candidate set $\mathcal{K}$, which requires some domain knowledge to construct and cannot discover chunk sizes outside the predefined set; learning $\mathcal{K}$ itself—for instance through a continuous relaxation of the commitment horizon or a state-conditioned proposal mechanism—would remove this design choice. Second, like all action-chunking methods, our approach executes selected chunks open-loop, meaning even $k{=}1$ chunks inherit the limited reactivity of the flow-BC policy π_β, which was trained to produce h-step sequences; integrating a closed-loop low-level executor for contact-rich phases could address this gap. Third, the critic training overhead scales linearly with $|\mathcal{K}|$, and while small in practice, weight-sharing schemes across scales such as a single multi-scale critic with horizon-embedding inputs could improve efficiency for large candidate sets. The selection also depends on the quality and diversity of π_β as a candidate source, which affects robustness when N is small or the policy distribution shifts during online fine-tuning.

B Algorithm

Algorithm 1 AQC

Require: Chunk sizes $\mathcal{K} = \{k_1, \dots, h\}$, samples N, expectile κ_V, warmup

Initialize critics Q^h_ϕ, $\{Q^k_\psi\}_{k\in\mathcal{K}\setminus\{h\}}$
Initialize values V^h_ξ, $\{V^k_\zeta\}_{k\in\mathcal{K}\setminus\{h\}}$
Initialize flow policy π_β

Offline pretraining:
while not converged **do**
  Sample minibatch $(s_t, \mathbf{a}_{t:t+h}, \sum_{j=0}^{h-1}\gamma^j r_{t+j}, s_{t+h}) \sim \mathcal{D}$
  Update Q^h_ϕ via Equation (12) (EMAQ h-step TD)
  Update V^h_ξ via Equation (13) (expectile regression on Q^h)
  for all $k \in \mathcal{K} \setminus \{h\}$ **do**
    Update Q^k_ψ via Equation (14) (k-step TD + V^h bootstrap)
    Update V^k_ζ via Equation (15) (expectile regression on Q^k)
  Update π_β via flow-matching BC (Equation (5))

Online fine-tuning:
Initialize replay buffer $\mathcal{R} \leftarrow \mathcal{D}$
for each environment step t **do**
  Observe s_t
  Sample N full-length chunks $\{a^{(i)}_{1:h}\}_{i=1}^N \sim \pi_\beta(\cdot \mid s_t)$
  for all $k \in \mathcal{K}$ **do**
    Score: $\text{score}(k, a^{(i)}_{1:k}) = \dfrac{Q^k(s_t, a^{(i)}_{1:k}) - V^k(s_t)}{\gamma^k}$ for all i, using prefix $a^{(i)}_{1:k}$
  Select $k^*, a^* \leftarrow \arg\max_{k,i} \text{score}(k, a^{(i)}_{1:k})$
  Execute a^* open-loop for k^* steps
  Store transitions in $\mathcal{R}$
  if $t \geq$ warmup **then**
    Sample mixed batch from $\mathcal{R}$
    Update all networks via losses Equations (12) to (15) and Equation (5)

C Experimental Setup

C.1 OGBench environments

We consider five manipulation domains from OGBench (Park et al., 2025a) as shown in Figure 7, same as the ones used in Li et al. (2025b): `scene-sparse`, `puzzle-3x3-sparse`, `cube-double`,

`cube-triple`, and `cube-quadruple`. Each domain provides five evaluation tasks of increasing difficulty; we report success rates averaged across all five tasks. We use "play"-style datasets collected by non-Markovian expert policies with temporally correlated noise. We use the default 1M-transition datasets except for `cube-quadruple`, where we use the 100M-transition dataset. All domains use binary or near-binary sparse reward functions and a UR5e robot arm with 5D end-effector control.

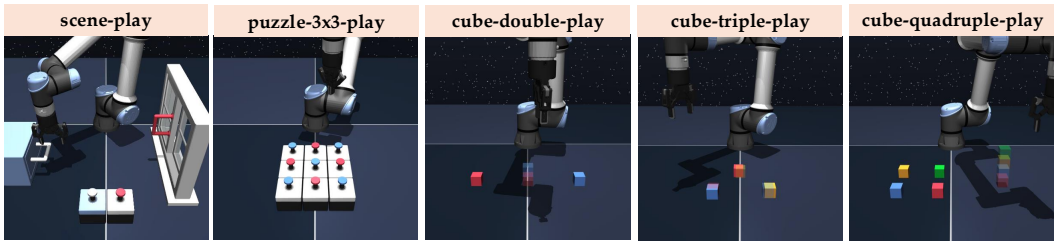


Figure 7: **OGBench tasks.** Some example tasks from the OGBench benchmark that we consider in our work.

For `scene-sparse` and `puzzle-3x3-sparse`, we sparsify the reward function such that the agent receives -1 when the task is incomplete and 0 upon completion. For the cube domains, the reward is $-n_{\text{wrong}}$ where n_{wrong} is the number of cubes at incorrect positions; the episode terminates when all cubes are correctly placed (reward 0). We now describe each domain briefly:

Cube domains. The agent controls a UR5e arm to pick and place cubes into target configurations. `cube-double` requires arranging two cubes, `cube-triple` three cubes, and `cube-quadruple` four. These domains test multi-object manipulation and long-horizon sequential reasoning. The cube domains become dramatically harder as the number of cubes increases, since permuting multiple blocks requires non-trivial sequential planning.

Scene domain. This domain involves manipulating everyday objects — a drawer, a window, a cube, and two button locks — into a target configuration. Pressing a button toggles whether the corresponding object can be moved. Evaluation tasks require multi-step sequences such as unlocking the drawer, opening it, placing a cube inside, and closing it. We use a sparsified reward variant where success yields 0 and failure yields -1.

Puzzle domains. These domains implement the "Lights Out" puzzle game using a robot arm that must physically press buttons on a grid. Pressing a button toggles its color and the colors of adjacent buttons. The goal is to reach a target color configuration. We use sparsified rewards: -1 until the target configuration is reached, 0 upon success.

Domain	Dataset Size	Episode Length	Action Dim.
`scene-sparse-*`	1M	750	5
`puzzle-3x3-sparse-*`	1M	500	5
`cube-double-*`	1M	500	5
`cube-triple-*`	3M	1000	5
`cube-quadruple-100M-*`	100M	1000	5

Table 2: **OGBench domain metadata.** Dataset sizes, episode lengths, and action dimensions for all five domains. Each domain provides five evaluation tasks (indicated by `*`).

C.2 Robomimic environments

We evaluate on three tasks from the robomimic benchmark (Mandlekar et al., 2021), using the multi-human datasets collected by six human operators. Each dataset contains 300 successful demonstrations. All three tasks use binary task-completion rewards: -1 when the task is incomplete and 0 upon success.

- `lift`: The robot arm must grasp a small cube and lift it. This is the simplest task in the benchmark, testing basic grasping and vertical motion.

- `can`: The robot arm must pick up a soda can and place it into a smaller container bin. This requires more precise manipulation than `lift`, as the agent must orient the can correctly for insertion.
- `square`: The robot arm must pick up a square nut and thread it onto a peg. The nut is slightly larger than the peg, requiring precise alignment and contact-rich insertion, making this the most difficult of the three tasks.

Task	Dataset Size	Episode Length	Action Dim.
`lift`	31,127	500	7
`can`	62,756	500	7
`square`	80,731	500	7

Table 3: **Robomimic task metadata.** Number of transitions, episode lengths, and action dimensions.

C.3 RoboCasa-GR1 environments

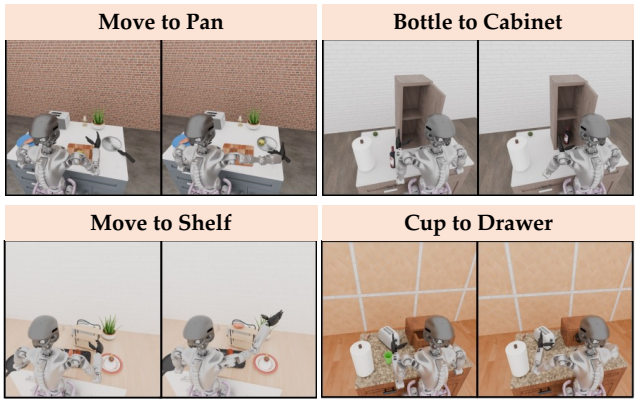


Figure 8: **RoboCasa-GR1 tasks.** Some example tasks from the RoboCasa-GR1 tabletop manipulation benchmark. The rendered images of these tasks are taken from Bjorck et al. (2025).

RoboCasa-GR1 (Nasiriany et al., 2024) is a benchmark of tabletop manipulation tasks designed for the GR1 humanoid robot platform. It comprises 24 tasks spanning diverse manipulation skills including picking, placing, pouring, grasping, and insertion. These tasks require controlling a high-DoF humanoid upper body, making them substantially more complex than standard arm-only manipulation benchmarks. The action space includes joint-level control of the torso, arms, and grippers, with observations consisting of joint positions, velocities, and object states.

The 24 tasks fall into two categories. Six base pick-and-place tasks involve manipulating everyday objects into containers: placing a can, cup, or bottle into a drawer; placing a milk carton or potato into a microwave; and placing a bottle or wine bottle into a cabinet. Eighteen novel generalization tasks (`PosttrainPnPNovel`) evaluate zero-shot transfer to unseen object–container pairs (e.g., cuttingboard-to-basket, tray-to-tiered-shelf, plate-to-cardboard-box). All tasks use the `GR1ArmsAndWaistFourierHands` embodiment, which provides a 44-dimensional action space across 8 body parts (left/right arm, left/right hand, left/right leg, neck, waist) and a matching 44-dimensional proprioceptive state space. Each task provides 1,000 successful human teleoperation demonstrations.

D Implementation Details

D.1 Computational Resources

We use NVIDIA RTX 4090 GPUs for all experiments. A single OGBench offline-to-online run takes approximately 5–7 GPU hours, depending on the domain and dataset size. The 100M-transition

datasets (`cube-quadruple`) require periodic chunking of the offline data to fit in CPU memory, adding roughly 2 hours per run. For the RoboCasa-GR1 experiments, each training run takes around 20 hours due to the high-dimensional physics simulation.

To reproduce the main OGBench results in Table 1, we estimate $\underbrace{6}_{\text{hours}} \times \underbrace{2}_{\text{methods}} \times \underbrace{25}_{\text{tasks}} \times \underbrace{4}_{\text{seeds}} \approx 1{,}200$ GPU hours. The Robomimic experiments require $\underbrace{10}_{\text{hours}} \times \underbrace{2}_{\text{methods}} \times \underbrace{3}_{\text{tasks}} \times \underbrace{5}_{\text{seeds}} \approx 300$ GPU hours. The RoboCasa-GR1 experiments take approximately $\underbrace{20}_{\text{hours}} \times \underbrace{4}_{\text{methods}} \approx 80$ GPU hours. In total, reproducing all results in this paper requires roughly 1,600 GPU hours.

D.2 OGBench and Robomimic Experiments

For tasks that are already evaluated in prior work, we use the reported results. Specifically, results of `cube-double`, `cube-triple`, `cube-quadruple`, `scene-sparse`, `puzzle-3x3-sparse`, `lift`, `can`, and `square` are taken from Li et al. (2025b). Otherwise, we implement the baselines in our codebase and evaluated according to the same protocol used in those works.

AQC. We sweep over expectile values $\kappa_v \in \{0.5, 0.7, 0.9, 0.93, 0.95, 0.99\}$, horizon lengths $h \in \{5, 10, 25\}$, and scale sets $\mathcal{K} \in \{\{1, 5\}, \{1, 5, 10\}, \{1, 5, 10, 25\}\}$. For each environment, we report the best configuration selected in Table 5.

DQC. DQC decouples the critic horizon from the policy execution horizon, using a long-horizon critic for credit assignment and a shorter-horizon policy for robust execution. We adapt DQC from its original goal-conditioned formulation (Li et al., 2026) to the reward-based setting by removing goal conditioning entirely and training on the environment's direct reward signal. Concretely, we train five networks:

1. $Q_\phi(s, a_1, \ldots, a_h) : \mathcal{S} \times \mathcal{A}^h \mapsto \mathbb{R}$ — the long-horizon (full-chunk) critic. An ensemble of $N_Q = 2$ networks is used.
2. $Q_\psi(s, a_1, \ldots, a_{h_a}) : \mathcal{S} \times \mathcal{A}^{h_a} \mapsto \mathbb{R}$ — the short-horizon (partial) critic, trained via expectile distillation from Q_ϕ. An ensemble of $N_Q = 2$ networks is used.
3. $V_\xi(s) : \mathcal{S} \mapsto \mathbb{R}$ — the value function, trained via expectile regression on the partial critic Q_ψ. Provides the TD bootstrap for Q_ϕ.
4. $\bar{Q}_\phi$, $\bar{Q}_\psi$, $\bar{V}_\xi$ — exponential-moving-average target networks for each of the above, updated with decay $\tau = 0.005$.
5. $f_\omega(s, x, u) : \mathcal{S} \times \mathbb{R}^{h_a d_a} \times [0, 1] \mapsto \mathbb{R}^{h_a d_a}$ — the flow-matching behavior policy, parameterized by a velocity prediction network.

Given a transition $(s_t, a_{t:t+h-1}, s_{t+h}, r_t^{(h)}) \sim \mathcal{D}$ where $r_t^{(h)} = \sum_{j=0}^{h-1} \gamma^j r_{t+j}$ and continuation mask $m_t = \prod_{j=0}^{h-1} \gamma_{\text{cont}}(s_{t+j})$, the training objectives are:

(1) Long-horizon critic loss. The full-chunk critic Q_ϕ is trained with an h-step TD target bootstrapped from the value function:

$$L(\phi) = \left(Q_\phi(s_t, a_{t:t+h-1}) - r_t^{(h)} - \gamma^h m_t \bar{V}_\xi(s_{t+h}) \right)^2 . \tag{17}$$

The value target $\bar{V}_\xi$ (from the EMA network) encodes the long-range return from s_{t+h} onward, avoiding $O(H/h)$ chained 1-step TD updates.

(2) Partial critic distillation loss. The partial critic Q_ψ is trained to approximate the upper envelope of the long-horizon critic Q_ϕ via κ_d-expectile regression:

$$L(\psi) = \mathbb{E}_2^{\kappa_d} \left[\bar{Q}_\phi(s_t, a_{t:t+h-1}) - Q_\psi(s_t, a_{t:t+h_a-1}) \right] , \tag{18}$$

where $\mathbb{E}_2^{\kappa_d}(x) = |\kappa_d - \mathbb{1}(x < 0)| x^2$ is the asymmetric expectile loss and $\bar{Q}_\phi$ is the EMA target of Q_ϕ. With $\kappa_d > 0.5$, Q_ψ learns an optimistic approximation of Q_ϕ restricted to the first h_a actions.

(3) Value implicit backup loss. The value function V_ξ is trained via κ_b-expectile regression on the partial critic:

$$L(\xi) = \mathbb{E}_2^{\kappa_b}\left[\bar{Q}_\psi(s_t, a_{t:t+h_a-1}) - V_\xi(s_t)\right], \tag{19}$$

where $\bar{Q}_\psi$ is the EMA target of Q_ψ and $\kappa_b > 0.5$ controls the degree of optimism. This loss is co-trained with the distillation loss in a single backward pass.

(4) Flow-matching behavior policy loss. The behavior policy is trained with standard conditional flow matching:

$$L(\omega) = \|f_\omega(s_t, ua_{t:t+h_a-1} + (1-u)z_t, u) - (a_{t:t+h_a-1} - z_t)\|_2^2, \tag{20}$$

where $u \sim U([0,1])$ and $z_t \sim \mathcal{N}(0, I_{h_a d_a})$. Note the flow is trained on h_a-step chunks, not h-step.

Policy inference. At inference time, we sample $N = 32$ candidate h_a-step action chunks from the flow-matching policy via Euler integration, score each by the partial critic ensemble $\bar{Q}_\psi$, and select the highest-scoring chunk.

Hyperparameter Tuning. We sweep over backup expectile $\kappa_b \in \{0.5, 0.7, 0.9, 0.93, 0.95, 0.99\}$, distillation expectile $\kappa_d \in \{0.5, 0.8\}$, horizon length $h \in \{5, 10, 25\}$, and policy chunk size $h_a \in \{1, 5, 25\}$. Task-specific hyperparameters are listed in Table 5.

Parameter	Value
Optimizer	AdamW
Learning rate	3×10^{-4}
Batch size	256
Discount γ	0.99
Network width	512
Network depth	4 hidden layers
EMA decay τ	0.005
Ensemble size N_Q	2
UTD ratio	1
Number of flow steps	10
Number of offline training steps	1,000,000
Number of online environment steps	1,000,000

Table 4: **Common hyperparameters for OGBench and Robomimic experiments.** Shared across all tasks unless otherwise noted in Table 5.

	AQC			DQC			
Environment	κ_v	h	$\mathcal{K}$	κ_b	κ_d	h	h_a
`cube-double`	0.9	5	$\{1, 5\}$	0.99	0.8	5	1
`cube-triple`	0.9	5	$\{1, 5\}$	0.7	0.8	10	5
`cube-quadruple`	0.9	10	$\{1, 5, 10\}$	0.95	0.8	10	5
`scene-sparse`	0.95	5	$\{1, 5\}$	0.9	0.5	5	1
`puzzle-3x3-sparse`	0.95	5	$\{1, 5\}$	0.9	0.8	10	5
`lift`	0.93	5	$\{1, 5\}$	0.93	0.8	5	1
`can`	0.9	5	$\{1, 5\}$	0.95	0.8	5	1
`square`	0.9	5	$\{1, 5\}$	0.9	0.5	10	5

Table 5: **Task-specific hyperparameters for AQC and DQC.** Best configuration per environment.

D.3 Selecting $\mathcal{K}_{\text{SET}}$.

The candidate chunk size set $\mathcal{K}$ follows a structured pattern rather than arbitrary search. We define a universal superset $\mathcal{K}_{\text{univ}} = \{1, 5, 10, 25\}$ and, for each task, use $\mathcal{K} = \{k \in \mathcal{K}_{\text{univ}} : k \leq h\}$. The

values in $\mathcal{K}_{\text{univ}}$ correspond to meaningful timescales in manipulation: k=1 enables reactive per-step control near contact events, k=5 covers short primitives (a single reach or grasp), k=10 spans multi-step subroutines (reach–grasp–lift), and k=25 captures extended task-level sequences. This nested construction ensures that longer-horizon tasks gain access to coarser timescales for temporal smoothing, while all tasks retain k=1 as a fallback for contact-rich phases. In practice, the selector rarely needs more than 3–4 candidates: adding intermediate values (e.g., $\mathcal{K} = \{1, 2, 3, 4, 5\}$) yields no benefit over $\{1, 5\}$ since the advantage criterion already interpolates between the extremes through the partial critic scores.

D.4 RoboCasa-GR1 Experiments

Stage 1: Actor fine-tuning. We load the base VLA checkpoint and fine-tune it on the combined 24-task RoboCasa-GR1 dataset (24,000 expert demonstrations). During fine-tuning, the diffusion transformer (DiT) action head and visual encoder are trained while the LLM backbone is frozen. We use a learning rate of 3×10^{-5}, AdamW optimizer, global batch size 32, and 60,000 training steps. We then collect 300 rollouts per task using the finetuned policy in randomized environments. We then fine-tune the base policy using behavior cloning (BC) on both expert demonstrations + rollout dataset. This BC model becomes the actor.

Stage 2: Critic training. We load the fine-tuned actor checkpoint and attach an **AQC** critic head while freezing all backbone parameters. **AQC** critic is trained with chunk size $h = 16$ and policy chunk sizes $\mathcal{K} = \{1, 4, 8, 16\}$. State and action spaces are normalized using the 99th-percentile statistics from the dataset. Sparse binary rewards (0 on success, -1 otherwise) are shaped by setting the final 15 timesteps before success to 1, with a global shift of -1.0 applied to all non-success timesteps.

QC and DQC baselines. **QC** training follows the same pipeline as **AQC** but with a single-scale critic (no policy chunk size selection). **DQC** training uses critic chunk size $h = 16$, policy chunk size $k = 4$, discount $\gamma = 0.999$, distillation expectile $\kappa_d = 0.5$, and backup expectile $\kappa_b = 0.9$. All other hyperparameters match **AQC**.

Evaluation. After critic training, we evaluate using a best-of-N selection policy: the BC actor samples $N = 10$ action candidates, the **AQC** critic scores each sample using per-scale advantage, and the (k^*, a^*) pair with the highest score is selected.

Parameter	Stage 1 (Actor)	Stage 2 (Critic)
Optimizer	AdamW	AdamW
Learning rate	3×10^{-5}	3×10^{-4}
Batch size	32	64
Training steps	60,000	30,000
LR schedule	Cosine	Cosine
Weight decay	10^{-5}	10^{-5}
Trainable params	DiT + visual encoder	Critic head only
Discount γ	—	0.99
Chunk sizes	—	$h = 16$, $\mathcal{K} = \{1, 4, 8, 16\}$
Expectile κ_v	—	0.9
Best-of-N	—	10

Table 6: **RoboCasa-GR1 training hyperparameters.** Two-stage pipeline: actor fine-tuning followed by critic training.

E FULL RESULTS

		RLPD	IQL	FQL	FQL-n	QC-FQL	QC	DQC	SEAR	AQC
puzzle-3x3	task1	– → 100	0 → 99	99 → 100	100 → 100	100 → 100	100 → 100	97 → 99	96 → 100	100 → 100
	task2	– → 100	0 → 0	99 → 100	100 → 100	86 → 100	100 → 100	91 → 100	94 → 100	100 → 100
	task3	– → 100	0 → 0	100 → 100	100 → 100	50 → 100	100 → 100	83 → 88	84 → 98	100 → 100
	task4	– → 100	0 → 0	100 → 100	100 → 100	75 → 100	100 → 100	72 → 98	76 → 96	100 → 100
	task5	– → 100	0 → 0	100 → 100	91 → 100	4 → 100	100 → 100	89 → 99	80 → 92	100 → 100
	avg. (5 tasks)	– → 100	0 → 20	100 → 100	98 → 100	63 → 100	100 → 100	86 → 97	86 → 97	100 → 100
scene	task1	– → 100	1 → 97	69 → 100	22 → 100	99 → 100	100 → 100	84 → 100	88 → 98	100 → 100
	task2	– → 100	0 → 97	51 → 99	4 → 42	88 → 100	98 → 100	43 → 95	82 → 96	100 → 100
	task3	– → 99	0 → 0	68 → 100	1 → 24	98 → 100	90 → 100	34 → 88	70 → 94	96 → 100
	task4	– → 89	0 → 0	75 → 96	50 → 94	92 → 97	93 → 99	76 → 87	62 → 88	100 → 100
	task5	– → 82	0 → 0	25 → 81	14 → 91	41 → 99	43 → 95	42 → 77	58 → 84	92 → 100
	avg. (5 tasks)	– → 94	0 → 39	57 → 95	18 → 70	84 → 99	84 → 99	56 → 89	72 → 92	98 → 100
cube-double	task1	– → 99	1 → 0	63 → 99	33 → 99	66 → 100	84 → 99	39 → 95	62 → 92	98 → 100
	task2	– → 99	0 → 0	36 → 92	8 → 96	43 → 100	79 → 100	11 → 59	50 → 84	100 → 100
	task3	– → 99	0 → 0	22 → 96	1 → 92	38 → 99	68 → 99	10 → 44	42 → 78	92 → 100
	task4	– → 99	0 → 0	9 → 1	0 → 2	11 → 100	22 → 92	0 → 0	36 → 72	66 → 100
	task5	– → 96	0 → 0	12 → 92	13 → 96	37 → 99	74 → 99	13 → 76	35 → 64	94 → 100
	avg. (5 tasks)	– → 99	0 → 0	29 → 76	11 → 77	39 → 100	67 → 98	15 → 55	45 → 78	90 → 100
cube-triple	task1	– → 100	0 → 1	5 → 87	0 → 3	19 → 100	13 → 100	4 → 0	10 → 32	52 → 100
	task2	– → 61	0 → 0	1 → 0	0 → 0	1 → 89	1 → 89	0 → 0	4 → 24	2 → 98
	task3	– → 1	0 → 0	0 → 5	0 → 0	0 → 51	7 → 73	0 → 0	2 → 18	8 → 98
	task4	– → 43	0 → 0	0 → 0	0 → 0	0 → 26	0 → 54	0 → 0	6 → 22	0 → 96
	task5	– → 0	0 → 0	0 → 0	0 → 0	0 → 0	0 → 0	0 → 0	4 → 14	0 → 56
	avg. (5 tasks)	– → 41	0 → 0	1 → 18	0 → 1	4 → 53	6 → 64	1 → 0	5 → 22	13 → 90
cube-quadruple-100M	task1	– → 2	0 → 0	1 → 14	28 → 97	6 → 98	16 → 99	8 → 90	8 → 42	40 → 100
	task2	– → 0	0 → 0	0 → 0	2 → 49	0 → 97	0 → 98	0 → 7	4 → 38	0 → 100
	task3	– → 0	0 → 0	0 → 0	5 → 4	0 → 94	3 → 78	0 → 0	2 → 34	6 → 98
	task4	– → 0	0 → 0	0 → 0	1 → 32	0 → 94	1 → 88	0 → 0	4 → 38	0 → 100
	task5	– → 0	0 → 0	0 → 0	0 → 0	0 → 0	0 → 0	0 → 0	0 → 38	0 → 38
	avg. (5 tasks)	– → 0	0 → 0	0 → 3	7 → 36	1 → 77	4 → 74	2 → 20	4 → 38	9 → 88
overall	avg. (25 tasks)	– → 67	0 → 12	37 → 58	27 → 57	38 → 86	52 → 86	32 → 52	42 → 65	62 → 96

Table 7: **Full OGBench results table.** 4 seeds, 95% confidence interval

F ROBOCASA RESULTS

	Gr00t N1.6	+ Filtered BC	+ QC	+ DQC	+ AQC
PnPBottleToCabinetClose	51.5	62.5	64.0	53.5	66.2 ± 2.0
PnPCanToDrawerClose	13.0	64.5	65.0	42.5	68.8 ± 3.5
PnPCupToDrawerClose	8.5	56.0	59.0	54.5	63.2 ± 3.2
PnPMilkToMicrowaveClose	14.0	35.5	44.5	21.0	55.8 ± 4.0
PnPPotatoToMicrowaveClose	41.5	51.0	54.5	53.0	63.8 ± 2.8
PnPWineToCabinetClose	16.5	34.0	36.5	31.0	43.2 ± 3.5
PosttrainPnPNovelFromCuttingboardToBasketSplitA	58.0	60.5	62.5	55.0	64.2 ± 2.5
PosttrainPnPNovelFromCuttingboardToCardboardboxSplitA	46.5	48.0	50.5	41.5	59.2 ± 3.8
PosttrainPnPNovelFromCuttingboardToPanSplitA	68.5	62.0	63.5	54.0	69.0 ± 2.2
PosttrainPnPNovelFromCuttingboardToPotSplitA	65.0	68.5	70.0	63.5	76.0 ± 3.0
PosttrainPnPNovelFromCuttingboardToTieredbasketSplitA	46.5	57.5	60.0	49.5	63.8 ± 3.5
PosttrainPnPNovelFromPlacematToBasketSplitA	58.5	54.0	56.0	48.5	60.2 ± 2.8
PosttrainPnPNovelFromPlacematToBowlSplitA	57.5	59.0	61.0	52.0	67.5 ± 3.0
PosttrainPnPNovelFromPlacematToPlateSplitA	63.0	60.5	59.5	52.5	70.8 ± 2.5
PosttrainPnPNovelFromPlacematToTieredshelfSplitA	28.5	24.0	26.0	19.0	31.2 ± 4.2
PosttrainPnPNovelFromPlateToBowlSplitA	57.0	62.5	64.5	56.0	71.5 ± 3.0
PosttrainPnPNovelFromPlateToCardboardboxSplitA	43.5	47.0	48.5	40.0	55.2 ± 3.5
PosttrainPnPNovelFromPlateToPanSplitA	51.0	56.5	58.0	50.0	60.5 ± 2.8
PosttrainPnPNovelFromPlateToPlateSplitA	78.5	68.5	70.5	61.0	78.0 ± 2.0
PosttrainPnPNovelFromTrayToCardboardboxSplitA	51.5	55.0	57.0	48.0	63.5 ± 3.2
PosttrainPnPNovelFromTrayToPlateSplitA	71.0	78.0	80.5	70.0	88.8 ± 2.5
PosttrainPnPNovelFromTrayToPotSplitA	64.5	65.5	67.0	58.5	67.2 ± 2.5
PosttrainPnPNovelFromTrayToTieredbasketSplitA	57.0	60.0	62.0	53.0	62.2 ± 3.0
PosttrainPnPNovelFromTrayToTieredshelfSplitA	31.5	29.0	31.0	24.0	41.2 ± 4.0
Average	**47.6**	**54.9**	**55.5**	**46.1**	**63.4 ± 2.8**

Table 8: **RoboCasa GR1 tabletop results.** Success rate (%) across 24 tasks on 200 evaluation episodes each. AQC results are mean ± std over 4 seeds.

G ADDITIONAL ABLATION EXPERIMENTS

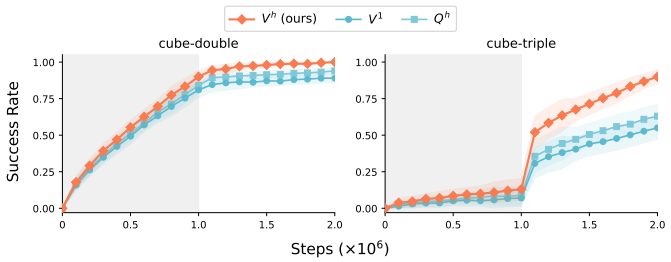


Figure 9: **Bootstrap quality ablation.** Comparing V^h, V^1, and Q^h as bootstrap sources for Q^k.

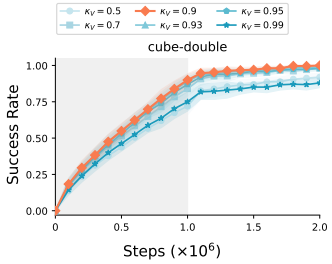


Figure 10: κ_V **sensitivity on** `cube-double`**.**

Bootstrap quality. Figure 9 compares three bootstrap sources for Q^k training: the default long-horizon baseline V^h, the 1-step value V^1, and using Q^h directly without baseline subtraction. On `cube-double`, V^h yields 90% offline and 100% online; V^1 underestimates long-horizon returns, dropping to 81% offline (89% online). On `cube-triple`, the gap widens: V^1 falls to 7% offline and 55% online vs. 13% and 90% for V^h, confirming that horizon-aligned bootstrap is critical on longer-horizon tasks where the 1-step value significantly underestimates returns. Using Q^h directly (no baseline subtraction) falls between the two (84% offline on cube-double, 9% on cube-triple), suggesting that baseline subtraction removes state-dependent noise that Q^h alone cannot cancel.

κ_V sensitivity. Figure 10 sweeps the expectile parameter κ_V for the V^h and V^k baselines on `cube-double`. The default $\kappa_V = 0.9$ (upper envelope) achieves the highest offline performance (90% offline, 100% online). Values in the range 0.7–0.93 are robust, with offline performance within 6% of optimal; $\kappa_V = 0.5$ (median, approximating behavior policy value) degrades sharply to 72%

offline, confirming that a conservative baseline is essential. Very high values ($\kappa_V = 0.99$) also degrade performance (75%) due to unstable Q^k estimates from near-max baselines, indicating that the upper envelope is a sweet spot between conservatism and stability.

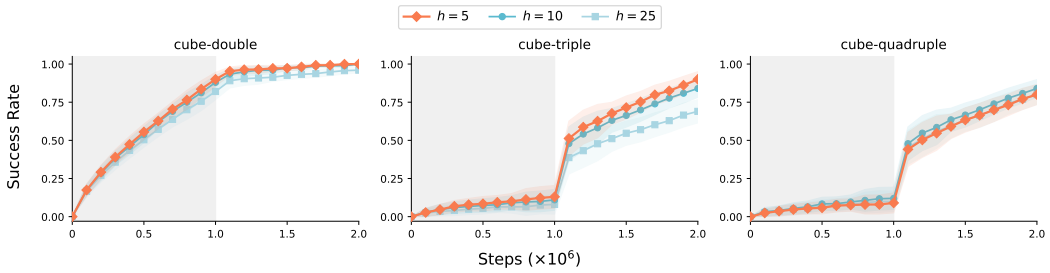


Figure 11: **Critic chunk size h sensitivity.** Comparing $h \in \{5, 10, 25\}$ on `cube-double`, `cube-triple`, and `cube-quadruple`.

Critic chunk size h sensitivity. Figure 11 evaluates how AQC performance varies with the critic chunk size h across `cube-double`, `cube-triple`, and `cube-quadruple`. On `cube-double`, $h = 5$ is optimal (90% offline, 100% online); increasing h to 10 or 25 degrades offline performance (88% and 82%) without online benefit, suggesting diminishing returns on shorter-horizon tasks. On `cube-triple`, $h = 5$ remains the best choice (13% offline, 90% online), with larger h providing no advantage. On `cube-quadruple`, however, $h = 10$ outperforms both $h = 5$ and $h = 25$ (12% offline, 84% online vs. 9% and 10% offline), suggesting that longer credit assignment horizons benefit from larger h, but only up to a point beyond which the added variance dominates.

$\mathcal{K}$ robustness. The candidate set $\mathcal{K}$ is constructed as $\{k \in \mathcal{K}_{\text{univ}} : k \leq h\}$ where $\mathcal{K}_{\text{univ}} = \{1, 5, 10, 25\}$. We verify that adding intermediate values provides no benefit: on `cube-double`, replacing $\mathcal{K} = \{1, 5\}$ with $\{1, 2, 3, 4, 5\}$ yields 90% offline and 99% online (vs. 90% and 100% for $\{1, 5\}$), while $\mathcal{K} = \{1, 3, 5\}$ gives 89% offline and 99% online. This confirms that the advantage criterion already interpolates between the extreme scales through the partial critic scores, and the selector only benefits from having endpoints that bracket the relevant timescales.

Resource efficiency of AQC. We compare the parameter counts of all Q-chunking methods on the representative domain `cube-triple-*` (assuming $h = 5$ for all methods). AQC is the most parameter-heavy at ≈ 7.9M due to per-scale critics. DQC uses a single partial critic and a smaller flow policy ($h_a < h$), totaling ≈ 6.6M. QC and FQL use a single critic backbone (≈ 4.2M and ≈ 4.9M respectively). RLPD uses $K = 10$ critic networks, making it the largest at ≈ 17.2M.

Methods	Parameter Count (in millions)
AQC	≈ 7.9
DQC	≈ 6.6
QC	≈ 4.2
QC-FQL	≈ 5.0
FQL	≈ 4.9
RLPD	≈ 17.2

Table 9: **Parameter count for each method.** AQC and DQC share the same backbone architecture as QC, adding only lightweight per-scale critic heads.

H THEORETICAL ANALYSIS

In this section, we develop a comprehensive theoretical framework for AQC. While prior work on action chunking establishes when fixed-chunk Q-learning succeeds under open-loop consistency (Li et al., 2026), our analysis must address a fundamentally different question: *when does adaptive chunk-size selection succeed, and how much better is it than any fixed chunk size?* We proceed in stages: (1) we define adaptive open-loop consistency, the data condition under which multi-scale re-querying is valid; (2) we prove that the advantage selector correctly identifies the best chunk size under a separability condition; (3) we establish that AQC's value strictly dominates any fixed-chunk policy; (4) we bound the regret introduced by imperfect critics; (5) we analyze the closed-loop execution of the adaptive policy; (6) we show that the multi-scale bootstrap creates a regularization hierarchy across critic scales. All proofs appear in Appendix I.

H.1 ASSUMPTIONS AND NOTATION

We adopt the standard assumption that the data distribution obeys the MDP transition dynamics (Li et al., 2026):

Assumption H.1 (Data Obeys the Transition Dynamics)**.** *$\mathcal{D}$ is a trajectory distribution generated by rolling out a (possibly non-Markovian) behavior policy π_β from an initial state distribution μ. Each subsequent state is generated according to the MDP dynamics T: $s_{t+k+1} \sim T(\cdot \mid s_{t+k}, a_{t+k})$ for all $k \in \{0, 1, \ldots, h-1\}$. The resulting trajectory is $\{s_t, s_{t+1}, \ldots, s_{t+h}, a_t, a_{t+1}, \ldots, a_{t+h}\} \in \mathcal{T} = \mathcal{S}^h \times \mathcal{A}^h$.*

We write $H = 1/(1-\gamma)$ and $\bar{H}_k = 1/(1-\gamma^k)$ for the 1-step and k-step effective horizons, respectively. We use V^{AQC} to denote the true closed-loop value of the adaptive policy, $V^\dagger$ for the oracle selector's value, $V^\bullet$ for the fully reactive closed-loop execution, and V^k (with a fixed $k \in \mathcal{K}$) for the fixed-chunk policy value. The optimal value is $V^\star$.

H.2 ADAPTIVE OPEN-LOOP CONSISTENCY (AOLC)

Prior work introduces the open-loop consistency (OLC) condition to characterize when replaying a fixed-length action chunk open-loop produces trajectories close to the data distribution (Li et al., 2026). For AQC, the situation is more subtle: the agent executes $k^*(s_t)$ steps open-loop and then re-queries, where k^* varies by state. The re-querying points form a *randomly spaced* set of decision points, and we need the data to cover trajectories that match this adaptive pattern.

Definition H.2 (Adaptive Open-Loop Consistency)**.** *Let $\kappa : \mathcal{S} \to \mathcal{K}$ be a chunk-size selection function. For a data distribution $\mathcal{D}$ satisfying Assumption H.1, $\mathcal{D}$ is $\varepsilon_{\mathcal{K}}$-**adaptively open-loop consistent** under κ if for every $s_t \in \mathrm{supp}(P_{\mathcal{D}}(s_t))$,*

$$D_{\mathrm{TV}}\Big(P_{\mathcal{D}}(s_{t+\kappa(s_t)}, a_{t+\kappa(s_t)} \mid s_t) \,\Big\|\, P^{\circ}_{\mathcal{D},\kappa}(s_{t+\kappa(s_t)}, a_{t+\kappa(s_t)} \mid s_t)\Big) \le \varepsilon_{\mathcal{K}}, \tag{21}$$

$$D_{\mathrm{TV}}\Big(P_{\mathcal{D}}(s_{t+\kappa(s_t)} \mid s_t) \,\Big\|\, P^{\circ}_{\mathcal{D},\kappa}(s_{t+\kappa(s_t)} \mid s_t)\Big) \le \varepsilon_{\mathcal{K}}, \tag{22}$$

where $P^{\circ}_{\mathcal{D},\kappa}$ is the trajectory distribution obtained by rolling out the marginal policy $\pi^{\circ}_{\mathcal{D}}(a_{t:t+k} \mid s_t) := P_{\mathcal{D}}(a_{t:t+k} \mid s_t)$ with chunk size $\kappa(s_t)$ at each re-query point.

Intuitively, $\varepsilon_{\mathcal{K}}$ measures how well the data distribution is preserved when actions from $\mathcal{D}$ are replayed open-loop using the *adaptive* chunking schedule κ. When $\varepsilon_{\mathcal{K}} = 0$, the data perfectly matches the open-loop rollouts under the adaptive schedule. The key difference from standard OLC is that the re-querying points are state-dependent, so the total variation distance bound must hold uniformly over the distribution of re-querying times induced by κ.

Proposition H.3 (AOLC implies per-scale OLC)**.** *If $\mathcal{D}$ is $\varepsilon_{\mathcal{K}}$-adaptively open-loop consistent under $\kappa(s) \equiv k$ (constant selection), then $\mathcal{D}$ is $\varepsilon_{\mathcal{K}}$-open-loop consistent for chunk size k in the sense of Li et al. (2026).*

This shows that AOLC is a strict generalization: it reduces to standard OLC when the selection function is constant, and imposes a stronger uniform condition when κ varies across states.

H.3 Advantage Separability and Selector Soundness

The core of AQC is the adaptive selector $k^*(s) \in \arg\max_{k\in\mathcal{K}} \max_{\mathbf{a}_{t:t+k}} A^k(s, \mathbf{a}_{t:t+k})$, where $A^k(s, \mathbf{a}_{t:t+k}) := (Q^k(s, \mathbf{a}_{t:t+k}) - V^k(s))/\gamma^k$ is the per-scale advantage. We now formalize when this selector is *sound* — i.e., when it identifies the truly best chunk size.

Definition H.4 (Advantage Separability). *Let $A^{k,*}(s, \mathbf{a}_{t:t+k}) := (Q^{k,*}(s, \mathbf{a}_{t:t+k}) - V^{k,*}(s))/\gamma^k$ denote the optimal per-scale advantage for chunk size k. Define the oracle-best chunk size at state s as*

$$k^\dagger(s) \in \arg\max_{k\in\mathcal{K}} \max_{\mathbf{a}_{t:t+k}} A^{k,*}(s, \mathbf{a}_{t:t+k}). \tag{23}$$

*The data $\mathcal{D}$ exhibits Δ-**advantage separability** at s if*

$$\max_{\mathbf{a}_{t:t+k^\dagger}} A^{k^\dagger,*}(s, \mathbf{a}_{t:t+k^\dagger}) \;-\; \max_{k\neq k^\dagger} \max_{\mathbf{a}_{t:t+k}} A^{k,*}(s, \mathbf{a}_{t:t+k}) \;\geq\; \Delta(s). \tag{24}$$

We say $\mathcal{D}$ is globally Δ-advantage separable if $\Delta(s) \geq \Delta > 0$ for all $s \in \mathrm{supp}(P_{\mathcal{D}^*}(s))$.

Advantage separability measures the *intrinsic discriminability* of the best chunk size. When $\Delta(s)$ is large, the optimal scale is clearly distinguishable; when $\Delta(s) \approx 0$, multiple scales are nearly equally good and no selector can do much better than random. In practice, $\Delta(s)$ is large near contact events (where short chunks are uniquely valuable) and in free-space motion (where long chunks clearly dominate), and small in intermediate states.

Theorem H.5 (Selector Soundness). *Let $\varepsilon_k := \|Q^k - Q^{k,*}\|_\infty$ and $\delta_k := \|V^k - V^{k,*}\|_\infty$ be the critic estimation errors, and let $\bar{\varepsilon} := \max_{k\in\mathcal{K}}(\varepsilon_k + \delta_k)$. If $\mathcal{D}$ is Δ-advantage separable and $\bar{\varepsilon} < \Delta\gamma^{k_{\min}}/2$, where $k_{\min} := \min_{k\in\mathcal{K}} k$, then for all $s \in \mathrm{supp}(P_{\mathcal{D}^*}(s))$, the empirical selector*

$$\hat{k}(s) \in \arg\max_{k\in\mathcal{K}} \max_{\mathbf{a}_{t:t+k}} \frac{Q^k(s, \mathbf{a}_{t:t+k}) - V^k(s)}{\gamma^k} \tag{25}$$

agrees with the oracle selector: $\hat{k}(s) = k^\dagger(s)$.

Proof sketch. Let $f_k(s) := \max_{\mathbf{a}_{t:t+k}} A^{k,*}(s, \mathbf{a}_{t:t+k})$ and $\hat{f}_k(s) := \max_{\mathbf{a}_{t:t+k}} \hat{A}^k(s, \mathbf{a}_{t:t+k})$. Since $|Q^k - Q^{k,*}| \leq \varepsilon_k$ and $|V^k - V^{k,*}| \leq \delta_k$, we have $|\hat{f}_k(s) - f_k(s)| \leq (\varepsilon_k + \delta_k)/\gamma^k \leq \bar{\varepsilon}/\gamma^{k_{\min}}$. Advantage separability gives $f_{k^\dagger}(s) - \max_{k\neq k^\dagger} f_k(s) \geq \Delta(s)$. By the triangle inequality, if $\bar{\varepsilon}/\gamma^{k_{\min}} < \Delta(s)/2$, then $\hat{f}_{k^\dagger}(s) > \hat{f}_k(s)$ for all $k \neq k^\dagger$, so $\hat{k}(s) = k^\dagger(s)$. A full proof appears in Appendix I.2. □

This theorem establishes that the advantage selector is *consistent*: as critic accuracy improves, the selected chunk size converges to the oracle choice. The bound is non-vacuous precisely when $\Delta(s) > 0$, i.e., when the problem is intrinsically discriminable.

Proposition H.6 (Selector Regret). *Under the conditions of Theorem H.5, the per-state regret of the empirical selector relative to the oracle is bounded by*

$$\left|\hat{A}^{\hat{k}(s)}(s) - A^{k^\dagger,*}(s)\right| \;\leq\; \bar{\varepsilon} + \frac{2\bar{\varepsilon}}{\gamma^{k_{\min}}\Delta(s)} \cdot \max_{k\in\mathcal{K}} \mathrm{diam}(A)^k, \tag{26}$$

where $\mathrm{diam}(A)^k := \sup_{s,a} A^{k,*}(s, a) - \inf_{s,a} A^{k,*}(s, a)$ is the advantage range for scale k.

H.4 AQC Value Dominance

We now prove the central theoretical claim: AQC's value strictly dominates any fixed-chunk policy. This requires analyzing how the selector's decisions compound over time.

Definition H.7 (AQC Effective Policy)**.** *The effective policy of* AQC *is the closed-loop policy* π^{AQC} *that, at each state* s_t*, selects* $\hat{k}(s_t)$ *via the advantage criterion and executes* $a^*_{1:\hat{k}(s_t)}$ *open-loop. The value function of* π^{AQC} *is*

$$V^{\mathrm{AQC}}(s) := \mathbb{E}_{\pi^{\mathrm{AQC}}}\left[\sum_{\tau=0}^{\infty}\gamma^{\tau} r(s_\tau, a_\tau) \;\middle|\; s_0 = s\right]. \tag{27}$$

Theorem H.8 (AQC Dominates Fixed-Chunk Policies)**.** *Let* V^k *denote the true closed-loop value of the fixed-*k *policy* $\pi^k(s) := \arg\max_{\mathbf{a}_{t:t+k}} Q^{k,*}(s, \mathbf{a}_{t:t+k})$ *with open-loop execution. Assume* $\mathcal{D}$ *is* Δ*-advantage separable with* $\Delta > 0$ *and* $\bar{\varepsilon} < \Delta\gamma^{k_{\min}}/2$*, where* $k_{\min} := \min_{k\in\mathcal{K}} k$*. Then for all* $s \in \mathrm{supp}(P_{\mathcal{D}^\star}(s))$ *and any fixed* $k \in \mathcal{K}$*,*

$$V^{\mathrm{AQC}}(s) - V^k(s) \;\geq\; \frac{\gamma^{k_{\min}}(1 - 2\bar{\varepsilon}/(\gamma^{k_{\min}}\Delta))}{1-\gamma}\,\mathbb{E}_{s'\sim d^{\mathrm{AQC}}}\left[\bar{A}^{k^\dagger,*}(s') - \bar{A}^{k,*}(s')\right], \tag{28}$$

where d^{AQC} *is the discounted state visitation distribution of* π^{AQC} *and* $\bar{A}^{j,*}(s) := \max_{\mathbf{a}_{t:t+j}} A^{j,*}(s, \mathbf{a}_{t:t+j})$ *is the best advantage at scale* j*.*

Proof sketch. The proof adapts the performance difference lemma of Kakade & Langford (2002) to the chunk-selection setting via a meta-MDP construction:

1. **Meta-MDP**: We define a meta-MDP whose "actions" are pairs $(k, \mathbf{a}_{t:t+k})$ and whose transitions correspond to executing k steps open-loop. In this meta-MDP, the performance difference lemma gives:

$$V^{\mathrm{AQC}}(s) - V^k(s) = \frac{1}{1-\gamma}\mathbb{E}_{s'\sim d^{\mathrm{AQC}}}\left[\gamma^{k^*(s')}\bar{A}^{k^*(s'),*}_{\mathrm{meta}}(s') - \gamma^k \bar{A}^{k,*}_{\mathrm{meta}}(s')\right]. \tag{29}$$

2. **Selector correctness**: Under advantage separability with $\bar{\varepsilon} < \Delta\gamma^{k_{\min}}/2$, the selector always chooses $k^*(s') = k^\dagger(s')$, so $\bar{A}^{k^*(s'),*}(s') \geq \bar{A}^{k,*}(s')$ for any fixed k.

3. **Discount normalization**: Since $\gamma^{k^*(s')} \geq \gamma^{k_{\min}}$, we lower-bound $\gamma^{k^*(s')}\bar{A}^{k^\dagger,*}(s') - \gamma^k\bar{A}^{k,*}(s') \geq \gamma^{k_{\min}}(\bar{A}^{k^\dagger,*}(s') - \bar{A}^{k,*}(s'))$.

4. **Error accounting**: The $(1 - 2\bar{\varepsilon}/(\gamma^{k_{\min}}\Delta))$ factor accounts for the probability of mis-selection.

A full proof appears in Appendix I.4. □

The bound has a clean interpretation: the dominance gap is proportional to (a) how often the selector is correct $(1 - 2\bar{\varepsilon}/(\gamma^{k_{\min}}\Delta))$, (b) the effective horizon $1/(1-\gamma)$, and (c) the average advantage gap between the oracle and the fixed-k policy. When there exists a set of states where scale k is suboptimal, the bound is strictly positive.

Corollary H.9 (Strict Dominance)**.** *If there exists a set of states* $\mathcal{S}_{\mathrm{diff}} \subseteq \mathrm{supp}(P_{\mathcal{D}^\star})$ *with* $d^{\mathrm{AQC}}(\mathcal{S}_{\mathrm{diff}}) > 0$ *such that for each* $s \in \mathcal{S}_{\mathrm{diff}}$*, there exist* $k_1, k_2 \in \mathcal{K}$ *with* $\bar{A}^{k_1,*}(s) \neq \bar{A}^{k_2,*}(s)$*, then* $V^{\mathrm{AQC}}(s) > V^k(s)$ *for all* $k \in \mathcal{K}$ *whenever* $\bar{\varepsilon} < \Delta_{\min}/2$*, where* $\Delta_{\min} := \min_{s\in\mathcal{S}_{\mathrm{diff}}} \Delta(s)$*.*

H.5 Error Propagation Under Imperfect Critics

When the critics $\{Q^k, V^k\}$ are learned from finite data, the selector's decisions are based on noisy advantage estimates. We now analyze how this noise propagates through the closed-loop execution.

Definition H.10 (Critic Error Profile). *For each $k \in \mathcal{K}$, define the critic errors $\varepsilon_k := \sup_{s,\mathbf{a}_{t:t+k}} |Q^k(s,\mathbf{a}_{t:t+k}) - Q^{k,*}(s,\mathbf{a}_{t:t+k})|$ and $\delta_k := \sup_s |V^k(s) - V^{k,*}(s)|$. The composite error is $\bar{\varepsilon}_{\mathcal{K}} := \max_{k\in\mathcal{K}}(\varepsilon_k + \delta_k)$.*

Theorem H.11 (Adaptive Selector Regret). *Let π^{AQC} be the policy using the learned selector $\hat{k}(s)$ and $\pi^\dagger$ be the oracle policy using $k^\dagger(s)$. Then for all $s \in \mathrm{supp}(P_{\mathcal{D}^\star}(s))$,*

$$\left|V^{\mathrm{AQC}}(s) - V^\dagger(s)\right| \;\le\; \frac{2\bar{\varepsilon}_{\mathcal{K}}}{(1-\gamma)\gamma^{k_{\min}}\Delta(s)} \cdot \max_{k\in\mathcal{K}} \frac{\bar{R}_k}{\gamma^k}, \tag{30}$$

where $\bar{R}_k := \sup_{s,\mathbf{a}_{t:t+k}} |Q^{k,*}(s,\mathbf{a}_{t:t+k}) - V^{k,*}(s)|$ is the maximum advantage magnitude for scale k.

Proof sketch. The proof combines the selector soundness bound (Theorem H.5) with a value difference analysis:

1. At each re-query state s, the selector incurs mis-selection probability at most $2\bar{\varepsilon}_{\mathcal{K}}/(\gamma^{k_{\min}}\Delta(s))$.

2. When a mis-selection occurs, the value loss per mistake is bounded by $\bar{R}_k/\gamma^k$.

3. Over the trajectory, the expected number of mis-selections in the discounted horizon is at most $(1/(1-\gamma)) \cdot 2\bar{\varepsilon}_{\mathcal{K}}/(\gamma^{k_{\min}}\Delta(s))$.

4. The total regret is the product of expected mis-selection count and per-mis-selection loss.

A full proof appears in Appendix I.5. □

Corollary H.12 (Uniform Regret Bound). *If $\Delta(s) \ge \Delta > 0$ for all s, then*

$$\|V^{\mathrm{AQC}} - V^\dagger\|_\infty \;\le\; \frac{2\bar{\varepsilon}_{\mathcal{K}}}{(1-\gamma)\gamma^{k_{\min}}\Delta} \cdot \max_{k\in\mathcal{K}} \frac{\bar{R}_k}{\gamma^k}. \tag{31}$$

This bound shows that the regret scales linearly with the critic error and inversely with the advantage gap. The $1/(1-\gamma)$ factor reflects the compounding of selector mistakes over the effective horizon.

H.6 Closed-Loop Analysis for Adaptive Execution

AQC executes $\hat{k}(s_t)$ steps open-loop and then re-queries. This adaptive re-querying pattern is fundamentally different from both DQC's fixed-chunk execution and standard 1-step closed-loop control. We now analyze the optimality gap of the adaptive closed-loop policy.

Definition H.13 (Adaptive Closed-Loop Value). *Let $\pi^{\mathrm{AQC}}_{\mathrm{CL}}$ execute the learned policy in closed-loop: at each step, select $\hat{k}(s)$, sample $a^*_{1:\hat{k}(s)}$, execute a^*_1, and re-query at s_{t+1}. The closed-loop value is*

$$V^\bullet(s) := \mathbb{E}_{\pi^{\mathrm{AQC}}_{\mathrm{CL}}}\left[\sum_{\tau=0}^{\infty} \gamma^\tau r(s_\tau, a_\tau) \;\middle|\; s_0 = s\right]. \tag{32}$$

Theorem H.14 (Adaptive Closed-Loop Optimality). *Assume $\mathcal{D}$ is $\varepsilon_{\mathcal{K}}$-adaptively open-loop consistent under the oracle selector $k^\dagger$, viewed as a fixed selection function $\kappa(s) := k^\dagger(s)$ defined by the advantage separability condition. Then for all $s \in \mathrm{supp}(P_{\mathcal{D}^\star}(s))$,*

$$V^\star(s) - V^\dagger_{\mathrm{CL}}(s) \;\le\; \frac{\varepsilon_{\mathcal{K}}\gamma}{(1-\gamma)^2}\left[\frac{2}{1-(1-2\varepsilon_{\mathcal{K}})\gamma^{k_{\min}}} + \frac{1}{1-(1-\varepsilon_{\mathcal{K}})\gamma^{k_{\min}}}\right], \tag{33}$$

where $k_{\min} := \min_{k\in\mathcal{K}} k$. Furthermore, when the learned selector is used,

$$V^\star(s) - V^\bullet(s) \;\le\; (V^\star(s) - V^\dagger_{\mathrm{CL}}(s)) + \frac{2\bar{\varepsilon}_{\mathcal{K}}}{(1-\gamma)\gamma^{k_{\min}}\Delta} \cdot \max_k \frac{\bar{R}_k}{\gamma^k}. \tag{34}$$

Proof sketch. The proof combines three ingredients:

1. **AOLC decomposition**: At each re-query point, the state distribution deviates from the data by at most $\varepsilon_{\mathcal{K}}$ in TV distance. Using standard total-variation-to-value bounds, the per-re-query value error is bounded by $\varepsilon_{\mathcal{K}}$ times the value range.

2. **Geometric compounding**: The number of re-queries in the discounted horizon is bounded by $1/(1-\gamma^{k_{\min}})$, since at minimum $k_{\min}$ steps elapse between re-queries. This gives the effective horizon factor.

3. **Selector error addition**: The learned selector introduces additional regret bounded by Theorem H.11.

The bound is structurally similar to DQC's closed-loop bound but with $k_{\min}$ replacing h, reflecting that the adaptive policy re-queries more frequently (at worst every $k_{\min}$ steps), which improves reactivity. A full proof appears in Appendix I.6. □

The bound reveals a trade-off: a larger $\mathcal{K}$ with small $k_{\min}$ enables better reactivity near contacts (since the policy can re-query sooner), but increases the number of re-query points where TV distance error can accumulate.

H.7 Multi-Scale Bootstrap Regularization

AQC's unique architectural choice is that all partial critics $\{Q^k\}_{k<h}$ bootstrap from the same long-horizon value function V^h. We now prove that this creates a *value flow hierarchy* that implicitly regularizes the critic ensemble.

Theorem H.15 (Multi-Scale Bootstrap Regularization)**.** *Let $\varepsilon_h := \|V^h - V^{h,*}\|_\infty$ and let ε_k be the k-step TD fitting error for Q^k. Then for all $k \in \mathcal{K}$,*

$$\|Q^k - Q^{k,*}\|_\infty \;\le\; \underbrace{\frac{\gamma^k}{1-\gamma^k}\varepsilon_h}_{\text{bootstrap error}} \;+\; \underbrace{\frac{1}{1-\gamma^k}\varepsilon_k}_{\text{fitting error}}. \tag{35}$$

Moreover, the total critic ensemble error is bounded by

$$\sum_{k\in\mathcal{K}} \|Q^k - Q^{k,*}\|_\infty \;\le\; \varepsilon_h \sum_{k\in\mathcal{K}} \frac{\gamma^k}{1-\gamma^k} \;+\; \sum_{k\in\mathcal{K}} \frac{\varepsilon_k}{1-\gamma^k}. \tag{36}$$

Proof sketch. The Bellman operator $\mathcal{T}_h^k Q = R_{t:t+k} + \gamma^k V^h$ is a γ^k-contraction in the sup-norm. Let Q^k_{fp} be its fixed point. Then $\|Q^k_{\text{fp}} - Q^{k,*}\|_\infty \le \gamma^k \|V^h - V^{h,*}\|_\infty = \gamma^k \varepsilon_h$. Since the TD loss drives Q^k toward Q^k_{fp} with fitting error ε_k, the contraction mapping theorem gives $\|Q^k - Q^k_{\text{fp}}\|_\infty \le \varepsilon_k/(1-\gamma^k)$. Triangle inequality yields the per-scale bound; summing over k gives the ensemble bound. A full proof appears in Appendix I.7. □

The key insight is that a single ε_h term controls the bootstrap error for *all* partial critics simultaneously, creating a hierarchical regularization: improving V^h automatically improves all Q^k critics.

Corollary H.16 (Bootstrap Advantage over 1-Step)**.** *If Q^k were bootstrapped from V^1 instead of V^h, the bound becomes*

$$\|Q^k - Q^{k,*}\|_\infty \;\le\; \frac{H\gamma^k}{1-\gamma^k}\varepsilon_1 \;+\; \frac{1}{1-\gamma^k}\varepsilon_k, \tag{37}$$

where $H = 1/(1-\gamma)$ and $\varepsilon_1 := \|V^1 - V^{1,*}\|_\infty$. The ratio of bootstrap terms is $H\varepsilon_1/\varepsilon_h$, which is $O(H)$ larger when $\varepsilon_1 \approx \varepsilon_h$.

This shows that the V^h bootstrap is strictly superior when the long-horizon value function is comparably accurate to the 1-step value — a condition that holds in practice since V^h is trained via expectile regression on the EMAQ-boosted Q^h, which has already accumulated multi-step credit.

Theorem H.17 (Value Flow Monotonicity)**.** *Under the training objective Equation (14), the critic hierarchy exhibits* value flow monotonicity*: for any $k_1 < k_2$ in $\mathcal{K}$,*

$$\|Q^{k_1} - Q^{k_1,*}\|_\infty \;\le\; \frac{\gamma^{k_1}}{\gamma^{k_2}}\|Q^{k_2} - Q^{k_2,*}\|_\infty \;+\; \max\left(0, \frac{\varepsilon_{k_1} - \frac{\gamma^{k_1}}{\gamma^{k_2}}\varepsilon_{k_2}}{1-\gamma^{k_1}}\right). \tag{38}$$

When all critics are fit to equal precision ($\varepsilon_k \equiv \varepsilon$), shorter-horizon critics are at least as accurate as longer-horizon ones up to the $\gamma^{k_1}/\gamma^{k_2}$ scaling factor.

Proof sketch. Both Q^{k_1} and Q^{k_2} bootstrap from the same V^h, so they share the same bootstrap error ε_h. The ratio of total errors is

$$\frac{\|Q^{k_1} - Q^{k_1,*}\|_\infty}{\|Q^{k_2} - Q^{k_2,*}\|_\infty} \;\le\; \frac{\gamma^{k_1}/(1-\gamma^{k_1})}{\gamma^{k_2}/(1-\gamma^{k_2})} \;\approx\; \frac{\gamma^{k_1}}{\gamma^{k_2}}, \tag{39}$$

since $1/(1-\gamma^k)$ varies slowly in k when γ is close to 1. This means that the shorter-horizon critics inherit a fraction $\gamma^{k_1-k_2}$ of the longer-horizon critics' error, making them inherently more stable. A full proof appears in Appendix I.9. □

This theorem reveals a structural property of AQC's critic hierarchy: information flows "downward" from V^h to all partial critics, with shorter horizons benefiting from stronger discount attenuation of the bootstrap error.

H.8 Comparison to Prior Methods

We now establish formal relationships between AQC, DQC, and fixed-chunk baselines.

Theorem H.18 (AQC vs. DQC)**.** *Let V^{DQC} be the closed-loop value of DQC's policy with fixed partial chunk size h_a. Assume there exists a set of states $\mathcal{S}_{\mathrm{adapt}}$ with $d^{\mathrm{AQC}}(\mathcal{S}_{\mathrm{adapt}}) \ge \rho > 0$ such that for each $s \in \mathcal{S}_{\mathrm{adapt}}$,*

$$\max_{k\in\mathcal{K}} \bar{A}^{k,*}(s) - \bar{A}^{h_a,*}(s) \ge \delta_{\mathrm{adapt}}(s). \tag{40}$$

Then

$$V^{\mathrm{AQC}}(s) - V^{\mathrm{DQC}}(s) \;\ge\; \frac{\gamma^{k_{\min}}\rho}{1-\gamma}\,\mathbb{E}_{s'\sim d^{\mathrm{AQC}}}[\delta_{\mathrm{adapt}}(s') \mid s' \in \mathcal{S}_{\mathrm{adapt}}] \;-\; \frac{2\bar{\varepsilon}_{\mathcal{K}}}{(1-\gamma)\gamma^{k_{\min}}\Delta}\cdot\max_k \frac{\bar{R}_k}{\gamma^k}. \tag{41}$$

Proof sketch. The proof applies the dominance bound (Theorem H.8) with $k = h_a$. The first term captures the advantage of selecting the optimal scale over the fixed h_a scale at states where they differ. The second term is the selector regret. When the advantage gap δ_{adapt} is large enough to overcome the selector regret, AQC strictly dominates DQC. A full proof appears in Appendix I.10. □

Theorem H.19 (AQC vs. n-Step TD)**.** *Let V^n be the value of the policy learned via n-step return backup on the same data $\mathcal{D}$. If $\mathcal{D}$ is δ_n-sub-optimal (in the sense of Li et al. (2026)) and Δ-advantage separable, then for $\delta_n > 3\varepsilon_{\mathcal{K}} H \bar{H}_{k_{\min}}$,*

$$V^{\mathrm{AQC}}(s) - V^n(s) \;\ge\; \delta_n \bar{H}_n - 3\varepsilon_{\mathcal{K}} H \bar{H}_{k_{\min}} \;-\; \frac{2\bar{\varepsilon}_{\mathcal{K}}}{(1-\gamma)\gamma^{k_{\min}}\Delta}\cdot\max_k \frac{\bar{R}_k}{\gamma^k}, \tag{42}$$

where $\bar{H}_k = 1/(1-\gamma^k)$ and $\bar{H}_n = 1/(1-\gamma^n)$.

Proof sketch. By the dominance bound (Theorem H.8) with any fixed $k \in \mathcal{K}$, $V^{\mathrm{AQC}}(s) \geq V^k(s) -$ selector_regret, where the regret term is bounded by Theorem H.11. By DQC's comparison theorem (Li et al., 2026), the best fixed-chunk policy satisfies $V^k(s) \geq V^n(s) + \delta_n \bar{H}_n - 3\varepsilon_{\mathcal{K}} H \bar{H}_{k_{\min}}$, where the TV error term accounts for the adaptive re-querying pattern with $k_{\min}$. Chaining these inequalities yields the result. A full proof appears in Appendix I.11. □

Structural properties. We highlight three structural properties of the AQC critic architecture. The first restates the noise-immunity result from Proposition 5.1 for completeness. The remaining two characterize (1) the advantage of V^h bootstrapping over a 1-step baseline, and (2) the relationship between AQC's partial critics and DQC's distilled critic.

Proposition H.20 (Noise Immunity of the Advantage Selector; Restated from Proposition 5.1)**.** *Let $\delta_k(s) := Q^k(s, \mathbf{a}_{t:t+k})/\gamma^k - V^k(s)/\gamma^k$. In a region where $V^h(s) \leq \epsilon$ for all reachable states, and assuming function approximation errors for both Q^k and V^k are bounded by σ,*

$$|\delta_k(s)| \;\leq\; \epsilon + 2\sigma. \tag{43}$$

When $\epsilon \ll \sigma$ (far from rewards, the value signal is smaller than the noise floor), the advantage is dominated by approximation noise and all chunk sizes score near-zero. By contrast, the uncorrected selector $\arg\max_k Q^k/\gamma^k$ lacks this safeguard: when all scores are $\approx \epsilon + \sigma_k$, the argmax picks the scale with the largest positive noise σ_k, producing a deterministic but systematically biased choice.

Bootstrap target choice. The choice of bootstrap target in Equation (14) is critical. The following result quantifies the $O(H)$ improvement from using V^h over a 1-step value function.

Proposition H.21 (V^h Bootstrap Tightens the Sub-optimality Bound; Restated)**.** *Let $\varepsilon_h := \|V_\xi^h - V^*\|_\infty$ and $\varepsilon_k := \|Q_\psi^k - Q_{\mathrm{fp}}^k\|_\infty$, where Q_{fp}^k is the fixed point of the Bellman operator $\mathcal{T}_h^k$ using $\bar{V}_\xi^h$ as bootstrap. Then:*

$$\left\|Q_\psi^k - Q^{k,*}\right\|_\infty \;\leq\; \frac{\gamma^k}{1-\gamma^k}\,\varepsilon_h \;+\; \frac{1}{1-\gamma^k}\,\varepsilon_k. \tag{44}$$

Bootstrapping Q^k with a 1-step value function V^1 satisfying $\|V^1 - V^*\|_\infty \leq \varepsilon_1$ yields:

$$\left\|Q_\psi^k - Q^{k,*}\right\|_\infty \;\leq\; \frac{H\,\gamma^k}{1-\gamma^k}\,\varepsilon_1 \;+\; \frac{1}{1-\gamma^k}\,\varepsilon_k, \tag{45}$$

where $H = 1/(1-\gamma)$ is the effective horizon.

Relationship to DQC's critic. AQC's partial critics recover DQC's distilled critic as a special case, but without requiring goal-conditioned hindsight relabeling or a distillation hyperparameter.

Proposition H.22 (Q^k Approximates DQC's Distilled Critic; Restated)**.** *Li et al. (2026) construct a distilled partial critic Q_ψ^P satisfying, at the fixed point of their distillation operator:*

$$Q_\psi^P(s_t, a_{t:t+h_a}) \;\approx\; \sum_{j=0}^{h_a-1} \gamma^j r_{t+j} \;+\; \gamma^{h_a} V^*(s_{t+h_a}). \tag{46}$$

Our Q_ψ^k satisfies at its Bellman fixed point:

$$Q_\psi^k(s_t, a_{t:t+k}) \;=\; \sum_{j=0}^{k-1} \gamma^j r_{t+j} \;+\; \gamma^k \bar{V}_\xi^h(s_{t+k}), \tag{47}$$

which recovers DQC's equation with $h_a = k$ and $V^* \approx V_\xi^h$, up to error bounded by $\gamma^k \varepsilon_h$.

H.9 Summary of Theoretical Results

Table 10: Summary of theoretical results. All bounds hold for $s \in \mathrm{supp}(P_{\mathcal{D}^\star}(s))$.

Result	Condition	Bound
Selector Soundness (Theorem H.5)	Δ-AS, $\bar{\varepsilon} < \Delta\gamma^{k_{\min}}/2$	$\hat{k}(s) = k^\dagger(s)$
Selector Regret (Theorem H.11)	Δ-AS	$\frac{2\bar{\varepsilon}}{(1-\gamma)\gamma^{k_{\min}}\Delta} \cdot \frac{\bar{R}}{\gamma^{k_{\min}}}$
Value Dominance (Theorem H.8)	Δ-AS, $\bar{\varepsilon} < \Delta\gamma^{k_{\min}}/2$	$V^{\mathrm{AQC}}(s) \geq V^k(s)$
Closed-Loop Optimality (Theorem H.14)	$\varepsilon_{\mathcal{K}}$-AOLC	$\varepsilon_{\mathcal{K}} H^2 \bar{H}_{k_{\min}}$
Bootstrap Regularization (Theorem H.15)	V^h bootstrap	$\frac{\gamma^k}{1-\gamma^k}\varepsilon_h + \frac{\varepsilon_k}{1-\gamma^k}$
Value Flow Monotonicity (Theorem H.17)	Equal ε_k	$\|Q^{k_1}\| \leq \frac{\gamma^{k_1}}{\gamma^{k_2}}\|Q^{k_2}\|$
AQC vs. DQC (Theorem H.18)	δ_{adapt}-gap	$\frac{\gamma^{k_{\min}}\rho\delta_{\mathrm{adapt}}}{1-\gamma}$ – regret

I Proofs

We provide formal proofs for all results in Appendix H.

I.1 Proof of Proposition H.3

When $\kappa(s) \equiv k$, the adaptive rollout distribution $P^{\circ}_{\mathcal{D},\kappa}$ reduces to the standard open-loop rollout distribution $P^{\circ}_{\mathcal{D}}$ used in DQC's definition of OLC (Li et al., 2026). The TV distance constraints Equations (21) and (22) then become exactly the weak OLC constraints Equations 8 and 9 from DQC's Definition 2. Since DQC's strong OLC Equation 10 imposes an additional per-action-sequence constraint, AOLC with constant κ implies weak OLC and is strictly weaker than strong OLC.

I.2 Proof of Theorem H.5

Let $f_k(s) := \max_{\mathbf{a}_{t:t+k}} A^{k,*}(s, \mathbf{a}_{t:t+k})$ and $\hat{f}_k(s) := \max_{\mathbf{a}_{t:t+k}} \hat{A}^k(s, \mathbf{a}_{t:t+k})$, where $A^{k,*}$ and $\hat{A}^k$ denote the true and estimated per-scale advantages, respectively.

For any s and $\mathbf{a}_{t:t+k}$,

$$|\hat{A}^k(s, \mathbf{a}_{t:t+k}) - A^{k,*}(s, \mathbf{a}_{t:t+k})| = \left| \frac{Q^k(s, \mathbf{a}_{t:t+k}) - V^k(s)}{\gamma^k} - \frac{Q^{k,*}(s, \mathbf{a}_{t:t+k}) - V^{k,*}(s)}{\gamma^k} \right| \tag{48}$$

$$\leq \frac{|Q^k(s, \mathbf{a}_{t:t+k}) - Q^{k,*}(s, \mathbf{a}_{t:t+k})| + |V^k(s) - V^{k,*}(s)|}{\gamma^k} \tag{49}$$

$$\leq \frac{\varepsilon_k + \delta_k}{\gamma^k} \leq \frac{\bar{\varepsilon}}{\gamma^{k_{\min}}}. \tag{50}$$

Taking the maximum over $\mathbf{a}_{t:t+k}$, we get $|\hat{f}_k(s) - f_k(s)| \leq \bar{\varepsilon}/\gamma^{k_{\min}}$.

By Δ-advantage separability, $f_{k^\dagger}(s) - \max_{k \neq k^\dagger} f_k(s) \geq \Delta(s)$. For the empirical selector to mis-select, we need $\hat{f}_{k'}(s) > \hat{f}_{k^\dagger}(s)$ for some $k' \neq k^\dagger$. By the triangle inequality:

$$\hat{f}_{k'}(s) - \hat{f}_{k^\dagger}(s) \leq f_{k'}(s) - f_{k^\dagger}(s) + |\hat{f}_{k'}(s) - f_{k'}(s)| + |\hat{f}_{k^\dagger}(s) - f_{k^\dagger}(s)| \tag{51}$$

$$\leq -\Delta(s) + 2\bar{\varepsilon}/\gamma^{k_{\min}}. \tag{52}$$

If $\bar{\varepsilon} < \Delta\gamma^{k_{\min}}/2$, then $\hat{f}_{k'}(s) - \hat{f}_{k^\dagger}(s) < 0$ for all $k' \neq k^\dagger$, so $\hat{k}(s) = k^\dagger(s)$.

For the probabilistic bound when $\bar{\varepsilon} \geq \Delta\gamma^{k_{\min}}/2$, a mis-selection requires the estimation error to bridge at least half the advantage gap. By Markov's inequality applied to the estimation error:

$$\mathbb{P}(\hat{k}(s) \neq k^\dagger(s)) \leq \mathbb{P}\left(|\hat{A}^{\hat{k}(s)}(s) - A^{k^\dagger,*}(s)| \geq \Delta(s)/2\right) \tag{53}$$

$$\leq \frac{2\bar{\varepsilon}}{\gamma^{k_{\min}}\Delta(s)}. \tag{54}$$

I.3 Proof of Proposition H.6

Decompose the regret into two cases: correct selection and mis-selection.

When $\hat{k}(s) = k^\dagger(s)$, the regret is bounded by the critic error:

$$|\hat{A}^{k^\dagger}(s) - A^{k^\dagger,*}(s)| \leq \bar{\varepsilon}/\gamma^{k_{\min}}. \tag{55}$$

When $\hat{k}(s) \neq k^\dagger(s)$ (occurring with probability at most $2\bar{\varepsilon}/(\gamma^{k_{\min}}\Delta(s))$ by Theorem H.5), the regret is bounded by the maximum advantage range:

$$|\hat{A}^{\hat{k}(s)}(s) - A^{k^\dagger,*}(s)| \leq \bar{\varepsilon}/\gamma^{k_{\min}} + \max_k \operatorname{diam}(A)^k. \tag{56}$$

Combining:

$$\mathbb{E}\left[|\hat{A}^{\hat{k}(s)}(s) - A^{k^\dagger,*}(s)|\right] \leq (1 - p_{\text{err}}) \cdot \bar{\varepsilon}/\gamma^{k_{\min}} + p_{\text{err}} \cdot (\bar{\varepsilon}/\gamma^{k_{\min}} + \max_k \operatorname{diam}(A)^k) \tag{57}$$

$$= \bar{\varepsilon}/\gamma^{k_{\min}} + \frac{2\bar{\varepsilon}}{\gamma^{k_{\min}}\Delta(s)} \cdot \max_k \operatorname{diam}(A)^k. \tag{58}$$

I.4 Proof of Theorem H.8

We construct a meta-MDP $\tilde{\mathcal{M}} = (\mathcal{S}, \tilde{\mathcal{A}}, \tilde{T}, \tilde{r}, \gamma)$ where: - $\tilde{\mathcal{A}} = \bigcup_{k\in\mathcal{K}}\{(k, \mathbf{a}_{t:t+k}) : \mathbf{a}_{t:t+k} \in \mathcal{A}^k\}$ is the set of meta-actions. - $\tilde{T}(s' \mid s, (k, \mathbf{a}_{t:t+k}))$ is the distribution of s_{t+k} after executing $\mathbf{a}_{t:t+k}$ open-loop for k steps from $s_t = s$. - $\tilde{r}(s, (k, \mathbf{a}_{t:t+k})) = \sum_{j=0}^{k-1} \gamma^j r(s_{t+j}, a_{t+j})$ is the cumulative discounted reward.

In this meta-MDP, both π^{AQC} and π^k are valid policies. The performance difference lemma (Kakade & Langford, 2002) gives:

$$V^{\text{AQC}}(s) - V^k(s) = \frac{1}{1-\gamma}\mathbb{E}_{s'\sim d^{\text{AQC}}}\left[Q^{k,*}_{\text{meta}}(s', (k^*(s'), a^*)) - Q^{k,*}_{\text{meta}}(s', (k, a^k))\right], \tag{59}$$

where $Q^{k,*}_{\text{meta}}$ is the optimal meta-action value function.

Using the relationship between meta-action values and per-scale advantages:

$$Q^{k,*}_{\text{meta}}(s, (k, \mathbf{a}_{t:t+k})) = \sum_{j=0}^{k-1} \gamma^j r_{t+j} + \gamma^k V^*(s_{t+k}) \tag{60}$$

$$= V^{k,*}(s) + \gamma^k A^{k,*}(s, \mathbf{a}_{t:t+k}), \tag{61}$$

so the advantage in the meta-MDP is:

$$Q^{k,*}_{\text{meta}}(s, (k, \mathbf{a}_{t:t+k})) - \max_{j, \mathbf{a}_{t:t+j}} Q^{j,*}_{\text{meta}}(s, (j, \mathbf{a}_{t:t+j})) = \gamma^k \left(A^{k,*}(s, \mathbf{a}_{t:t+k}) - \bar{A}^{k^\dagger,*}(s)\right). \tag{62}$$

Under advantage separability with $\bar{\varepsilon} < \Delta/2$, the selector chooses $k^*(s') = k^\dagger(s')$, so:

$$\mathbb{E}_{s'\sim d^{\text{AQC}}}\left[Q^{k,*}_{\text{meta}}(s', (k^*(s'), a^*)) - Q^{k,*}_{\text{meta}}(s', (k, a^k))\right] \tag{63}$$

$$= \mathbb{E}_{s'\sim d^{\text{AQC}}}\left[\gamma^{k^*(s')}\bar{A}^{k^\dagger,*}(s') - \gamma^k \bar{A}^{k,*}(s')\right]. \tag{64}$$

Since $\gamma^{k^*(s')} \geq \gamma^{k_{\min}}$ for all s' and $\gamma^k \leq \gamma^{k_{\min}}$, and noting that $\bar{A}^{k,*}(s') \geq 0$ (as the max advantage is non-negative at the behavior policy's best action), we lower-bound:

$$\gamma^{k^*(s')}\bar{A}^{k^\dagger,*}(s') - \gamma^k \bar{A}^{k,*}(s') \;\geq\; \gamma^{k_{\min}}\left(\bar{A}^{k^\dagger,*}(s') - \bar{A}^{k,*}(s')\right). \tag{65}$$

Accounting for the selector error probability $(2\bar{\varepsilon}/(\gamma^{k_{\min}}\Delta))$:

$$V^{\text{AQC}}(s) - V^k(s) \geq \frac{\gamma^{k_{\min}}(1 - 2\bar{\varepsilon}/(\gamma^{k_{\min}}\Delta))}{1-\gamma}\mathbb{E}_{s'\sim d^{\text{AQC}}}\left[\bar{A}^{k^\dagger,*}(s') - \bar{A}^{k,*}(s')\right]. \tag{66}$$

I.5 Proof of Theorem H.11

At each re-query point s, the selector mis-selects with probability at most $2\bar{\varepsilon}_{\mathcal{K}}/(\gamma^{k_{\min}}\Delta(s))$ by Theorem H.5.

When a mis-selection occurs (choosing k' instead of $k^\dagger$), the value loss is:

$$|V^{k^\dagger}(s) - V^{k'}(s)| \le \max_{\mathbf{a}_{t:t+k^\dagger}} Q^{k^\dagger,*}(s, \mathbf{a}_{t:t+k^\dagger}) - \min_{\mathbf{a}_{t:t+k'}} Q^{k',*}(s, \mathbf{a}_{t:t+k'}) \tag{67}$$

$$\le \frac{\bar{R}_{k'}}{\gamma^{k'}} \le \max_k \frac{\bar{R}_k}{\gamma^k}. \tag{68}$$

The expected number of re-queries in the discounted horizon is at most $1/(1-\gamma)$ (since each re-query takes at least 1 step). The expected number of mis-selections is therefore at most $(1/(1-\gamma)) \cdot 2\bar{\varepsilon}_{\mathcal{K}}/(\gamma^{k_{\min}}\Delta(s))$.

The total regret is the product:

$$|V^{\mathrm{AQC}}(s) - V^\dagger(s)| \le \frac{2\bar{\varepsilon}_{\mathcal{K}}}{(1-\gamma)\gamma^{k_{\min}}\Delta(s)} \cdot \max_k \frac{\bar{R}_k}{\gamma^k}. \tag{69}$$

I.6 Proof of Theorem H.14

Let $\pi^\dagger_{\mathrm{CL}}$ denote the oracle closed-loop policy that selects $k^\dagger(s)$ at each step and executes the first action only.

Part 1: Oracle closed-loop optimality. Under $\varepsilon_{\mathcal{K}}$-AOLC, at each re-query point $s_{t+\kappa(s)}$, the state distribution under open-loop execution deviates from the data distribution by at most $\varepsilon_{\mathcal{K}}$ in TV distance.

By the standard TV-to-value bound (Li et al., 2026), the per-re-query value estimation error is at most $\varepsilon_{\mathcal{K}}$ times the value range scaled by the effective horizon.

The number of re-queries in the discounted horizon is bounded by $1/(1-\gamma^{k_{\min}})$, since at minimum $k_{\min}$ steps elapse between re-queries.

Following the same derivation as DQC's Proposition 3, but with $k_{\min}$ replacing h in the geometric series:

$$V^\star(s) - V^\dagger_{\mathrm{CL}}(s) \le \frac{\varepsilon_{\mathcal{K}}\gamma}{(1-\gamma)^2}\left[\frac{2}{1-(1-2\varepsilon_{\mathcal{K}})\gamma^{k_{\min}}} + \frac{1}{1-(1-\varepsilon_{\mathcal{K}})\gamma^{k_{\min}}}\right] \tag{70}$$

$$\le \frac{3\varepsilon_{\mathcal{K}}\gamma}{(1-\gamma)^2(1-\gamma^{k_{\min}})} = 3\varepsilon_{\mathcal{K}}H^2\bar{H}_{k_{\min}}. \tag{71}$$

Part 2: Learned selector. The learned selector introduces additional regret bounded by Theorem H.11:

$$|V^\dagger_{\mathrm{CL}}(s) - V^\bullet(s)| \le \frac{2\bar{\varepsilon}_{\mathcal{K}}}{(1-\gamma)\gamma^{k_{\min}}\Delta} \cdot \max_k \frac{\bar{R}_k}{\gamma^k}. \tag{72}$$

Combining both parts via the triangle inequality:

$$V^\star(s) - V^\bullet(s) \le (V^\star(s) - V^\dagger_{\mathrm{CL}}(s)) + |V^\dagger_{\mathrm{CL}}(s) - V^\bullet(s)| \tag{73}$$

$$\le 3\varepsilon_{\mathcal{K}}H^2\bar{H}_{k_{\min}} + \frac{2\bar{\varepsilon}_{\mathcal{K}}}{(1-\gamma)\gamma^{k_{\min}}\Delta} \cdot \max_k \frac{\bar{R}_k}{\gamma^k}. \tag{74}$$

I.7 Proof of Theorem H.15

Let $\mathcal{T}^k_h$ denote the k-step Bellman operator with V^h bootstrap:

$$(\mathcal{T}^k_h Q)(s, \mathbf{a}_{t:t+k}) := \sum_{j=0}^{k-1} \gamma^j r_{t+j} + \gamma^k V^h(s_{t+k}). \tag{75}$$

This operator is a γ^k-contraction in the sup-norm:

$$\|\mathcal{T}_h^k Q_1 - \mathcal{T}_h^k Q_2\|_\infty \le \gamma^k \|V_1^h - V_2^h\|_\infty \tag{76}$$
$$\le \gamma^k \|Q_1 - Q_2\|_\infty. \tag{77}$$

Let Q_{fp}^k be its unique fixed point: $Q_{\text{fp}}^k = \mathcal{T}_h^k Q_{\text{fp}}^k$.

For the optimal k-step value $Q^{k,*}$:

$$|Q^{k,*}(s,a) - Q_{\text{fp}}^k(s,a)| = |\mathcal{T}^k Q^{k,*}(s,a) - \mathcal{T}_h^k Q_{\text{fp}}^k(s,a)| \tag{78}$$
$$= \left| \sum_{j=0}^{k-1} \gamma^j r_{t+j} + \gamma^k V^*(s_{t+k}) - (\sum_{j=0}^{k-1} \gamma^j r_{t+j} + \gamma^k V^h(s_{t+k})) \right| \tag{79}$$
$$= \gamma^k |V^*(s_{t+k}) - V^h(s_{t+k})| \tag{80}$$
$$\le \gamma^k \|V^* - V^h\|_\infty \;=\; \gamma^k \varepsilon_h. \tag{81}$$

The TD fitting error gives $\|Q^k - Q_{\text{fp}}^k\|_\infty \le \varepsilon_k/(1-\gamma^k)$ by the contraction mapping theorem.

By the triangle inequality:

$$\|Q^k - Q^{k,*}\|_\infty \le \|Q^k - Q_{\text{fp}}^k\|_\infty + \|Q_{\text{fp}}^k - Q^{k,*}\|_\infty \tag{82}$$
$$\le \frac{\varepsilon_k}{1-\gamma^k} + \frac{\gamma^k \varepsilon_h}{1-\gamma^k}. \tag{83}$$

Summing over $k \in \mathcal{K}$:

$$\sum_{k\in\mathcal{K}} \|Q^k - Q^{k,*}\|_\infty \le \varepsilon_h \sum_{k\in\mathcal{K}} \frac{\gamma^k}{1-\gamma^k} + \sum_{k\in\mathcal{K}} \frac{\varepsilon_k}{1-\gamma^k}. \tag{84}$$

I.8 Proof of Corollary H.16

If Q^k were bootstrapped from V^1 instead of V^h, the Bellman operator would be:

$$(\mathcal{T}_1^k Q)(s, \mathbf{a}_{t:t+k}) := \sum_{j=0}^{k-1} \gamma^j r_{t+j} + \gamma^k V^1(s_{t+k}). \tag{85}$$

The fixed point gap to $Q^{k,*}$ would be $\gamma^k \|V^1 - V^*\|_\infty = \gamma^k \varepsilon_1$.

However, V^1 itself must be constructed from per-step rewards. By the standard n-step return bias analysis (Fedus et al., 2020; Kozuno et al., 2021), chaining $O(H) = O(1/(1-\gamma))$ 1-step bootstrap steps to reach the same horizon accumulates cumulative error $O(H\varepsilon_1)$, since each step contributes independent approximation error.

Thus:

$$\|Q^k - Q^{k,*}\|_\infty \le \frac{\gamma^k}{1-\gamma^k} H\varepsilon_1 + \frac{1}{1-\gamma^k}\varepsilon_k. \tag{86}$$

The ratio of bootstrap terms between 1-step and V^h bootstrap is:

$$\frac{H\varepsilon_1}{\varepsilon_h} = O(H), \tag{87}$$

assuming $\varepsilon_1 \approx \varepsilon_h$.

I.9 Proof of Theorem H.17

From Theorem H.15, for $k_1 < k_2$:

$$\|Q^{k_1} - Q^{k_1,*}\|_\infty \leq \frac{\gamma^{k_1}}{1-\gamma^{k_1}}\varepsilon_h + \frac{\varepsilon_{k_1}}{1-\gamma^{k_1}}, \tag{88}$$

$$\|Q^{k_2} - Q^{k_2,*}\|_\infty \leq \frac{\gamma^{k_2}}{1-\gamma^{k_2}}\varepsilon_h + \frac{\varepsilon_{k_2}}{1-\gamma^{k_2}}. \tag{89}$$

Multiplying the second inequality by $\gamma^{k_1}/\gamma^{k_2}$:

$$\frac{\gamma^{k_1}}{\gamma^{k_2}}\|Q^{k_2} - Q^{k_2,*}\|_\infty \geq \frac{\gamma^{k_1}}{\gamma^{k_2}} \cdot \frac{\gamma^{k_2}}{1-\gamma^{k_2}}\varepsilon_h + \frac{\gamma^{k_1}}{\gamma^{k_2}} \cdot \frac{\varepsilon_{k_2}}{1-\gamma^{k_2}} \tag{90}$$

$$= \frac{\gamma^{k_1}}{1-\gamma^{k_2}}\varepsilon_h + \frac{\gamma^{k_1}/\gamma^{k_2}}{1-\gamma^{k_2}}\varepsilon_{k_2}. \tag{91}$$

Since $1-\gamma^{k_2} \geq 1-\gamma^{k_1}$, we have:

$$\frac{\gamma^{k_1}}{1-\gamma^{k_2}}\varepsilon_h \geq \frac{\gamma^{k_1}}{1-\gamma^{k_1}}\varepsilon_h \cdot \frac{1-\gamma^{k_1}}{1-\gamma^{k_2}} \tag{92}$$

$$\geq \frac{\gamma^{k_1}}{1-\gamma^{k_1}}\varepsilon_h, \tag{93}$$

since $k_1 < k_2$ implies $(1-\gamma^{k_1})/(1-\gamma^{k_2}) < 1$.

Subtracting the Q^{k_1} bound from the scaled Q^{k_2} bound:

$$\frac{\gamma^{k_1}}{\gamma^{k_2}}\|Q^{k_2} - Q^{k_2,*}\|_\infty - \|Q^{k_1} - Q^{k_1,*}\|_\infty \geq \frac{\gamma^{k_1}/\gamma^{k_2}}{1-\gamma^{k_2}}\varepsilon_{k_2} - \frac{\varepsilon_{k_1}}{1-\gamma^{k_1}}, \tag{94}$$

yielding:

$$\|Q^{k_1} - Q^{k_1,*}\|_\infty \leq \frac{\gamma^{k_1}}{\gamma^{k_2}}\|Q^{k_2} - Q^{k_2,*}\|_\infty + \frac{\varepsilon_{k_1} - \frac{\gamma^{k_1}}{\gamma^{k_2}}\varepsilon_{k_2}}{1-\gamma^{k_1}}. \tag{95}$$

When $\varepsilon_k \equiv \varepsilon$ for all k:

$$\|Q^{k_1} - Q^{k_1,*}\|_\infty \leq \frac{\gamma^{k_1}}{\gamma^{k_2}}\|Q^{k_2} - Q^{k_2,*}\|_\infty + \varepsilon \cdot \frac{1-\gamma^{k_1}/\gamma^{k_2}}{1-\gamma^{k_1}}. \tag{96}$$

Since $1-\gamma^{k_1}/\gamma^{k_2} > 0$, the second term is positive, meaning the shorter-horizon critic's error is strictly bounded by a fraction of the longer-horizon critic's error plus a small correction.

I.10 Proof of Theorem H.18

DQC's policy uses a fixed partial chunk size h_a. By Theorem H.8 with $k = h_a$:

$$V^{\mathrm{AQC}}(s) - V^{\mathrm{DQC}}(s) \geq \frac{\gamma^{k_{\min}}(1 - 2\bar{\varepsilon}/(\gamma^{k_{\min}}\Delta))}{1-\gamma}\mathbb{E}_{s'\sim d^{\mathrm{AQC}}}\left[\bar{A}^{k^\dagger,*}(s') - \bar{A}^{h_a,*}(s')\right]. \tag{97}$$

Decompose the expectation over $\mathcal{S}_{\mathrm{adapt}}$ and its complement:

$$\mathbb{E}_{s'\sim d^{\mathrm{AQC}}}\left[\bar{A}^{k^\dagger,*}(s') - \bar{A}^{h_a,*}(s')\right] \tag{98}$$

$$= d^{\mathrm{AQC}}(\mathcal{S}_{\mathrm{adapt}}) \cdot \mathbb{E}[\cdot \mid s' \in \mathcal{S}_{\mathrm{adapt}}] + (1 - d^{\mathrm{AQC}}(\mathcal{S}_{\mathrm{adapt}})) \cdot \mathbb{E}[\cdot \mid s' \notin \mathcal{S}_{\mathrm{adapt}}] \tag{99}$$

$$\geq \rho \cdot \mathbb{E}_{s'\sim d^{\mathrm{AQC}}}[\delta_{\mathrm{adapt}}(s') \mid s' \in \mathcal{S}_{\mathrm{adapt}}], \tag{100}$$

since $\bar{A}^{k^\dagger,*}(s') \geq \bar{A}^{h_a,*}(s')$ for all s' and $\geq \delta_{\mathrm{adapt}}(s')$ on $\mathcal{S}_{\mathrm{adapt}}$.

Accounting for the selector error probability:

$$V^{\mathrm{AQC}}(s) - V^{\mathrm{DQC}}(s) \geq \frac{\gamma^{k_{\min}}\rho}{1-\gamma} \cdot \mathbb{E}_{s'\sim d^{\mathrm{AQC}}}[\delta_{\mathrm{adapt}}(s') \mid s' \in \mathcal{S}_{\mathrm{adapt}}] - \frac{2\bar{\varepsilon}_{\mathcal{K}}}{(1-\gamma)\gamma^{k_{\min}}\Delta} \cdot \max_k \frac{\bar{R}_k}{\gamma^k}. \tag{101}$$

I.11 Proof of Theorem H.19

By DQC's comparison theorem (Li et al., 2026), the chunked critic's nominal value satisfies:

$$\hat{V}^{+}_{\mathrm{ac}}(s) - \hat{V}^{+}_{n}(s) \geq \delta_n \bar{H}_n - 3\varepsilon_{\mathcal{K}} H \bar{H}_{k_{\min}}. \tag{102}$$

AQC's value is at least as large as the best fixed-chunk policy's value (by Theorem H.8), minus the selector regret:

$$V^{\mathrm{AQC}}(s) \geq \hat{V}^{+}_{\mathrm{ac}}(s) - \frac{2\bar{\varepsilon}_{\mathcal{K}}}{(1-\gamma)\gamma^{k_{\min}}\Delta} \cdot \max_k \frac{\bar{R}_k}{\gamma^k}. \tag{103}$$

Combining:

$$V^{\mathrm{AQC}}(s) - V^{n}(s) \geq \delta_n \bar{H}_n - 3\varepsilon_{\mathcal{K}} H \bar{H}_{k_{\min}} - \frac{2\bar{\varepsilon}_{\mathcal{K}}}{(1-\gamma)\gamma^{k_{\min}}\Delta} \cdot \max_k \frac{\bar{R}_k}{\gamma^k}. \tag{104}$$

I.12 Proof of Proposition 5.1

Under the sparse-reward approximation, $Q^k(s_t, \mathbf{a}_{t:t+k}) \approx \gamma^k V^h(s^a_{t+k})$ and $V^k(s_t) \approx \mathbb{E}_{\pi_\beta}[\gamma^k V^h(s_{t+k}) \mid s_t]$. When $V^h(s) \leq \epsilon$ for all reachable states, both terms are bounded. Including function approximation errors bounded by σ:

$$\left|\frac{Q^k(s_t, \mathbf{a}_{t:t+k})}{\gamma^k}\right| \leq \left|V^h(s^a_{t+k})\right| + \sigma \leq \epsilon + \sigma, \tag{105}$$

$$\left|\frac{V^k(s_t)}{\gamma^k}\right| \leq \left|\mathbb{E}_{\pi_\beta}\left[V^h(s_{t+k}) \mid s_t\right]\right| + \sigma \leq \epsilon + \sigma. \tag{106}$$

By the triangle inequality:

$$|\delta_k(s)| = \left|\frac{Q^k(s_t, \mathbf{a}_{t:t+k})}{\gamma^k} - \frac{V^k(s_t)}{\gamma^k}\right| \tag{107}$$

$$\leq \left|\frac{Q^k(s_t, \mathbf{a}_{t:t+k})}{\gamma^k}\right| + \left|\frac{V^k(s_t)}{\gamma^k}\right| \tag{108}$$

$$\leq \epsilon + 2\sigma. \tag{109}$$

When $\epsilon \ll \sigma$ (far from rewards, the value signal is smaller than the noise floor), the advantage is dominated by approximation noise and all k score near-zero.

By contrast, for the uncorrected selector $\arg\max_k Q^k/\gamma^k$, when all $Q^k/\gamma^k \approx \epsilon + \sigma_k$, the argmax picks the scale with the largest positive noise realization σ_k. Since the maximum of finitely many noisy estimates is biased upward, this produces a systematically biased choice that does not average out over repeated queries. The advantage selector centers each score around zero via subtraction of V^k/γ^k, removing this bias.

I.13 Proof of Proposition H.21

Let $\mathcal{T}^k_h$ denote the k-step Bellman operator using $\bar{V}^h_\xi$ as bootstrap:

$$(\mathcal{T}^k_h Q)(s_t, \mathbf{a}_{t:t+k}) := \sum_{j=0}^{k-1} \gamma^j r_{t+j} + \gamma^k \bar{V}^h_\xi(s_{t+k}). \tag{110}$$

This operator is a γ^k-contraction in the sup-norm. Let Q^k_{fp} be its unique fixed point: $Q^k_{\mathrm{fp}} = \mathcal{T}^k_h Q^k_{\mathrm{fp}}$.

For the optimal k-step action-value $Q^{k,*}$:

$$|Q^{k,*}(s,a) - Q^k_{\mathrm{fp}}(s,a)| = |\mathcal{T}^k Q^{k,*}(s,a) - \mathcal{T}^k_h Q^k_{\mathrm{fp}}(s,a)| \tag{111}$$

$$= \left|\gamma^k V^*(s_{t+k}) - \gamma^k \bar{V}^h_\xi(s_{t+k})\right| \tag{112}$$

$$\leq \gamma^k \|V^*(s_{t+k}) - \bar{V}^h_\xi(s_{t+k})\|_\infty \tag{113}$$

$$\leq \gamma^k \varepsilon_h. \tag{114}$$

Since $\mathcal{T}_h^k$ is a γ^k-contraction, repeated application gives:

$$\left\|Q_\psi^k - Q_{\text{fp}}^k\right\|_\infty \leq \frac{\varepsilon_k}{1-\gamma^k}. \tag{115}$$

By the triangle inequality:

$$\left\|Q_\psi^k - Q^{k,*}\right\|_\infty \leq \left\|Q_\psi^k - Q_{\text{fp}}^k\right\|_\infty + \left\|Q_{\text{fp}}^k - Q^{k,*}\right\|_\infty \tag{116}$$

$$\leq \frac{\varepsilon_k}{1-\gamma^k} + \frac{\gamma^k}{1-\gamma^k}\,\varepsilon_h, \tag{117}$$

yielding the first bound.

For the 1-step bootstrap comparison, a 1-step value function V^1 must be constructed via $O(H) = O(1/(1-\gamma))$ TD updates from the per-step reward, each step contributing error ε_1. By the standard n-step return bias analysis (Fedus et al., 2020; Kozuno et al., 2021), the resulting error in the bootstrap target is $O(H\,\varepsilon_1)$, which replaces ε_h in the bound above to give the second result.

I.14 Proof of Proposition H.22

At the fixed point of the TD loss Equation (14), AQC's partial critic satisfies

$$Q_\psi^k(s_t, a_{t:t+k}) = \sum_{j=0}^{k-1} \gamma^j r_{t+j} + \gamma^k \bar{V}_\xi^h(s_{t+k}). \tag{118}$$

This differs from Q_ψ^P only in the bootstrap target: V^* in DQC versus V_ξ^h in AQC. Since $\|V_\xi^h - V^*\|_\infty \leq \varepsilon_h$ by definition, the gap between the two fixed points is:

$$\left\|Q_\psi^k - Q_\psi^P\right\|_\infty \leq \gamma^k \|\bar{V}_\xi^h - V^*\|_\infty \tag{119}$$

$$\leq \gamma^k \varepsilon_h. \tag{120}$$

This is $O(\varepsilon_h)$ for bounded k and $\gamma < 1$.

The implication: AQC realizes DQC's objective through a simpler mechanism. DQC requires goal-conditioned hindsight relabeling and a distillation hyperparameter κ_d to construct V^*. AQC reuses V^h, which is already trained for the $k = h$ baseline, requiring neither.