NOT ALL TRANSITIONS ARE EQUAL: UNCERTAINTY-DRIVEN CONTRASTIVE LEARNING FOR VISUAL RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent works show that prioritizing transitions from the replay buffer can improve representation learning by focusing on informative experiences, potentially guiding exploration and improving sample efficiency. We present UCRL: **U**ncertainty-Driven **C**ontrastive Learning for Visual **R**einforcement **L**earning, a method that leverages transition uncertainty to prioritize experiences for contrastive representation learning. Unlike existing contrastive RL methods, which uniformly sample transitions for representation learning, UCRL modifies only the transition sampling distribution of the contrastive objective. It prioritizes transitions according to epistemic uncertainty estimated from an ensemble of critics. We theoretically establish that nonzero epistemic uncertainty lower-bounds the encoder gradient induced by the critic objective, identifying epistemic uncertainty as an observable proxy for representation learning potential. We apply UCRL to CURL and evaluate the resulting method on the DeepMind Control Suite 100K and Atari 100K benchmarks, achieving relative performance gains of 9.5% and 14.8%, respectively, over CURL. Across five of the six main DeepMind Control Suite tasks, UCRL reaches CURL's 100k performance using 46.4% fewer environment interactions on average, with reductions of up to 65% across individual tasks. Code will be available upon acceptance.

1 INTRODUCTION

Reinforcement learning (RL) from high-dimensional visual observations requires agents to jointly learn task-relevant representations and control policies directly from raw sensory inputs (Hafner et al., 2022; Laskin et al., 2020; Li et al., 2025; Pan et al., 2022; Zhang et al., 2024). The quality of these representations is therefore fundamental to effective control, yet learning useful representations from high-dimensional observations is challenging, particularly under limited interaction data. This challenge is reflected in the substantially lower sample efficiency of visual RL compared with RL using low-dimensional state representations (Zhang et al., 2026).

To address this challenge, prior works have introduced representation learning objectives that improve the quality of latent features learned from visual observations (Poudel et al., 2024; Kich et al., 2024; Laskin et al., 2020; Hafner et al., 2025; Zhalehmehrabi et al., 2026). Among these approaches, contrastive learning has become a widely adopted paradigm for improving visual representations in RL. Methods such as CURL (Laskin et al., 2020) learn representations by maximizing agreement between augmented views of the same observation while contrasting them against other samples in the batch, achieving substantial gains in sample efficiency over reconstruction-based objectives and even outperforming several model-based approaches in low-data settings.

An important consideration in representation learning is determining which samples should receive representation-learning updates (Robinson et al., 2021). Existing contrastive RL methods typically construct contrastive batches via *uniform replay sampling*, treating all stored transitions identically despite substantial variation in their usefulness for representation learning. Replay buffers may contain stale, well-learned transitions collected by outdated policies. As a result, uniformly sampled batches can contain a substantial fraction of such transitions (Meng et al., 2026; Schaul et al., 2015). Sampling these transitions can introduce two problems. Repeatedly applying contrastive updates to already well-learned transitions can drive representations toward dimensional collapse (Jing et al., 2022), while more broadly, over-updating on familiar data is known to reduce a network's ability to

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.

benefit from new experience (Nikishin et al., 2022). Consequently, uniform sampling may allocate representation-learning updates to transitions that provide limited benefit or may even hinder further representation learning. This raises a fundamental question: *which transitions should be prioritized for improving visual representations?*

A natural solution is to prioritize transitions using temporal-difference (TD) error, as in prioritized experience replay (PER) (Schaul et al., 2015). However, TD error reflects the critic's value prediction error rather than the quality of the learned visual representation, and it does not directly identify transitions that are most beneficial for improving the representation (An et al., 2021; Carrasco-Davis et al., 2025). We argue that transition selection for contrastive learning should account for the learning state of each transition in the RL branch, which can provide information about its representation-learning potential. We therefore seek a prioritization signal that identifies transitions whose representations remain amenable to further adaptation.

To this end, we propose UCRL (**U**ncertainty-Driven **C**ontrastive Learning for Visual **RL**), a general uncertainty-driven transition-prioritization strategy for contrastive representation learning. We instantiate and evaluate this strategy within CURL, yielding UCRL-C (UCRL applied to CURL, See Figure 1), using epistemic uncertainty to prioritize transitions for constructing contrastive batches while leaving the RL branch unchanged. Epistemic uncertainty estimated from an ensemble of critics identifies regions where current value estimates remain uncertain because of insufficient knowledge (Lakshminarayanan et al., 2017; Osband et al., 2016; Kendall & Gal, 2017). We show in Section 4 that epistemic uncertainty lower-bounds the encoder gradient magnitude, identifying transitions that necessarily induce nonzero encoder updates. It therefore provides the learning signal we seek for prioritizing transitions with greater potential for representation learning.

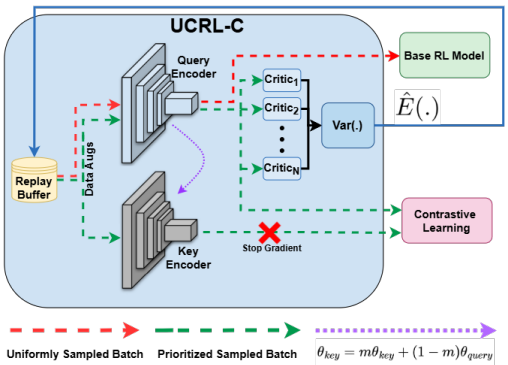


Figure 1: Architecture and data flow of UCRL-C

Unlike PER, which modifies the sampling distribution of the RL objective itself, we restrict prioritization exclusively to the contrastive learning branch. This design choice is motivated by the different roles of the two objectives. Applying sample prioritization to the RL objective can bias policy-gradient estimation and destabilize value learning in actor-critic methods (Saglam et al., 2023; Li et al., 2024), because the RL objectives are defined with respect to the replay distribution (Fujimoto et al., 2020; Cicek et al., 2021). We therefore retain uniform sampling for the RL branch, keeping its sampling distribution and optimization objective unchanged. Instead, the contrastive objective can benefit from selectively emphasizing informative transitions (Robinson et al., 2021; Meng et al., 2026). Although the contrastive updates modify the shared encoder and therefore indirectly affect the representations used by the policy and critics, they do not alter the sampling distribution or loss formulation of the RL updates themselves. This separation allows UCRL to improve representation learning while preserving the underlying RL optimization procedure.

We evaluate UCRL-C on the DeepMind Control (Tassa et al., 2018) and Atari (Kaiser et al., 2020) benchmarks on the standard 100k budget to assess the effect of uncertainty-driven transition prioritization under limited interaction budgets. On the full DeepMind Control Suite, when CURL is combined with SAC (Haarnoja et al., 2018) as the underlying RL algorithm, UCRL-C outperforms CURL on 10 of 16 tasks, performs comparably on 4, and underperforms on 2. Considering only tasks on which both UCRL-C and CURL achieve nonzero performance, UCRL-C achieves a mean relative improvement of 9.5% and a median per-task gain of 12.9%. On the five of six main DeepMind Control tasks where UCRL reaches CURL's 100k-interaction performance within the evaluation budget, it does so using 46.4% fewer environment interactions on average. Moreover, UCRL-C surpasses strong existing methods on three out of six main tasks. On Atari 100K, where CURL is built on DQN (Mnih et al., 2015), UCRL-C outperforms CURL on 17 of 26 games and improves the interquartile mean (IQM) human-normalized score (HNS) from 11.72% to 15.45%, corresponding to a 31.8% relative improvement. These results across two visual RL benchmarks and two underlying RL algorithms show that selectively allocating representation-learning updates can improve performance without

modifying the underlying RL objective, supporting transition selection as an important dimension of representation learning in visual RL.

Our contributions are summarized as follows:

- We propose UCRL, an uncertainty-driven transition selection mechanism that allocates contrastive representation-learning updates toward informative experiences without modifying the underlying RL objective.
- We theoretically show that epistemic uncertainty lower-bounds the encoder gradient induced by the critic objective and can identify transitions associated with continued encoder adaptation, providing a principled distinction from TD-error-based prioritization.
- We apply UCRL to CURL and evaluate the resulting UCRL-C across two visual RL benchmarks and two underlying RL algorithms. UCRL-C outperforms CURL on 10 of 16 DeepMind Control Suite tasks and 17 of 26 Atari 100K games, and surpasses strong existing methods on three of the six main DeepMind Control tasks.

2 Related Works

Visual RL methods improve representations through data augmentation (Yarats et al., 2021b;a), auxiliary objectives such as reconstruction (Yarats et al., 2021c), self-prediction (Schwarzer et al., 2021), and contrastive learning (Laskin et al., 2020), or latent dynamics models (Lee et al., 2020; Hafner et al., 2020). These works focus on *what* objective to optimize, whereas UCRL addresses *which* transitions receive representation-learning updates. Prioritized replay methods instead reweight transitions by TD error (Schaul et al., 2015; Saglam et al., 2023; Yenicesu et al., 2024) or epistemic uncertainty (Carrasco-Davis et al., 2025), but they modify the sampling distribution of the RL objective itself. UCRL leaves the RL branch untouched and applies uncertainty-based prioritization only to the contrastive branch. An extended discussion is provided in Appendix A.

3 Method

UCRL proposes a transition-sampling strategy for the contrastive learning branch, prioritizing transitions according to their estimated epistemic uncertainty. The RL branch remains unchanged, making UCRL a lightweight, modular component that can be integrated into existing visual RL frameworks.

Transition Prioritization. To quantify epistemic uncertainty, we compute the variance of the predictions of an ensemble of critics, $\hat{E}(s,a) = \mathrm{Var}_{n=1,\dots,N}\left(Q_{\psi_n}(f_\phi(s),a)\right)$, where $\psi = \{\psi_1,\dots,\psi_N\}$ denotes the parameters of the ensemble critics, N is the number of critics, and ϕ denotes the parameters of the shared encoder. The ensemble is evaluated under `no_grad`, so computing $\hat{E}$ does not itself contribute gradients to ϕ; it is used purely as a scoring function. Each transition is then assigned a priority according to $p_i \propto \hat{E}(s_i,a_i)$. Transitions with higher uncertainty are sampled more frequently when constructing contrastive batches. Following PER (Schaul et al., 2015), we convert priorities into sampling probabilities, $P(i) = p_i^\alpha / \sum_k p_k^\alpha$, where α controls the strength of prioritization, with $\alpha = 0$ recovering uniform sampling. See Alg. 1 in Appendix D for the pseudo-code.

Unlike standard PER, we do not apply importance-sampling correction. In PER, these weights correct the sampling distribution to preserve an unbiased estimate of the RL objective. In contrast, our non-uniform sampling of contrastive transitions is intentional: the contrastive objective is used to direct representation learning toward selected regions of the input distribution. We therefore treat the sampling shift as an intended inductive bias rather than a bias requiring correction.

General Algorithm and Design Choices. UCRL is agnostic to the underlying visual RL algorithm: it requires only (i) a shared visual encoder f_ϕ feeding both a RL objective and a contrastive objective, and (ii) some estimate of epistemic uncertainty over (s,a) pairs. Two flexibilities are left open by this general formulation:

- *Uncertainty estimator.* We use ensemble-critic disagreement throughout this work, but any epistemic uncertainty estimate compatible with the RL algorithm's (e.g., bootstrapped heads, MC-dropout) could substitute for it.
- *Interaction with the base algorithm's own sampling.* UCRL is compatible with RL algorithms using either uniform sampling or prioritized replay. In all cases, UCRL preserves the original sampling strategy of the RL branch and applies its uncertainty-based prioritization independently to the contrastive branch. Thus, UCRL does not modify the sampling or optimization procedure of the underlying RL algorithm.

Instantiations. We instantiate UCRL on CURL (Laskin et al., 2020) in two settings that differ in RL algorithm and replay structure. On DeepMind Control, CURL is built on SAC (Haarnoja et al., 2018); we extend its critics to an ensemble used only to compute $\hat{E}(s, a)$ (Alg. 1). On Atari 100k, CURL is built on DQN with TD-error-based prioritized replay (Schaul et al., 2015); we keep this prioritization for the RL batch and maintain a separate uncertainty priority, from a critic ensemble, for the contrastive batch. In both cases, the RL branch retains its native sampling, and only the contrastive batch is prioritized using UCRL. Implementation details are in Appendix D.

4 Theoretical Analysis

This section provides a theoretical basis for using epistemic uncertainty to prioritize transitions for representation learning in UCRL. We first establish a two-sided bound relating the encoder gradient induced by the ensemble critic objective to epistemic uncertainty and Bellman error (Lemma 2). A direct consequence is that epistemic uncertainty alone lower-bounds encoder gradient magnitude, independently of Bellman error (Corollary 1). We then use this result to distinguish epistemic-uncertainty-driven, Bellman-driven, and saturated transitions, and further characterize the epistemic-uncertainty-driven transitions in terms of representation learning potential. In particular, we argue that epistemic uncertainty provides a targeted signal for transitions with non-negligible encoder activity, whereas TD-error prioritization also responds to value-estimation error. We then discuss the implications of these types for transition sampling and motivate uncertainty-based prioritization as a means of allocating representation-learning updates toward transitions with non-negligible encoder activity.

Notation. Let $z = f_\phi(s) \in \mathbb{R}^d$ denote the latent representation produced by the shared encoder, and let $Q_i(\cdot, a) : \mathbb{R}^d \to \mathbb{R}$ denote the i-th critic head. Define

$$J_\phi(s) := \frac{\partial f_\phi(s)}{\partial \phi} \in \mathbb{R}^{d\times p}, \qquad \Theta_\phi(s) := J_\phi(s) J_\phi(s)^\top \in \mathbb{R}^{d\times d},$$

where d is the dimension of the representation and $p = \dim(\phi)$ is the number of trainable encoder parameters, with $p \gg d$. Following the neural tangent kernel interpretation (Jacot et al., 2018), $\Theta_\phi(s)$ characterizes the latent-space directions accessible through encoder parameter updates. By the chain rule,

$$\nabla_\phi Q_i(z, a) = J_\phi(s)^\top \nabla_z Q_i(z, a),$$

so the encoder gradient depends on both the critic sensitivity $\nabla_z Q_i$ and the encoder Jacobian J_ϕ.

Assumption 1 (Local sensitivity bounds)**.** *At training step t, let ϕ_t denote the encoder parameters and $\mathcal{D}_t$ the replay buffer. We assume that there exist finite, strictly positive constants $L_f(t)$, $U_f(t)$, $\lambda_0(t)$, and $\Lambda_0(t)$, with $0 < L_f(t) \le U_f(t) < \infty$ and $0 < \lambda_0(t) \le \Lambda_0(t) < \infty$, such that, for every transition $(s, a) \in \mathrm{supp}(\mathcal{D}_t)$ and every critic head $i \in 1, \dots, N$, the following conditions hold:*

$$\textit{(i) Critic sensitivity:} \quad L_f(t) \;\le\; \left\|\nabla_z Q_i(z, a)\right\|_{z=f_{\phi_t}(s)} \;\le\; U_f(t), \tag{1}$$

$$\textit{(ii) Encoder conditioning:} \quad \lambda_0(t) \;\le\; \lambda_{\min}\big(\Theta_{\phi_t}(s)\big) \;\le\; \lambda_{\max}\big(\Theta_{\phi_t}(s)\big) \;\le\; \Lambda_0(t), \tag{2}$$

where $\Theta_\phi(s) = J_\phi(s) J_\phi(s)^\top$ and $\lambda_{\min}, \lambda_{\max}$ denote its smallest and largest eigenvalues. Here, $\mathrm{supp}(\mathcal{D}_t)$ denotes the set of state-action pairs contained in the replay buffer at step t.

Empirical validation and interpretation are available in Appendix G.

Lemma 1 (Encoder gradient transfer)**.** *Under Assumption 1, at step t, for every $(s,a) \in \mathrm{supp}(\mathcal{D}_t)$ and every head i,*

$$c_0(t) \;:=\; \sqrt{\lambda_0(t)}\, L_f(t) \;\;\le\;\; \|\nabla_\phi Q_i\| \;\;\le\;\; \sqrt{\Lambda_0(t)}\, U_f(t) \;=:\; C_0(t).$$

Proof available in Appendix H.2.

Consider the encoder gradient of the ensemble critic loss, $\nabla_\phi \mathcal{L}_Q = \frac{2}{N}\sum_{i=1}^{N}(Q_i - y)\,\nabla_\phi Q_i$. Its squared norm splits into two parts:

$$\|\nabla_\phi \mathcal{L}_Q\|^2 = \sum_{i=1}^{N}(Q_i - y)^2 \|\nabla_\phi Q_i\|^2 \;+\; \sum_{i\neq j}(Q_i - y)(Q_j - y)\,\big\langle \nabla_\phi Q_i,\, \nabla_\phi Q_j \big\rangle. \tag{3}$$

The first term reflects each critic's individual error. The second captures interactions between critics and can take either sign. If it were negative, critics with large errors could push the encoder in opposing directions and cancel each other's signal. We therefore assume that such cancellation is sufficiently limited in aggregate, as formalized by the following Assumption.

Assumption 2 (Non-negative cross-term sum)**.** *For all (s,a) encountered during training, and with y the target of the RL critic loss, the ensemble critic gradients satisfy*

$$\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N}(Q_i - y)(Q_j - y)\,\big\langle \nabla_\phi Q_i,\, \nabla_\phi Q_j \big\rangle \;\ge\; 0. \tag{4}$$

where $\langle ., . \rangle$ is the inner product. This is plausible because the critics share the encoder, architecture, objective, and target, and differ only in initialization, so their encoder gradients tend to align. We validate it empirically in Appendix H.1.

Lemma 2 (Two-sided encoder gradient bound)**.** *Under Assumptions 1 and 2, the encoder gradient magnitude induced by the ensemble critic objective admits the following lower and upper bounds in terms of the ensemble epistemic uncertainty and Bellman error:*

$$c(t)\,\sqrt{\hat{E}(s,a) + (\bar{Q} - y)^2} \;\;\le\;\; \|\nabla_\phi \mathcal{L}_Q\| \;\;\le\;\; C(t)\,\sqrt{\hat{E}(s,a) + (\bar{Q} - y)^2}, \tag{5}$$

where $\hat{E}(s,a) = \widehat{\mathrm{Var}}_i[Q_i(s,a)]$ denotes the ensemble epistemic uncertainty, $(\bar{Q} - y)^2$ is the squared Bellman error, and $c(t) = \frac{2c_0(t)}{\sqrt{N}}$, $C(t) = 2C_0(t)$.

Proof available in Appendix H.3.

Corollary 1 (Epistemic Lower Bound)**.** *Since both terms on the left-hand side of Eq. 5 are non-negative, dropping the Bellman-error term yields, for all t,*

$$c(t)\,\sqrt{\hat{E}(s,a)} \;\le\; \|\nabla_\phi \mathcal{L}_Q\|, \tag{6}$$

with $c(t)$ as in Lemma 2. This is strictly weaker than Eq. 5 but it isolates a gradient floor that depends on ensemble disagreement alone, independent of the current Bellman error.

From the bound to a prioritization rule. Under the stated assumptions, Corollary 1 shows that non-negligible epistemic uncertainty guarantees a non-vanishing encoder gradient. To interpret ensemble disagreement as a meaningful epistemic signal, we adopt the standard interpretation of deep ensembles that disagreement among independently initialized heads provides a practical measure of epistemic uncertainty (Lakshminarayanan et al., 2017; Osband et al., 2016; Kendall & Gal, 2017). Under this interpretation, disagreement indicates uncertainty in the value estimate associated with the current data and representation, rather than merely reflecting optimization noise.

Under this premise, Lemma 2 partitions transitions into three regimes: *epistemic-uncertainty-driven* (non-negligible $\hat{E}(s,a)$), *Bellman-driven* (negligible $\hat{E}(s,a)$ but non-negligible Bellman error, so encoder gradients are driven by unconverged value estimation rather than representational ambiguity), and *saturated* (both negligible, with no meaningful encoder activity). Although $\hat{E}(s,a)$ measures disagreement among the ensemble critics, this disagreement itself may arise from both value-estimation

and representation uncertainty. To analyze the sources of ensemble disagreement, we consider the shared encoder parameters ϕ and critic parameters ψ as random variables and apply the law of total variance. This gives

$$\hat{E}(s,a) = \underbrace{\mathbb{E}_{\phi}\big[\mathrm{Var}_{\psi}\big(Q_{\psi}(f_{\phi}(s),a) \mid \phi\big)\big]}_{\text{value uncertainty}} + \underbrace{\mathrm{Var}_{\phi}\big(\mathbb{E}_{\psi}\big[Q_{\psi}(f_{\phi}(s),a) \mid \phi\big]\big)}_{\text{representation uncertainty}}. \tag{7}$$

Thus, we further distinguish epistemic-uncertainty-driven transitions into *representation-improvable* and *value-function-limited* cases. Under the conditions of Corollary 1, the *value-function-limited* case is ruled out: non-negligible epistemic uncertainty necessarily implies non-vanishing encoder activity. Thus, within our theoretical framework, epistemic-uncertainty-driven transitions are representation-improvable. Full derivation of the types and their implications for uniform and TD-error-based sampling is given in Appendix C.

Uniform sampling provides no mechanism to distinguish between transition types, and therefore allocates contrastive updates in proportion to their prevalence in the replay buffer. As training progresses, this can increasingly allocate updates to saturated or Bellman-driven transitions, where the remaining representation-learning signal is negligible or the dominant signal is associated with value-estimation error rather than representational uncertainty. TD-error prioritization does not resolve this: it can upweight Bellman-driven transitions just as readily as representation-improvable ones, despite their different implications for the encoder (See Appendix C). UCRL instead prioritizes by $\hat{E}(s,a)$ alone, shifting contrastive updates toward representation-improvable transitions without relying on their prevalence in the replay buffer or on value-estimation error.

5 Experiments

In this section, we evaluate UCRL-C on the DeepMind Control Suite and Atari Games under the standard 100k-interaction setting. On DeepMind Control Suite, we compare UCRL-C with established visual RL methods, while on Atari we focus on comparison with CURL to evaluate the transferability of uncertainty-driven transition selection to a different RL algorithm. We additionally evaluate the theoretical claims empirically and analyze performance, sample efficiency, and the contribution of key components through ablations. Implementation details and hyperparameters are provided in Appendix D.

5.1 Baselines

CURL (Laskin et al., 2020) is our primary baseline, as UCRL-C retains the CURL architecture and modifies only the transition-sampling strategy of its contrastive branch. We additionally consider **CURL+** and **CURL++**, which extend CURL with DrQ-style data augmentation. These methods enable controlled, same-family comparisons to isolate the effect of uncertainty-based transition selection. For broader evaluation on the DeepMind Control Suite, we include **Pixel-SAC** and **State-SAC** as basic pixel-based and state-based SAC references (Haarnoja et al., 2018), respectively. We further consider **DrQ** (Yarats et al., 2021b) and **DrQ-v2** (Yarats et al., 2021a), which leverage data augmentation, as well as **Dreamer** (Hafner et al., 2020), **SAC+AE** (Yarats et al., 2021c), **SLAC** (Lee et al., 2020), **SPR** (Schwarzer et al., 2021), and **PlayVirtual** (Yu et al., 2021), which employ model-based, representation-learning, or predictive approaches. All DeepMind Control experiments follow the standard 100K-interaction protocol introduced by Laskin et al. (2020), which evaluates sample-efficient visual control from pixel observations.

Table 1: **Effect of sampling strategy vs. augmentation. All methods share the CURL backbone.**

	UCRL-C	**CURL**	**CURL+**	**CURL++**
Sampling	Prioritized	Uniform	Uniform	Uniform
DrQ-style Aug.	None	None	[2;2]	[5;5]
Finger, Spin	**805** $\pm$ 85	767 $\pm$ 56	<u>780</u> $\pm$ 96	735 $\pm$ 120
Cartpole, Swingup	**715** $\pm$ 37	582 $\pm$ 146	<u>694</u> $\pm$ 87	665 $\pm$ 122
Reacher, Easy	**880** $\pm$ 64	538 $\pm$ 233	<u>541</u> $\pm$ 190	479 $\pm$ 216
Cheetah, Run	170 $\pm$ 31	<u>299</u> $\pm$ 48	**302** $\pm$ 50	264 $\pm$ 53
Walker, Walk	**555** $\pm$ 114	403 $\pm$ 24	484 $\pm$ 61	<u>504</u> $\pm$ 142
Ball-in-Cup, Catch	**952** $\pm$ 20	<u>769</u> $\pm$ 43	687 $\pm$ 260	728 $\pm$ 143

On Atari, we compare UCRL-C only with CURL. This comparison examines whether uncertainty-driven transition selection can be transferred to a different RL algorithm, beyond the original DeepMind Control setting.

Table 2: **Score achieved at the standard 100k-interaction budget against external baselines.**

	UCRL-C	CURL	DrQ	DrQ-v2	Dreamer	SAC+AE	SLAC	SPR	PlayVirtual	Pixel-SAC	State-SAC
Finger, spin	805	767	784	557	341	740	693	**840**	683	179	811
	± 85	± 56	± 173	± 325	± 70	± 64	± 141	± 143	± 189	± 66	± 46
Cartpole, swingup	715	582	675	**829**	326	311	-	815	812	416	835
	± 37	± 146	± 174	± 33	± 27	± 11		± 48	± 66	± 40	± 22
Reacher, easy	**880**	538	682	339	314	274	-	684	663	145	746
	± 64	± 233	± 86	± 108	± 155	± 14		± 186	± 214	± 30	± 25
Cheetah, run	170	299	332	478	235	267	319	452	**510**	197	616
	± 31	± 48	± 36	± 19	± 137	± 24	± 56	± 117	± 38	± 15	± 18
Walker, walk	**555**	403	492	351	277	394	361	397	499	42	891
	± 114	± 24	± 267	± 381	± 12	± 22	± 73	± 220	± 161	± 12	± 82
Ball in cup, catch	**952**	769	828	574	246	391	512	807	939	312	746
	± 20	± 43	± 131	± 437	± 174	± 82	± 110	± 165	± 20	± 63	± 91

Evaluation protocol. Our evaluation consists of two complementary comparisons. On DMControl, we compare UCRL-C with both CURL and its augmented variants for controlled, same-family comparisons, and a broader set of visual RL baselines to contextualize its performance across different approaches to representation learning and sample efficiency. On Atari, we focus exclusively on CURL to investigate the transferability of uncertainty-driven transition selection to a different RL algorithm. All experiments are averaged over 5 seeds.

5.2 Performance on DeepMind Control Suite and Atari at 100k Steps

Table 1, Table 2, and Figure 2 report performance on DeepMind Control suite at the standard 100k-interaction budget. The full Deepmind Control and Atari scores are reported in Tables 7 and 8 in Appendix I.

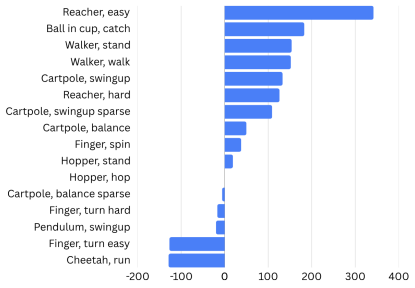


Figure 2: **Performance difference between UCRL-C and CURL across 16 DeepMind Control tasks.** Bars show mean return difference (UCRL-C − CURL), with positive values favoring UCRL-C.

Comparison with CURL and its variants. On the main DeepMind Control Suite tasks (Table 1), UCRL-C substantially improves over the CURL backbone, outperforming it on 5 of 6 tasks, with gains ranging from 5% on Finger, spin to 48% on Reacher, easy. UCRL-C is also competitive with the stronger same-family variants CURL+ and CURL++, outperforming both on 5 of 6 tasks. On the full DMControl tasks (Figure 2 and Table 7 in the Appendix), UCRL-C outperforms on 10 of 16 tasks, is comparable on 4, and underperforms on 2. On Atari 100k, UCRL-C outperforms CURL on 17 of 26 games, achieving a higher IQM of HNS (15.45% vs. 11.72%) and a smaller Optimality Gap (71.42% vs. 74.13%), though the bootstrapped confidence intervals overlap. UCRL's mean and median HNS (33.91% and 13.74%) also exceed CURL's (32.26% and 12.33%), with a mean per-game relative improvement of 14.81%. Together, these results indicate that uncertainty-based prioritization is not subsumed by DrQ-style augmentation and generalizes across environments and algorithms, with substantial gains on both DeepMind Control and Atari 100k.

Table 3: **Sample reduction of UCRL relative to CURL's 100k-interaction budget.** N_{UCRL} is the number of interactions UCRL needs to match CURL's mean performance at 100k interactions.

Task	N_{UCRL}	Sample Reduction (%)
Ball-in-Cup, Catch	44K	56
Walker, Walk	56K	44
Reacher, Easy	39K	61
Cartpole, Swingup	35K	65
Finger, Spin	94K	6
Mean	–	**46.4**

Comparison with broader visual RL methods. Despite its simple design, UCRL remains competitive with established visual RL methods (Table 2). It achieves the best reported performance on 3 tasks and is competitive on the other 2. On Cheetah, run, however, UCRL trails the strongest reported baselines, suggesting that the benefit of uncertainty-based prioritization is task-dependent. A possible explanation is that Cheetah, run is particularly difficult in this low-data regime, as reflected in the relatively low returns of the state-based agent (See State-SAC at Table 2). When value estimation itself is difficult, critic disagreement may become a less reliable signal, reducing its usefulness for prioritizing transitions for representation learning: UCRL may end up prioritizing transitions that are uncertain for value estimation but uninformative for representation learning. This remains a hypothesis, and we leave a deeper analysis to future work.

5.3 Sample Efficiency

To quantify the sample-efficiency gains of UCRL, we measure the number of environment interactions required to reach the mean performance achieved by CURL at 100k interactions. For each task, we identify the point at which the mean UCRL learning curve reaches this target, linearly interpolating between consecutive evaluation points. Of the six tasks, UCRL reaches the CURL target on five; it does not reach this target within the evaluation budget on Cheetah, run, consistent with the weaker relative performance discussed above. As shown in Table 3, across these five tasks UCRL requires between 35K and 94K interactions, corresponding to reductions of 6%–65%, and reaches CURL's 100k performance using 46.4% fewer environment interactions on average.

We note that UCRL also uses a smaller replay batch size than CURL (256 vs. 512 transitions per iteration), so the reduction in total replay samples processed is correspondingly larger than the reduction in environment interactions; we report interaction counts here as the primary efficiency metric, since it is agnostic to this implementation detail.

5.4 Computation time

A direct wall-clock comparison between UCRL and CURL is complicated by the two methods' different computational graphs: CURL uses a single 512-sample batch for both the RL and contrastive losses, so the shared encoder and each branch process 512 samples per iteration. UCRL instead draws two 256-sample batches, so the shared encoder still processes 512 samples in total, but each branch processes only 256. We therefore treat the total number of samples drawn from the replay buffer, rather than the per-branch batch size, as the fair basis for comparison; under this matched setting, UCRL takes 1.19 hours on average versus 2.01 hours for CURL, a 41% reduction. Beyond this matched setting, UCRL's improved sample efficiency (Section 5.3) lets us further reduce the per-branch batch size to 128 without sacrificing performance, more than offsetting the prioritization overhead. As shown in Figure 3, UCRL completes 100k steps in 0.6 hours with batch size 128, versus 2.01 hours for CURL, a 70% reduction in wall-clock time.

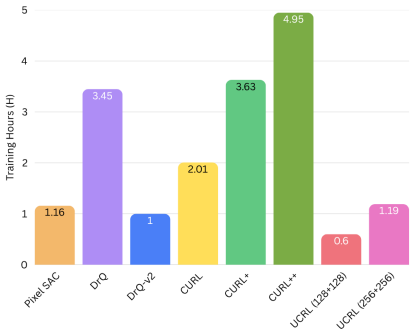


Figure 3: **Average training time over 5 tasks.** Baselines use one 512-sized batch; UCRL uses two batches (uniform + prioritized) of matched size.

5.5 Ablation Study

Table 4 isolates the contributions of contrastive learning, transition prioritization, and the additional ensemble value functions. Removing contrastive learning substantially degrades performance, highlighting its importance for representation learning. Adding transition prioritization further improves performance, with ensemble-based prioritization outperforming TD-error prioritization under the matched batch configuration. These results support the use of epistemic uncertainty as a more informative signal for transition selection than value-estimation error alone.

Table 4: **Ablation study of contrastive learning, transition prioritization, and ensemble gradient flow.** Batch configuration denotes uniformly sampled vs. prioritized transitions. Ensemble (SG) stops gradients from the additional critics to the shared encoder. Ensemble uses a matched total batch size of 512. Scores are averaged across tasks.

CL Loss	Prioritization	Batch	#Critics	Avg. Score
✗	✗	512	2	215.17
✓	✗	512	2	559.67
✓	TD	256+256	2	581.47
✓	Ensemble	256+256	2	615.97
✓	Ensemble (SG)	256+256	5	616.47
✓	Ensemble	256+256	5	632.54
✓	Ensemble	128+128	5	**664.29**

The stop-gradient (SG) ablation further separates the contribution of the additional ensemble critics from that of the uncertainty-based prioritization signal. Although stopping gradients from the additional critics to the shared encoder leads to only a modest performance decrease, most of the improvement is retained. This suggests that the primary benefit of the additional critics comes from the uncertainty signal used for prioritization rather than from their direct contribution to representation learning. Finally, reducing the batch size to 128+128 yields the highest average score, despite smaller batches typically introducing noisier gradients under uniform sampling. We hypothesize this reflects a distinct effect of prioritization: a smaller batch is more concentrated toward high-uncertainty transitions, and this increased selection pressure can outweigh the added gradient noise. This suggests

the benefits of uncertainty-driven prioritization are not merely additive with batch size, and may partially offset the instability usually associated with smaller batches.

5.6 Empirical Validation of the Theoretical Analysis

We empirically validate the bounds of Section 4 on replay-buffer transitions (Appendix E). Since the constants depend on quantities that cannot be computed over the full buffer, we set the lower and upper constants to the 5th and 95th percentiles of the ratio $\|\nabla_\phi \mathcal{L}_Q\|^2/(\hat{E}(s,a) + (\bar{Q} - y)^2)$ on a calibration split. This is robust to outliers, unlike the sample min/max, and targets 95% coverage by construction. We then test whether this coverage holds on a held-out split. For Lemma 2, held-out coverage stays near 95% throughout training (93.08–94.84% lower, 93.83–96.02% upper across 10k/50k/100k), and the bound tightens as $\hat{C}_0/\hat{c}_0$ falls from 13.2 at 10k to 2.7–3.8 from 60k onward. Using $\hat{E}(s,a)$ alone (Corollary 1) yields comparable coverage (94.92–95.31% lower, 94.30–96.40% upper), showing that epistemic uncertainty alone remains strongly associated with encoder gradient magnitude even near convergence.

5.7 Sensitivity Analysis

We sweep the prioritization coefficient α and ensemble size N on Ball-in-Cup, Catch (Appendix I.1). Performance is stable across most of the grid, with our default configuration (N=5, α=0.7) near the top of this range; degradation is concentrated at large ensembles (N=10), which show lower mean return and higher variance, suggesting oversized ensembles yield a noisier uncertainty estimate rather than a better one.

5.8 Gradient Interaction Between the RL and Contrastive Branches

To examine how epistemic uncertainty relates to the interaction between the RL and contrastive objectives, we measure the cosine similarity between $\nabla_\phi \mathcal{L}_C(s)$ and $\nabla_\phi \mathcal{L}_Q(s,a)$ across training (Appendix F). The population-level cosine similarity stays close to zero throughout training for both sampling schemes, but uncertainty-prioritized transitions retain heavier-tailed, more dispersed interaction than uniformly sampled ones by late training, and this dispersion concentrates specifically among high-uncertainty transitions (Figure 10 in Appendix F). This supports our theoretical picture: uncertainty-based sampling preferentially selects transitions that remain representation-improvable rather than ones whose encoder activity has saturated.

6 Conclusion

We presented UCRL, which prioritizes transitions for the contrastive branch of visual RL methods using epistemic uncertainty from a critic ensemble, while leaving the RL objective and its sampling distribution unchanged. We showed that epistemic uncertainty lower-bounds the critic-induced encoder gradient, distinguishing it from TD-error prioritization, which conflates epistemic uncertainty with Bellman error. We instantiated UCRL on CURL, yielding UCRL-C, which outperforms CURL on 10 of 16 DeepMind Control tasks, with a 9.5% mean relative gain, and on 17 of 26 Atari 100k games, raising IQM HNS from 11.72% to 15.45%. On five of the six main DMControl tasks, UCRL-C reaches CURL's 100k performance with 46.4% fewer interactions, and it reduces wall-clock time by 41% at a matched batch size. Ablations show that uncertainty-based prioritization outperforms TD-based prioritization and that the gains stem from the uncertainty signal rather than the additional critics.

UCRL underperforms on Cheetah, run, suggesting that critic disagreement is a less reliable signal when value estimation itself is difficult. Several directions remain open. Our lower bound discards the cross-head gradient terms; although Assumption 2 holds empirically (Appendix H.1), incorporating these terms could yield a tighter bound. Since UCRL only modifies the contrastive sampling distribution, it could be applied to other auxiliary objectives, such as reconstruction or self-predictive losses. Finally, cheaper uncertainty estimators could reduce the overhead of the critic ensemble.

AI USE STATEMENT

We used large language models (LLMs) for paper retrieval and discovery, as well as for editing support (clarity/grammar) and minor non-core code cleanup (formatting, docstrings, renaming, etc.). All LLM-assisted outputs and edits were reviewed by the authors, who remain fully responsible under the ICLR 2027 policy.

REFERENCES

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297, 2021.

Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.

Rodrigo Antonio Carrasco-Davis, Sebastian Lee, Claudia Clopath, and Will Dabney. Uncertainty prioritized experience replay. In *Reinforcement Learning Conference*, 2025. URL https://openreview.net/forum?id=S4coEjOWBG.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

Dogan C Cicek, Enes Duran, Baturay Saglam, Furkan B Mutlu, and Suleyman S Kozat. Off-policy correction for deep deterministic policy gradient algorithms via batch prioritized experience replay. In *2021 IEEE 33rd international conference on tools with artificial intelligence (ICTAI)*, pp. 1255–1262. IEEE, 2021.

Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

Scott Fujimoto, David Meger, and Doina Precup. An equivalence between loss functions and non-uniform sampling in experience replay. *Advances in neural information processing systems*, 33: 14219–14230, 2020.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

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.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pp. 1735–1742. IEEE, 2006.

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

Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep hierarchical planning from pixels. *Advances in Neural Information Processing Systems*, 35:26091–26104, 2022.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy P. Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, April 2025. doi: 10.1038/s41586-025-08744-2. URL https://doi.org/10.1038/s41586-025-08744-2.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=YevsQ05DEN7.

Łukasz Kaiser, Mohammad Babaeizadeh, Piotr Miłos, Błażej Osiński, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for atari. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1xCPJHtDB.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Victor A Kich, Jair A Bottega, Raul Steinmetz, Ricardo B Grando, Ayano Yorozu, and Akihisa Ohya. Curling the dream: Contrastive representations for world modeling in reinforcement learning. In *2024 24th International Conference on Control, Automation and Systems (ICCAS)*, pp. 952–957. IEEE, 2024.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

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

Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in neural information processing systems*, 33:741–752, 2020.

Changling Li, Zhang-Wei Hong, Pulkit Agrawal, Divyansh Garg, and Joni Pajarinen. ROER: Regularized optimal experience replay. *Reinforcement Learning Journal*, 4:1598–1618, 2024.

Jiajian Li, Qi Wang, Yunbo Wang, Xin Jin, Yang Li, Wenjun Zeng, and Xiaokang Yang. Open-world reinforcement learning over long short-term imagination. In *The Thirteenth International Conference on Learning Representations*, 2025.

Changpu Meng, Jie Yang, Wanqing Li, and Yi Guo. Snaphard contrast learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *International conference on machine learning*, pp. 16828–16847. PMLR, 2022.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

Minting Pan, Xiangming Zhu, Yunbo Wang, and Xiaokang Yang. Iso-dream: Isolating and leveraging noncontrollable visual dynamics in world models. *Advances in neural information processing systems*, 35:23178–23191, 2022.

Rudra PK Poudel, Harit Pandya, Stephan Liwicki, and Roberto Cipolla. Recore: Regularized contrastive representation learning of world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22904–22913, 2024.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=CR1XOQ0UTh-.

Baturay Saglam, Furkan B Mutlu, Dogan C Cicek, and Suleyman S Kozat. Actor prioritized experience replay. *Journal of Artificial Intelligence Research*, 78:639–672, 2023.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uCQfPZwRaUu.

Shivakanth Sujit, Somjit Nath, Pedro Braga, and Samira Ebrahimi Kahou. Prioritizing samples in reinforcement learning with reducible loss. *Advances in Neural Information Processing Systems*, 36:23237–23258, 2023.

Chenyu Sun, Hangwei Qian, and Chunyan Miao. Cclf: A contrastive-curiosity-driven learning framework for sample-efficient reinforcement learning. *arXiv preprint arXiv:2205.00943*, 2022.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021a.

Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=GY6-6sTvGaf.

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the aaai conference on artificial intelligence*, volume 35, pp. 10674–10681, 2021c.

Arda Sarp Yenicesu, Furkan B Mutlu, Suleyman S Kozat, and Ozgur S Oguz. Cuer: Corrected uniform experience replay for off-policy continuous deep reinforcement learning algorithms. *arXiv preprint arXiv:2406.09030*, 2024.

Tao Yu, Cuiling Lan, Wenjun Zeng, Mingxiao Feng, Zhizheng Zhang, and Zhibo Chen. Playvirtual: Augmenting cycle-consistent virtual trajectories for reinforcement learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=GSHFVNejxs7.

Yang Yue, Bingyi Kang, Zhongwen Xu, Gao Huang, and Shuicheng Yan. Value-consistent representation learning for data-efficient reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11069–11077, 2023.

Amirhossein Zhalehmehrabi, Tiziano Tezze, Alberto Castellini, and Alessandro Farinelli. Robust scene transfer for pointgoal navigation via privileged-sensor–guided contrastive learning. *IEEE Robotics and Automation Letters*, 11(10):11912–11919, 2026. doi: 10.1109/LRA.2026.3728340.

Lixuan Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Prelar: World model pre-training with learnable action representation. In *European Conference on Computer Vision*, pp. 185–201. Springer, 2024.

Weipu Zhang, Adam Jelley, Trevor McInroe, Amos Storkey, and Gang Wang. Object-centric world models from few-shot annotations for sample-efficient reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=qmEyJadwHA.

Appendix for

Not All Transitions Are Equal: Uncertainty-Driven Contrastive Learning for Visual RL

Appendix organization:

A Extended Related Works **13**

B Preliminaries **14**

B.1 Reinforcement Learning 14

B.2 Contrastive Learning 14

B.3 Uncertainty Decomposition 14

C Transition Types: Full Derivation **14**

D Implementation Details **16**

D.1 DeepMind Control Suite 16

D.2 Atari Games 18

E Empirical Validation of Theoretical Bounds **18**

F Gradient Interaction Between RL and Contrastive Branches: Full Statistics **21**

G Empirical Validation and Interpretation of Assumption 1 **24**

G.1 Assumption 1(i): Critic Sensitivity 24

G.2 Assumption 1(ii): Encoder Conditioning 24

H Theoretical Analysis **26**

H.1 Interpretation of Assumption 2 26

H.2 Proof of Lemma 1 27

H.3 Proof of Lemma 2 27

I Extended Results **29**

I.1 Sensitivity Analysis 29

A Extended Related Works

Visual Representation Learning in RL. Learning effective representations from high-dimensional observations is a central challenge in visual RL. Two broad strategies have been used to address this. Data-augmentation methods such as DrQ (Yarats et al., 2021b) and DrQ-v2 (Yarats et al., 2021a) regularize the RL objective directly by computing it over augmented views of the same observation, without introducing a separate representation-learning loss. A complementary line of work instead introduces auxiliary representation-learning objectives trained alongside the RL loss: SAC+AE (Yarats et al., 2021c) and SPR (Schwarzer et al., 2021) use reconstruction-based and self-predictive objectives respectively, while contrastive methods such as CURL (Laskin et al., 2020) learn representations by encouraging agreement between augmented views of the same observation and contrasting them against other samples in the minibatch. Latent dynamics methods, such as SLAC (Lee et al., 2020) and Dreamer (Hafner et al., 2020), further improve representation learning by learning compact models of the environment's dynamics in a latent space. These approaches primarily focus on the representation-learning objective or latent dynamics, while replay transitions are generally sampled without explicitly considering their value for representation learning. In

contrast, UCRL focuses on the selection of replay transitions used for representation learning, without modifying the underlying RL objective or representation-learning architecture.

Prioritized Replay and Uncertainty-Based Sampling. PER (Schaul et al., 2015) allocates more updates to transitions with large TD errors, improving learning efficiency by focusing on samples with larger value-function errors. Several subsequent works have refined TD-error-based replay to address limitations of the original PER formulation. LA3P (Saglam et al., 2023), for example, combines uniformly sampled transitions with low- and high-TD-error transitions for continuous control, while CUER (Yenicesu et al., 2024) adjusts sampling probabilities to account for the changing importance and age of replayed transitions. However, TD error itself is not necessarily a reliable measure of how informative a transition is for learning, as it can remain high due to irreducible stochasticity. Recent analyses (Sujit et al., 2023; Carrasco-Davis et al., 2025) show that TD-error-based sampling can consequently over-prioritize noisy transitions, motivating the use of uncertainty-based scores that better distinguish epistemic uncertainty from irreducible stochasticity. UPER (Carrasco-Davis et al., 2025), for example, uses epistemic uncertainty to prioritize transitions while reducing emphasis on errors caused by aleatoric uncertainty. Unlike these approaches, which modify the sampling distribution for the RL objective itself, UCRL applies uncertainty-based prioritization specifically to the representation-learning objective. It retains uniform sampling for the underlying RL objective and uses epistemic uncertainty only to construct contrastive learning batches, treating transition selection as a representation-learning problem and prioritizing transitions according to their representation plasticity — i.e., regions where the encoder's representation has not yet stabilized.

B Preliminaries

B.1 Reinforcement Learning

We consider a Markov Decision Process $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $P(s' \mid s, a)$ the transition dynamics, $r(s, a)$ the reward function, and $\gamma \in [0, 1)$ the discount factor. At each step, the agent selects $a_t \sim \pi(\cdot \mid s_t)$ and transitions according to $s_{t+1} \sim P(\cdot \mid s_t, a_t)$. The policy is trained to maximize the expected discounted return $\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$.

B.2 Contrastive Learning

Contrastive learning learns representations by encouraging agreement between related samples while distinguishing them from unrelated samples Hadsell et al. (2006); Oord et al. (2018); He et al. (2020); Chen et al. (2020). Given an observation o, two augmented views are encoded into an anchor representation q and a positive representation k^+, respectively. Representations $\{k_i\}_{i=1}^K$ obtained from other observations in the minibatch serve as negative samples. The InfoNCE objective is then

$$\mathcal{L}_c = -\log \frac{\exp(q^\top W k^+)}{\exp(q^\top W k^+) + \sum_{i=1}^K \exp(q^\top W k_i)}, \tag{8}$$

where W is a learnable bilinear similarity matrix.

B.3 Uncertainty Decomposition

Predictive uncertainty can be decomposed into aleatoric uncertainty, arising from irreducible stochasticity, and epistemic uncertainty, arising from limited knowledge and reducible with additional data Kendall & Gal (2017); Der Kiureghian & Ditlevsen (2009); Abdar et al. (2021); Gal & Ghahramani (2016); Lakshminarayanan et al. (2017). We denote the total uncertainty by

$$\hat{U}(s, a) = \hat{E}(s, a) + \hat{A}(s, a), \tag{9}$$

where $\hat{E}(s, a)$ and $\hat{A}(s, a)$ denote the epistemic and aleatoric components, respectively.

C Transition Types: Full Derivation

Transition types and implications for prioritization. The theoretical results characterize the encoder activity induced by a transition in terms of two distinct signals: epistemic uncertainty and

Bellman error. Since UCRL applies contrastive updates to the shared encoder, our objective is to allocate these updates toward transitions for which the representation remains improvable and can still benefit from further learning. We therefore distinguish transitions according to the source and magnitude of their encoder activity.

Why disagreement provides a useful epistemic signal. Our interpretation of epistemic uncertainty follows the standard deep-ensemble view that independently initialized heads trained on shared representations provide diverse estimates of the value function and that their disagreement serves as a practical measure of epistemic uncertainty (Lakshminarayanan et al., 2017; Osband et al., 2016; Kendall & Gal, 2017). Under this interpretation, disagreement indicates that the current data and representation do not yet determine a consistent value estimate. In contrast, when the heads agree but their predictions differ from the bootstrapped target, the resulting Bellman error primarily reflects unconverged value estimation rather than disagreement about the value induced by the current representation.

Lemma 2 formalizes this distinction by relating encoder-gradient magnitude to the sum of epistemic uncertainty and squared Bellman error. This yields three primary transition regimes.

1. **Epistemic-uncertainty-driven:** These transitions have non-negligible epistemic uncertainty, $\hat{E}(s, a)$, and therefore, by Corollary 1, necessarily exhibit non-vanishing encoder activity under the stated assumptions. We further distinguish two theoretical subclasses based on the decomposition in Eq. 7:
    - **Representation-improvable:** The transition exhibits non-negligible representation uncertainty (See Eq. 7) and non-negligible encoder activity. These transitions correspond to regions where the current representation remains improvable and can still undergo corrective updates.
    - **Value-function-limited:** The transition exhibits non-negligible epistemic uncertainty but negligible encoder activity. This case would correspond to disagreement that is attributable primarily to value estimation rather than to uncertainty that can be acted upon through the shared representation. However, under Corollary 1, this case is ruled out within our theoretical framework: non-negligible epistemic uncertainty necessarily implies non-vanishing encoder activity.
2. **Bellman-driven:** These transitions have negligible epistemic uncertainty but non-negligible Bellman error. Their encoder activity is therefore associated primarily with unconverged value estimation rather than disagreement among the critics. Such transitions may still induce encoder updates, but these updates are not associated with unresolved epistemic uncertainty in the representation.
3. **Saturated:** These transitions have both negligible epistemic uncertainty and negligible Bellman error. By Lemma 2, their induced encoder activity is consequently negligible.

Thus, within our theoretical framework, the epistemic-uncertainty-driven regime coincides with transitions exhibiting representation improvable, while Bellman-driven transitions correspond to encoder activity associated primarily with value-estimation error. Saturated transitions provide little critic-induced encoder activity.

Implications for uniform sampling. Uniform sampling does not distinguish between these regimes and therefore allocates contrastive updates independently of their current learning signal. As training progresses and the replay buffer accumulates transitions for which the critics already agree, a larger fraction of sampled transitions may become saturated or Bellman-driven. Contrastive updates on saturated transitions provide little critic-induced encoder activity, while updates on Bellman-driven transitions are associated primarily with value-estimation error rather than epistemic uncertainty.

In contrast, UCRL prioritizes transitions according to $\hat{E}(s, a)$ and therefore shifts the contrastive sampling distribution toward epistemic-uncertainty-driven transitions. Under the theoretical framework above, these transitions are representation-improvable, so UCRL allocates more contrastive updates to transitions where the shared encoder remains active.

Algorithm 1 UCRL

1: Initialize replay buffer $\mathcal{D}$
2: Initialize SAC (actor π_θ, critic Q_ψ) and encoder f_ϕ
3: **for** each environment step **do**
4: Observe transition (s, a, r, s')
5: Store (s, a, r, s') in $\mathcal{D}$ with initial priority p_0
6: Sample batch $\mathcal{B}_{RL} \sim \mathcal{D}$ (uniform)
7: Update SAC using $\mathcal{B}_{RL}$
8: Sample Contrastive batch $\mathcal{B}_C \sim \mathcal{D}$ using prioritized sampling $P(i) \propto p_i^\alpha$
9: Compute contrastive loss $\mathcal{L}_c$ on $\mathcal{B}_C$
10: Update encoder f_ϕ
11: Update priorities for $i \in \mathcal{B}_C$: $p_i \leftarrow \mathrm{Var}_k(Q_{\psi_k}(s_i, a_i)) + \epsilon$
12: **end for**

Implications for TD-error-based prioritization. The two-sided bound also clarifies why TD-error prioritization does not isolate the signal targeted by UCRL. For a single critic, $\hat{E}(s, a) = 0$, and Lemma 2 reduces to

$$c(t)|\delta(s, a)| \leq \|\nabla_\phi \mathcal{L}_Q\| \leq C(t)|\delta(s, a)|,$$

where $\delta(s, a) = Q(s, a) - y$ denotes the TD error. Thus, for a single critic, encoder-gradient magnitude and TD error are related up to step-dependent multiplicative constants.

For an ensemble, the corresponding quantity is

$$\frac{1}{N}\sum_{i=1}^{N}(Q_i - y)^2 = \hat{E}(s, a) + (\bar{Q} - y)^2.$$

Hence, ensemble TD-error-based prioritization combines epistemic uncertainty with Bellman error rather than isolating epistemic uncertainty.

This distinction is important for representation learning. Consider a Bellman-driven transition for which $\hat{E}(s, a) \approx 0$ but $(\bar{Q} - y)^2$ is non-negligible. Such a transition can induce a substantial encoder gradient according to Lemma 2 and can therefore receive high priority under TD-error-based sampling. However, because the epistemic uncertainty is negligible, the critics largely agree on the transition. The resulting encoder activity is therefore associated primarily with value-estimation error rather than unresolved epistemic uncertainty in the representation.

UCRL instead prioritizes exclusively according to $\hat{E}(s, a)$. Consequently, transitions with negligible epistemic uncertainty receive low priority regardless of their Bellman error. This suppresses the sampling of both saturated and Bellman-driven transitions in the contrastive branch while increasing the relative sampling probability of epistemic-uncertainty-driven transitions, which are representation-plastic under our theoretical assumptions.

D Implementation Details

UCRL is implemented as a modular extension of CURL Laskin et al. (2020). The method modifies only the transition sampling distribution used to construct contrastive learning batches, while the underlying RL objective and its sampling distribution remain unchanged. A step-by-step summary of UCRL-C is provided in Alg. 1 with its core contributions highlighted in blue. The implementation details for the two evaluation settings are described separately below.

D.1 DeepMind Control Suite

UCRL uses a shared visual encoder for contrastive representation learning and a Soft Actor-Critic (SAC) agent Haarnoja et al. (2018). We evaluate UCRL on the DeepMind Control Suite Tassa et al. (2018) for 100K environment steps, using the same set of tasks as the main CURL results to enable a fair comparison.

We extend the standard two-critic SAC architecture with three additional critics, resulting in an ensemble of five Q-functions, following the approach of An et al. (2021). All five critics are trained using the same SAC critic objective and backpropagate through the shared visual encoder. The variance of the five critics' value predictions is used as an estimate of epistemic uncertainty and to prioritize transitions for the contrastive learning objective. Importantly, this prioritization is applied only to the contrastive learning branch; RL updates continue to use uniformly sampled transitions, and the underlying SAC optimization remains unchanged.

We implement prioritized replay following Schaul et al. (2015) in parallel with standard uniform replay sampling, such that prioritized sampling affects only contrastive learning updates. All hyperparameters are reported in Table 5. Experiments are conducted on a machine equipped with an NVIDIA GeForce RTX 4070 Ti GPU.

For the baselines, we use results reported in prior work under the same DeepMind Control Suite 100K setting. Specifically, results for CURL, State-SAC, and Pixel-SAC are taken from Laskin et al. (2020); results for DrQ, CURL+, and CURL++ are taken from Sun et al. (2022); and results for Dreamer, SAC+AE, SLAC, SPR, and PlayVirtual are taken from Yue et al. (2023). For DrQ-v2 Yarats et al. (2021a), we use the authors' official implementation to evaluate the method at the 100K interaction budget, as these results were not reported in the original paper.

Table 5: Hyperparameters of UCRL for DeepMind Control Suite

Hyperparameter	**Value**
Random crop	True
Observation rendering	(100, 100)
Observation downsampling	(84, 84)
Replay buffer size	100000
Initial steps	1000
Stacked frames	3
Action repeat	2 finger, spin; walker, walk 8 cartpole, swingup 4 otherwise
Hidden units (MLP)	1024
Evaluation episodes	10
Optimizer	Adam
$(\beta_1, \beta_2) \rightarrow (f_\theta, \pi_\psi, Q_\phi)$	(.9, .999)
$(\beta_1, \beta_2) \rightarrow (\alpha)$	(.5, .999)
Learning rate $(f_\theta, \pi_\psi, Q_\phi)$	$(1.4e{-}4, 9.7e{-}4, 2e{-}4)$ cheetah, run $1e{-}3$ otherwise
Learning rate (α)	$1e{-}4$
Batch Size	128
CL Batch Size	128
Q function EMA τ	0.01
Critic target update freq	2
Convolutional layers	4
Number of filters	32
Non-linearity	ReLU
Encoder EMA τ	0.05
Latent dimension	50
Discount γ	.99
Initial temperature	0.1
α	0.7
β	0.5
N	5
Warmup steps	10000

D.2 ATARI GAMES

For Atari 100K, UCRL is implemented on top of the DQN-based CURL framework Laskin et al. (2020). We use the official CURL implementation to reproduce the CURL results under the standard Atari 100K setting. Unlike the DeepMind Control implementation, the DQN-based CURL replay buffer already uses prioritized sampling for both the RL and contrastive learning branches. We therefore duplicate the replay buffer into two separate buffers: one retains the original TD-error-based prioritization and is used exclusively for the RL updates, while the other uses the uncertainty-based priority and is used exclusively for the contrastive learning updates.

As in the DeepMind Control Suite experiments, epistemic uncertainty is used exclusively to prioritize transitions for the contrastive learning objective. The RL branch retains the original TD-error-based sampling of DQN-based CURL, while the contrastive branch samples transitions according to epistemic uncertainty. For the uncertainty estimate, we use the same ensemble configuration as in the DeepMind Control Suite experiments, consisting of five critics.

We evaluate UCRL under the standard Atari 100K protocol for 100K environment steps and compare its performance with the corresponding CURL results. All Atari-specific hyperparameters are reported in Table 6. Experiments are conducted on a machine equipped with an Intel Xeon processor running at 2.20–2.30 GHz with four CPU cores.

E EMPIRICAL VALIDATION OF THEORETICAL BOUNDS

We empirically evaluate the bounds established in Section 4 using held-out transitions from the replay buffer. At each evaluation step, we draw a fresh batch of transitions and split it evenly into calibration and held-out sets. For each transition i, we compute $r_i = \frac{||\nabla_\phi L_Q(s_i,a_i)||^2}{\hat{E}(s_i,a_i)+(\bar{Q}_i-y_i)^2}$. We estimate $\hat{a}$ as the 5th percentile of r_i on the calibration set. This choice provides a conservative lower-tail estimate of the ratio and targets approximately 95% empirical coverage on the calibration data. Symmetrically, we estimate an upper-bound constant $\hat{b}$ as the 95th percentile of r_i on the same calibration set, targeting approximately 95% one-sided coverage from above. Importantly, the held-out transitions are not used to determine $\hat{a}$ or $\hat{b}$, allowing us to test whether the fitted bounds generalize beyond the calibration data. We report the fraction of held-out transitions satisfying

$$||\nabla_\phi L_Q(s_i,a_i)||^2 \geq \hat{a}\cdot\big(\hat{E}(s_i,a_i)+(\bar{Q}_i-y_i)^2\big) \quad \text{and} \quad ||\nabla_\phi L_Q(s_i,a_i)||^2 \leq \hat{b}\cdot\big(\hat{E}(s_i,a_i)+(\bar{Q}_i-y_i)^2\big),$$

respectively. Thus, calibration coverage is approximately 95% by construction for both bounds, whereas held-out coverage provides the empirical test of each. The ratio $\hat{C}_0/\hat{c}_0$ recovered from these fitted constants (Appendix E, Figure 4) additionally lets us track how tightly the two-sided bound sandwiches the true encoder gradient over the course of training.

Lemma 1. Lemma 1 requires only that the bound constants satisfy $0 < c_0(t) \leq C_0(t) < \infty$. We recover this constant from the fitted $\hat{a}$ as $c_0(t) = \sqrt{\hat{a}N/4}$ and from $\hat{b}$ as $C_0(t) = \sqrt{\hat{b}}/2$. As shown in Figure 5, $c_0(t)$ remains strictly positive and $C_0(t)$ remains finite throughout training, with $c_0(t) \leq C_0(t)$ holding at every step, satisfying the requirements of Lemma 1.

Lemma 2. Using $\hat{E}(s,a) + (\bar{Q} - y)^2$, held-out coverage for the two-sided bound is strong at all three checkpoints (Figure 6). Averaged over five seeds, lower-bound coverage is 94.84%, 93.08%, and 94.77% at 10K, 50K, and 100K environment steps, respectively, while upper-bound coverage is 93.83%, 96.02%, and 94.61% at the same checkpoints. Both bounds remain above 90% throughout training and across seeds, with coverage staying stable even as training progresses and the Bellman-error contribution shrinks near convergence, providing empirical evidence that the sandwich in Lemma 2 holds tightly on both sides and is closely related to epistemic uncertainty in this regime.

Tightness of the bound over training. Beyond coverage, we examine how tightly the two-sided bound sandwiches the true encoder gradient by tracking the ratio of the fitted upper- and lower-bound constants, $\hat{C}_0/\hat{c}_0$, over training (Figure 4). This ratio starts large and variable across seeds, then contracts sharply and monotonically, stabilizing around 2.7–3.8 from 60K onward with markedly reduced seed variance. This indicates that while the bound holds throughout training (per the coverage results below), it is loose early in training and becomes substantially tighter as the encoder and critics

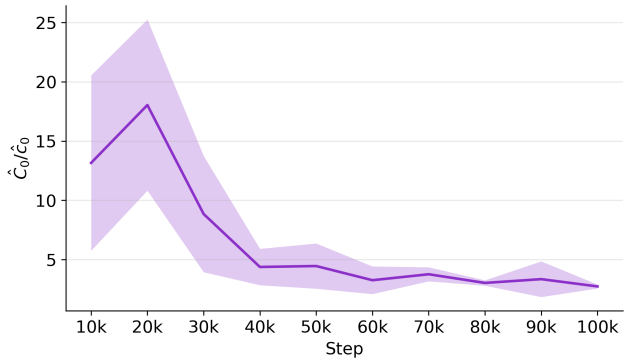


Figure 4: **Ratio of fitted bound constants $\hat{C}_0/\hat{c}_0$ over training.** Averaged over 5 seeds (shaded: std). The ratio starts large and variable early in training (mean 13.2 at 10k, peaking at 18.0 at 20k) and contracts sharply, stabilizing around 2.7–3.8 from 60k onward with markedly reduced seed variance, indicating the two-sided bound in Lemma 2 tightens substantially as training progresses.

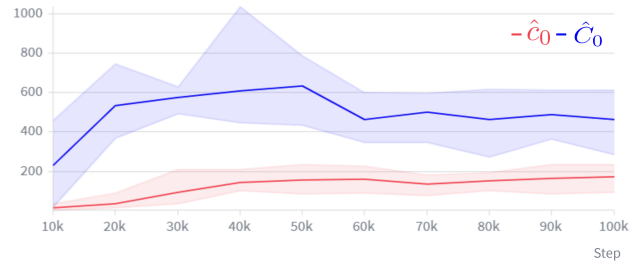


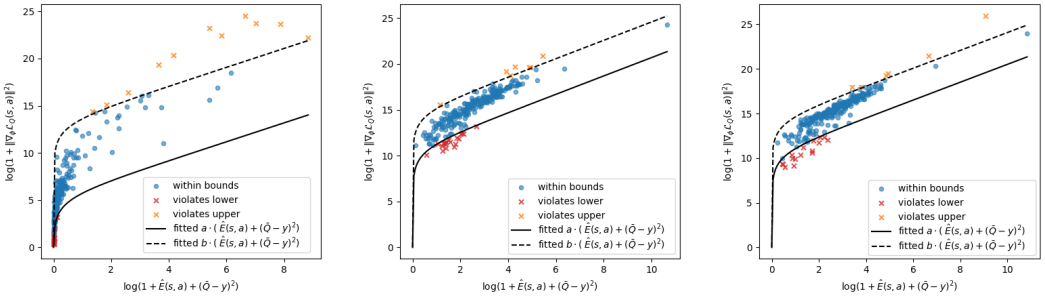
Figure 5: **Fitted bound constants $\hat{c}_0$ and $\hat{C}_0$ over training.** $\hat{c}_0 = \sqrt{\hat{a}N/4}$ and $\hat{C}_0 = \sqrt{\hat{b}}/2$ (Lemma 1), averaged over 5 seeds (shaded: std). Both remain finite and positive throughout training, with $c_0(t) \leq C_0(t)$ holding at every step and their gap narrowing over time (see Figure 4 for the ratio).

Table 6: Hyperparameters of UCRL for the Atari 100K benchmark

Hyperparameter	**Value**
Random crop	True
Image size	(84, 84)
Data Augmentation	Random Crop (Train)
Replay buffer size	100000
Training frames	400000
Training steps	100000
Frame skip	4
Stacked frames	4
Action repeat	4
Replay period every	1
Q network: channels	32, 64
Q network: filter size	5×5, 5×5
Q network: stride	5, 5
Q network: hidden units	256
Momentum (EMA for CURL) τ	0.001
Non-linearity	ReLU
Reward Clipping	$[-1, 1]$
Multi step return	20
Minimum replay size for sampling	1600
Max frames per episode	108K
Update	Distributional Double Q
Target Network Update Period	every 2000 updates
Support-of-Q-distribution	51 bins
Discount γ	0.99
Batch Size	32
Optimizer	Adam
Optimizer: learning rate	0.0001
Optimizer: β_1	0.9
Optimizer: β_2	0.999
Optimizer ϵ	0.000015
Max gradient norm	10
Exploration	Noisy Nets
Noisy nets parameter	0.1
α	0.5
β	$0.4 \rightarrow 1$
Hardware	CPU
N	5
ϵ	0.0 boxing 0.01 otherwise
Warmup steps	10000

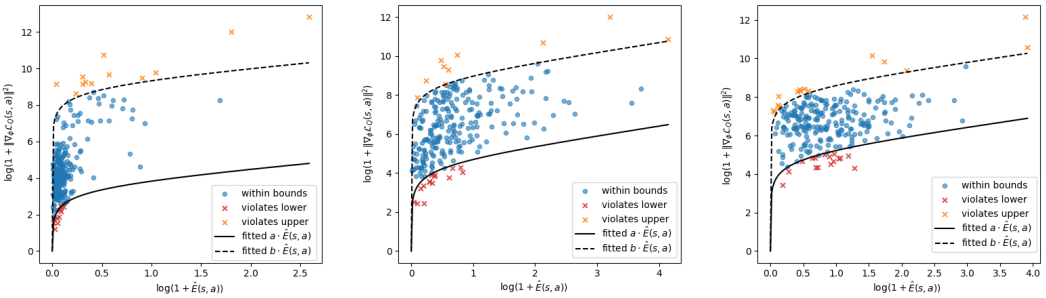
converge, consistent with Assumption 1 allowing the sensitivity and conditioning constants to vary with t.

Corollary 1. Dropping the Bellman-error term and using $\hat{E}(s, a)$, we test whether the sandwich still holds empirically on both sides using the epistemic term alone (Figure 7). Averaged over five seeds, lower-bound coverage is 95.31%, 95.23%, and 94.92% at 10K, 50K, and 100K steps, respectively, while upper-bound coverage is 96.40%, 94.92%, and 94.30% at the same checkpoints. Both bounds remain above 90% throughout training, closely tracking the coverage obtained with the full right-hand side in Lemma 2. This is consistent with Corollary 1, which follows from Lemma 2 by dropping the non-negative Bellman-error term: the comparable coverage on both sides indicates that epistemic uncertainty alone is nearly as informative as the full bound, even though Corollary 1 is derived only as a lower bound and provides no formal guarantee on the upper side.



(a) Step 10k (early) (b) Step 50k (mid) (c) Step 100k (convergence)

Figure 6: **Held-out validation of Lemma 2's gradient bound.** $\hat{E}(s,a)+(\bar{Q}-y)^2$ vs. $\|\nabla_\phi L_Q(s,a)\|^2$ (log-scaled), sampled uniformly from the replay buffer. $\hat{a}$ is fit on a calibration half and evaluated on the held-out half shown. Held-out coverage of lower-upper bounds: **94.84%-93.83%** (a, step 10k), **93.08%-96.02%** (b, step 50k), **94.77%-94.61%** (c, step 100k). Coverage stays high near convergence (c), where Bellman error has largely vanished, showing the bound holds on epistemic uncertainty alone.



(a) Step 10k (early) (b) Step 50k (mid) (c) Step 100k (convergence)

Figure 7: **Held-out validation of Corollary 1's gradient bound.** $\hat{E}(s,a)$ vs. $\|\nabla_\phi L_Q(s,a)\|^2$ (log-scaled), sampled uniformly from the replay buffer. $\hat{a}$ is fit on a calibration half and evaluated on the held-out half shown. Held-out coverage for lower-upper bounds: **95.31%-96.40%** (a, step 10k), **95.23%-94.92%** (b, step 50k), **94.92%-94.30%** (c, step 100k), consistent with Corollary 1.

F Gradient Interaction Between RL and Contrastive Branches: Full Statistics

To empirically assess whether the contrastive objective interferes with the RL objective at the shared encoder, we measure the cosine similarity between the encoder gradients induced by each loss,

$$\cos\left(\nabla_\phi \mathcal{L}_C(s), \nabla_\phi \mathcal{L}_Q(s,a)\right),$$

for 2048 samples from both uniformly sampled and uncertainty-prioritized transitions, at three stages of training (10k, 50k, and 100k steps) as shown in Figures 8 and 9 in Appendix F.

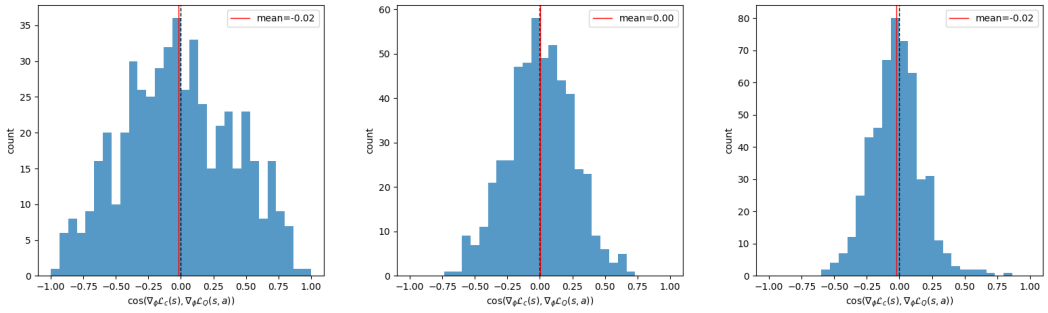
At early training (10k steps), the cosine similarity distribution is wide, with substantial mass at both strongly positive and strongly negative values (uniform: mean$=0.02$, frac$>0=0.46$; priority: mean$=-0.01$, frac$>0=0.50$). This indicates that, at the level of individual transitions, the two objectives frequently pull the encoder in either aligned or conflicting directions; the near-zero population mean reflects cancellation across transitions rather than the absence of gradient interaction.

As training progresses, the distributions narrow substantially. At 50k steps the histograms concentrate more tightly around zero (uniform: mean$=0.00$; priority: mean$=-0.02$), and by 100k steps both distributions are sharply peaked with most mass within ± 0.25 (uniform: mean$=-0.02$,

frac$> 0 = 0.44$; priority: mean$= -0.00$, frac$> 0 = 0.4$). Notably, at convergence the uniformly sampled distribution collapses almost entirely to a narrow spike, whereas the priority-sampled distribution retains visibly heavier tails. This suggests that transitions selected by our uncertainty-based prioritization continue to exhibit non-trivial gradient coupling with the RL objective even at late training, consistent with our decomposition (Section 4): prioritized transitions are disproportionately drawn from the representation-improvable regime, where the encoder has not yet stabilized, while the bulk of uniformly sampled transitions have entered the saturated regime and contribute near-orthogonal, low-magnitude gradients from either branch.

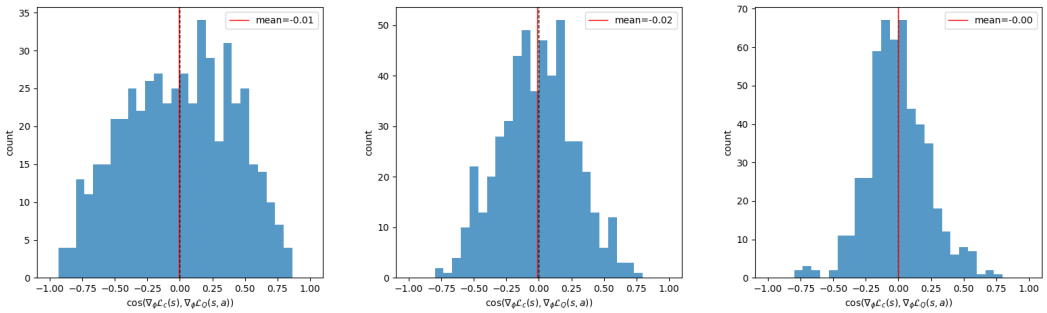
To examine this relationship directly, Figure 10 shows the joint density of cosine similarity and estimated epistemic uncertainty $\hat{E}(s, a)$ across training, from step 10k to step 100k in increments of 10k steps. Two patterns emerge. First, the low-uncertainty mass, which dominates the sample at all stages, does not narrow monotonically: at 10k steps the density is flat and cosine-agnostic, then broadens into a diffuse, low-density spread over steps 20k–40k as uncertainty estimates and encoder gradients become more variable, before re-concentrating into a sharp mode near $\cos \approx 0$ from 50k steps onward and remaining stable through 100k. Second, and more informative, the sparse high-uncertainty tail does not track this trend at any stage: transitions with elevated $\hat{E}(s, a)$ remain scattered across the full cosine range, including non-trivial positive and negative values, even once the bulk population has re-concentrated near zero from 50k onward. This indicates that epistemic uncertainty is associated with the *dispersion* of gradient interaction rather than its sign: high-uncertainty transitions are precisely those for which the RL and contrastive gradients retain directional structure, while low-uncertainty transitions contribute increasingly redundant, near-orthogonal updates as training stabilizes. This provides direct evidence for the mechanism underlying our prioritization scheme: by sampling transitions proportionally to $\hat{E}(s, a)$, UCRL preferentially selects the transitions where the contrastive branch still meaningfully interacts with the shared representation, rather than expending contrastive updates on transitions where both objectives have already converged to independent, low-magnitude gradients.

Taken together, these results indicate that the contrastive branch does not systematically oppose the RL objective at the shared encoder: the population-level cosine similarity remains close to zero throughout training, and its variance decreases as training progresses. At the same time, this population-level trend masks substantial transition-level variation, particularly among transitions with high epistemic uncertainty. These transitions continue to exhibit diverse and non-trivial gradient interactions with the RL objective even at late stages of training, supporting our interpretation that uncertainty prioritization identifies transitions for which the shared representation remains actively shaped by the RL objective. Thus, the observed gradient interaction is not characterized by a persistent conflict between the two objectives, but rather by localized interaction concentrated on transitions that retain greater representation-learning potential.



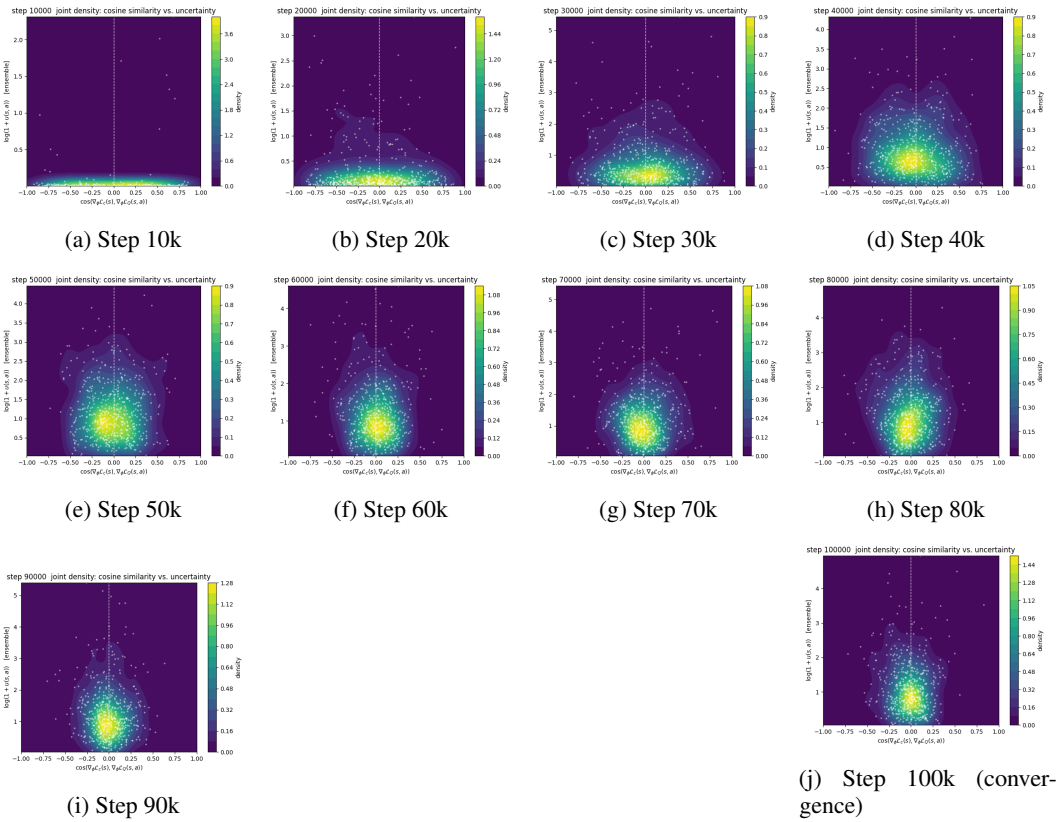
(a) Step 10k (early) (b) Step 50k (mid) (c) Step 100k (convergence)

Figure 8: Cosine similarity between the RL and contrastive encoder gradients, $\cos(\nabla_\phi \mathcal{L}_C(s), \nabla_\phi \mathcal{L}_Q(s, a))$, computed over 2048 *uniformly sampled* transitions at training steps (a) 10k, (b) 50k, and (c) 100k. Dashed red line indicates the sample mean; frac> 0 denotes the fraction of transitions with positive cosine similarity.



(a) Step 10k (early) (b) Step 50k (mid) (c) Step 100k (convergence)

Figure 9: Cosine similarity between the RL and contrastive encoder gradients, computed over 2048 *uncertainty-prioritized* transitions at training steps (a) 10k, (b) 50k, and (c) 100k, using the same axes and statistics as Figure 8 for direct comparison.



(a) Step 10k (b) Step 20k (c) Step 30k (d) Step 40k

(e) Step 50k (f) Step 60k (g) Step 70k (h) Step 80k

(i) Step 90k (j) Step 100k (convergence)

Figure 10: Joint density of gradient cosine similarity $\cos(\nabla_\phi \mathcal{L}_C(s), \nabla_\phi \mathcal{L}_Q(s,a))$ and estimated epistemic uncertainty $\log(1+\hat{E}(s,a))$ across training, from step 10k (early) to step 100k (convergence), in increments of 10k steps. Each point is a sampled transition; contours show kernel density estimates, and the dashed vertical line marks $\cos = 0$. Low-uncertainty transitions concentrate around $\cos \approx 0$ throughout training, with a broadening of the joint density around steps 20k–40k followed by re-concentration from 50k onward; the sparse high-uncertainty tail retains dispersion across the full cosine range at all stages. Color scales are normalized independently per panel and are not directly comparable across steps.

G Empirical Validation and Interpretation of Assumption 1

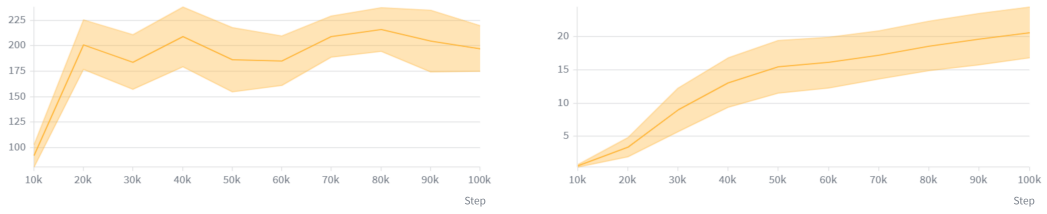
G.1 Assumption 1(i): Critic Sensitivity

We assess Assumption 1(i) on the Ball-in-Cup, Catch task, which requires a single pair of constants $0 < L_f(t) \leq U_f(t) < \infty$ such that $L_f(t) \leq \|\nabla_z Q_i(z,a)\|_{z=f_{\phi_t}(s)} \leq U_f(t)$ for every $(s,a) \in \text{supp}(D_t)$ and every critic head i – i.e. one band shared across all heads, not a per-head one.

Procedure. At each checkpoint, we sample 2048 real (s,a) pairs jointly from the replay buffer, pass s through the encoder to obtain $z = f_{\phi_t}(s)$, and compute $\|\nabla_z Q_i(z,a)\|$ for every head i via a single backward pass per head. We report the pooled sample minimum and maximum, $\hat{L}_f(t)$ and $\hat{U}_f(t)$, across all heads and samples jointly, as finite sample stand-ins for $L_f(t)$ and $U_f(t)$, together with the robust [p1, p99] percentile band. We additionally break down the per-head mean, percentile band, and sample extrema separately, since Assumption 1(i) requires the bound to hold for *every* head, not merely on average across heads. All results are averaged over 5 seeds on the Ball-in-Cup, Catch task.

Results: pooled bound. Figure 11(a) shows $\hat{L}_f(t)$ and Figure 11(b) shows $\hat{U}_f(t)$ over training. $\hat{L}_f(t)$ starts close to zero at 10K steps (0.61, averaged across seeds, with minimum of 0.28) and rises steadily to ≈ 20 by 100K, with across-seed variance *widening* rather than shrinking over training – the opposite trend from the encoder-conditioning constants in Appendix G.2. $\hat{U}_f(t)$ rises sharply from ≈ 90 at 10K to ≈ 200 by 20K, then fluctuates within a ≈ 155–240 band without a clear monotonic trend through 100K.

Results: per-head breakdown. Figure 12 shows the per-head mean, [p1, p99] band, and sample min/max at 10K, 50K, and 100K steps, against the pooled band (dashed/dotted lines). At every checkpoint, all 5 heads' means and percentile bands overlap almost exactly, and no head's distribution sits systematically outside the band occupied by the others. This directly supports the "for every head i" clause of Assumption 1(i): the bound is not merely satisfied on average across heads, but by each head individually. The raw sample maxima (red upward triangles) sit well above the pooled p99 line at every checkpoint (e.g. ≈ 100–120 vs. a p99 ceiling of ≈ 60 at 100K) – a heavy right tail that is consistent and shared across all heads rather than isolated to one, indicating a property of the sensitivity distribution itself rather than a per-head anomaly.



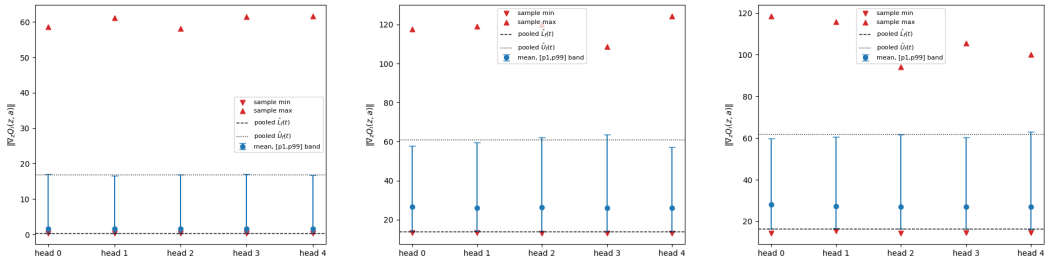
(a) $\hat{U}_f(t)$ (b) $\hat{L}_f(t)$

Figure 11: Empirical validation of Assumption 1(i), pooled across heads and samples. **(a)** $\hat{U}_f(t)$ and **(b)** $\hat{L}_f(t)$, the sample maximum and minimum of $\|\nabla_z Q_i(z,a)\|$ over 2048 sampled (s,a) pairs per head, averaged over 5 seeds (shaded: std) on Ball in Cup. $\hat{L}_f(t)$ rises from near zero at 10K to ≈ 20 by 100K, with across-seed variance widening over training.

G.2 Assumption 1(ii): Encoder Conditioning

We empirically assess Assumption 1(ii), which requires a single pair of constants $0 < \lambda_0(t) \leq \Lambda_0(t) < \infty$ such that $\lambda_0(t) \leq \lambda_{\min}(\Theta_{\phi_t}(s)) \leq \lambda_{\max}(\Theta_{\phi_t}(s)) \leq \Lambda_0(t)$ for every $(s,a) \in \text{supp}(D_t)$, where $\Theta_\phi(s) = J_\phi(s) J_\phi(s)^\top$ and $J_\phi(s) = \partial f_\phi(s)/\partial \phi$ is the encoder's parameter Jacobian. This condition underlies Lemma 1 and, transitively, the two-sided bound in Lemma 2.

Procedure. At each training checkpoint, we sample 128 states from the replay buffer, compute the true parameter Jacobian $J_\phi(s)$ for each state via `vmap(jacrev(·))`, form $\Theta_\phi(s)$, and record



(a) Step 10K (b) Step 50K (c) Step 100K

Figure 12: Per-head critic sensitivity $\|\nabla_z Q_i(z,a)\|$ at (a) 10K, (b) 50K, and (c) 100K steps, $n = 2048$ samples per head. Blue markers show mean and [p1, p99] band per head; red triangles show the sample min/max. Dashed and dotted horizontal lines mark the pooled $\hat{L}_f(t)$ and $\hat{U}_f(t)$. All 5 heads' bands overlap closely at every checkpoint, supporting a single shared bound across heads as required by Assumption 1(i).

its minimum and maximum eigenvalue. Because the full support $\mathrm{supp}(D_t)$ cannot be enumerated due to computational costs, we report $\hat{\lambda}_0(t)$ and $\hat{\Lambda}_0(t)$ as the sample minimum and maximum of $\lambda_{\min}(\Theta_\phi(s))$ and $\lambda_{\max}(\Theta_\phi(s))$, respectively, over these 128 states, as finite-sample stand-ins for the true constants $\lambda_0(t)$ and $\Lambda_0(t)$. This is repeated for 5 seeds on the Ball in Cup task; we report the across-seed mean and standard deviation of these per-run extrema.

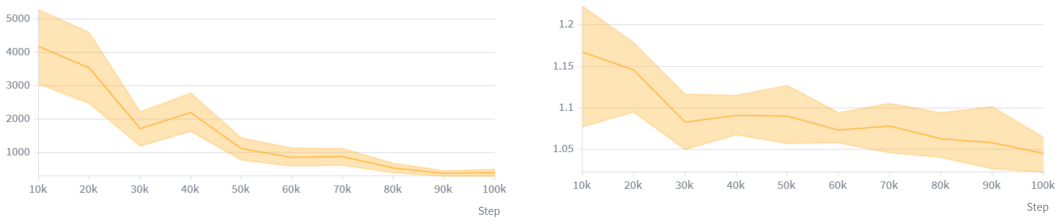
Since $\Theta_\phi(s)$ is symmetric positive semidefinite by construction, $\lambda_{\min} \geq 0$ holds trivially and is not itself informative; what Assumption1(ii) requires is $\lambda_{\min}$ *bounded away from zero*, i.e. a full-rank parameter Jacobian at every sampled state. We therefore additionally report the fraction of sampled states that are near-singular, defined as $\lambda_{\min}(\Theta_\phi(s)) < \varepsilon$ for a small threshold $\varepsilon = 1e-8$.

Results. Figure 13a shows $\hat{\Lambda}_0(t)$ and Figure 13b shows $\hat{\lambda}_0(t)$ over training. $\hat{\lambda}_0(t)$ remains bounded well away from zero throughout training, decreasing mildly from ≈ 1.17 at 10K steps to ≈ 1.05 by 100K steps, with shrinking across-seed variance. $\hat{\Lambda}_0(t)$ is substantially larger and noisier early in training (peaking near $\approx$ 4,200–5,200 at 10K steps) and decreases sharply as training progresses, stabilizing below ≈ 500 from 80K steps onward. Both quantities show their widest across-seed spread early in training and contract as the encoder and critics converge, mirroring the tightening of the fitted bound constants $\hat{C}_0/\hat{c}_0$ reported in Appendix E (Figure 4).

Across all checkpoints and all 5 seeds (640 sampled states per checkpoint), **zero** states were flagged as near-singular ($\lambda_{\min}(\Theta_\phi(s)) < \varepsilon$). Combined with $\hat{\lambda}_0(t)$ remaining comfortably positive at every checkpoint, this provides direct evidence that the encoder's parameter Jacobian retains full rank at sampled states throughout training, consistent with the lower bound in Assumption 1(ii).

Limitations. With only 128 samples per checkpoint, $\hat{\lambda}_0(t)$ and $\hat{\Lambda}_0(t)$ are sample extrema rather than means, and extrema converge more slowly in sample size than averages or percentiles; the true supremum/infimum over the full replay buffer may be more extreme than what 128 random draws capture, particularly early in training where the across-seed spread is largest. We view these curves as evidence of qualitative non-degeneracy and boundedness of $\Theta_\phi(s)$ rather than tight estimates of $\lambda_0(t)$ and $\Lambda_0(t)$ themselves.

Discussion. Taken together with Appendix G.1, these results suggest that Assumption 1's two constants behave differently over the course of training: the encoder-conditioning band $[\hat{\lambda}_0(t), \hat{\Lambda}_0(t)]$ tightens and stabilizes as training progresses, while the critic-sensitivity lower bound $\hat{L}_f(t)$ starts weak and only firms up over time. Since $c_0(t) = \sqrt{\lambda_0(t) L_f(t)}$ (Lemma 1), a weak $L_f(t)$ early in training directly weakens the overall encoder-gradient lower bound during this period – consistent with the large, noisy fitted ratio $\hat{C}_0/\hat{c}_0$ we observe at 10K–20K in Appendix E (Figure 4), which only stabilizes from $\approx$ 60K onward. We view this as a coherent picture across independent diagnostics rather than a weakness specific to one of them: Assumption 1 holds most cleanly in the regime the paper's main claims rely on (post-convergence), and is weaker but not violated during early training.



(a) $\hat{\Lambda}_0(t)$ (b) $\hat{\lambda}_0(t)$

Figure 13: Empirical validation of Assumption 1(ii). **(a)** $\hat{\Lambda}_0(t)$, the sample maximum of $\lambda_{\max}(\Theta_\phi(s))$ over 128 sampled states, and **(b)** $\hat{\lambda}_0(t)$, the sample minimum of $\lambda_{\min}(\Theta_\phi(s))$ over the same states, both averaged over 5 seeds (shaded: std) on Ball in Cup. $\hat{\lambda}_0(t)$ stays bounded away from zero throughout training, and 0/640 sampled states per checkpoint (128 $\times$ 5 seeds) were flagged as near-singular at any step.

H Theoretical Analysis

H.1 Interpretation of Assumption 2

Assumption 2 states that, in aggregate, the ensemble's critic gradients do not destructively interfere when combined. Each error term $(Q_i - y)$ reflects how far critic i deviates from the shared target, and the inner product $\langle \nabla_\phi Q_i, \nabla_\phi Q_j \rangle$ reflects whether critics i and j update the shared encoder in similar or opposing directions. Since the critics share the encoder ϕ, the head architecture, the objective, and the target, and differ only in their initial weights, they remain close throughout training, so critics with similar value estimates also tend to push the encoder in similar directions. This shared training shapes the value estimates and the encoder gradients of the critics so that critics near each other in value also push the encoder in similar directions. The assumption rules out the case where critics with large, same-signed errors happen to push the encoder in systematically opposing directions, which would cause the aggregate gradient to shrink despite substantial individual disagreement. Under this assumption, the squared norm of the aggregate gradient lower-bounds the sum of the individual squared terms,

$$\| \sum_{i=1}^{N} (Q_i - y) \nabla_\phi Q_i \|^2 \;\geq\; \sum_{i=1}^{N} (Q_i - y)^2 \, \|\nabla_\phi Q_i\|^2, \tag{10}$$

which allows us to interpret ensemble disagreement in the value estimates as a reliable proxy for the magnitude of the corrective signal received by the shared encoder. This is plausible in our setting because the shared encoder and the shared target jointly encourage the critics to move in coordinated directions, so gradients with respect to ϕ are more likely to be aligned than adversarially opposed. We validate this assumption empirically throughout training.

Metric. At each evaluation step, we sample a batch of 512 state-action pairs (s, a) from the replay buffer and compute, for each pair, the exact squared gradient norm $\|\nabla_\phi \mathcal{L}_Q\|^2$ alongside its diagonal approximation $\frac{4}{N^2} \sum_{i=1}^{N} (Q_i - y)^2 \|\nabla_\phi Q_i\|^2$. We then report the *relative gap*

$$\Delta(s, a) \;=\; \frac{\|\nabla_\phi \mathcal{L}_Q\|^2 - (\text{diagonal approx.})}{\|\nabla_\phi \mathcal{L}_Q\|^2}, \tag{11}$$

which, up to the positive normalization constant $4/N^2$, has the same sign as the off-diagonal sum in Assumption 2. A positive $\Delta(s, a)$ for a given (s, a) pair therefore certifies that the assumption holds pointwise at that sample; averaging Δ over the 512-sample batch gives an aggregate, per-step check of the assumption over the current state-action distribution.

Setup. We track the mean of $\Delta(s, a)$ over the 512 sampled pairs at every evaluation step across training, repeated over 5 random seeds. We report the mean relative gap together with the min–max range across seeds.

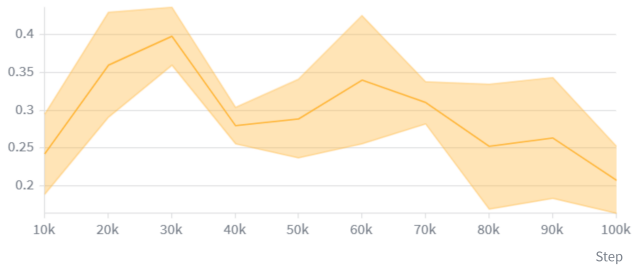


Figure 14: Relative gap $\Delta(s,a) = (\|\nabla_\phi \mathcal{L}_Q\|^2 - \text{diagonal approx.})/\|\nabla_\phi \mathcal{L}_Q\|^2$, averaged over 512 sampled state-action pairs per evaluation step, across 5 seeds (shaded region shows min–max range). The gap remains strictly positive throughout training, empirically validating Assumption 2.

Results. Figure 14 shows the resulting relative gap over 100k training steps. The mean relative gap is not only strictly positive at every evaluation step, for every seed, but remains comfortably away from zero, fluctuating between approximately 0.15 and 0.45 across training. Almost all individual samples contribute a positive gap, and the aggregate margin is substantial rather than marginal. This provides direct empirical support for Assumption 2: although individual off-diagonal gradient-alignment terms need not vanish (nor even be individually non-negative), their sum remains positive throughout training. The magnitude of the gap further indicates that the diagonal approximation is not merely a valid lower bound but a noticeably conservative one, consistently underestimating the true squared gradient norm by roughly 20–40%. This is consistent with Figure 14: as the critics converge toward a shared value estimate, their updates to the encoder increasingly point in the same direction, so the aggregate gradient is dominated by the aligned component rather than canceled by conflicting ones.

H.2 Proof of Lemma 1

Proof. Let $\nabla_z Q_i \in \mathbb{R}^d$ denote the gradient of the i-th critic with respect to the latent representation $z = f_\phi(s)$. By the chain rule,

$$\nabla_\phi Q_i = J_\phi(s)^\top \nabla_z Q_i,$$

and therefore

$$\left\|\nabla_\phi Q_i\right\|^2 \;=\; \nabla_z Q_i{}^\top J_\phi(s) J_\phi(s)^\top \nabla_z Q_i \;=\; \nabla_z Q_i{}^\top \Theta_\phi(s)\, \nabla_z Q_i.$$

Lower bound. By the Rayleigh quotient bound for the symmetric positive semidefinite matrix $\Theta_\phi(s)$,

$$\nabla_z Q_i{}^\top \Theta_\phi(s)\, \nabla_z Q_i \;\geq\; \lambda_{\min}\big(\Theta_\phi(s)\big)\, \|\nabla_z Q_i\|^2 \;\geq\; \lambda_0(t)\, L_f(t)^2,$$

where the second inequality follows lower-bounds from Assumption 1(i) and (ii).

Upper bound. By the same Rayleigh quotient bound applied from above,

$$\nabla_z Q_i{}^\top \Theta_\phi(s)\, \nabla_z Q_i \;\leq\; \lambda_{\max}\big(\Theta_\phi(s)\big)\, \|\nabla_z Q_i\|^2 \;\leq\; \Lambda_0(t)\, U_f(t)^2,$$

where the second inequality follows upper-bounds from Assumption 1(i) and (ii).

Combining both bounds and taking square roots completes the proof. □

H.3 Proof of Lemma 2

Proof. **Step 1: Encoder gradient of the ensemble critic loss.** With a shared encoder, the ensemble loss is

$$\mathcal{L}_Q(\phi) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_i(\phi), \qquad \mathcal{L}_i = \big(Q_i(f_\phi(s), a) - y\big)^2, \tag{12}$$

where we assume a shared target y across heads. Differentiating with respect to the encoder parameters gives

$$\nabla_\phi \mathcal{L}_Q = \frac{2}{N}\sum_{i=1}^{N} (Q_i - y)\, \nabla_\phi Q_i. \tag{13}$$

We now bound $\|\nabla_\phi \mathcal{L}_Q\|$ from below and above separately.

Step 2a: Lower bound — squared norm. Taking the squared norm of Eq.13 and expanding,

$$\|\nabla_\phi \mathcal{L}_Q\|^2 = \frac{4}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}(Q_i - y)(Q_j - y)\,\langle \nabla_\phi Q_i,\, \nabla_\phi Q_j\rangle. \tag{14}$$

Under Assumption 2, the sum of all cross terms, $\sum_{i\neq j}(\cdot)$, is non-negative and can therefore be dropped, leaving only the diagonal terms:

$$\|\nabla_\phi \mathcal{L}_Q\|^2 \geq \frac{4}{N^2}\sum_{i=1}^{N}(Q_i - y)^2\,\|\nabla_\phi Q_i\|^2. \tag{15}$$

The coefficients $(Q_i - y)^2$ are non-negative, so substituting the per-head lower bound $\|\nabla_\phi Q_i\| \geq c_0(t)$ from Lemma 1 preserves the inequality term by term:

$$\|\nabla_\phi \mathcal{L}_Q\|^2 \;\geq\; \frac{4\,c_0(t)^2}{N^2}\sum_{i=1}^{N}(Q_i - y)^2. \tag{16}$$

Step 2b: Upper bound — triangle inequality. Applying the triangle inequality directly to Eq.13, followed by the per-head upper bound $\|\nabla_\phi Q_i\| \leq C_0(t)$ from Lemma 1,

$$\|\nabla_\phi \mathcal{L}_Q\| \;\leq\; \frac{2}{N}\sum_{i=1}^{N}|Q_i - y|\,\|\nabla_\phi Q_i\| \;\leq\; \frac{2\,C_0(t)}{N}\sum_{i=1}^{N}|Q_i - y|. \tag{17}$$

By Cauchy–Schwarz, $\sum_{i=1}^{N}|Q_i - y| \leq \sqrt{N}\Big(\sum_{i=1}^{N}(Q_i - y)^2\Big)^{1/2}$, so

$$\|\nabla_\phi \mathcal{L}_Q\| \;\leq\; \frac{2\,C_0(t)}{\sqrt{N}}\Big(\sum_{i=1}^{N}(Q_i - y)^2\Big)^{1/2}. \tag{18}$$

Note that, unlike the lower bound, this step does not require Assumption 2: the triangle inequality controls the sum directly, without needing the cross terms in the expanded squared norm to vanish.

Step 3: Decomposition into epistemic and Bellman components. The sum $\sum_{i=1}^{N}(Q_i - y)^2$, common to both bounds, decomposes exactly about the ensemble mean $\bar{Q} = \frac{1}{N}\sum_{j=1}^{N} Q_j$. Writing $Q_i - y = (Q_i - \bar{Q}) + (\bar{Q} - y)$ and expanding,

$$\sum_{i=1}^{N}(Q_i - y)^2 = \sum_{i=1}^{N}(Q_i - \bar{Q})^2 + 2(\bar{Q} - y)\sum_{i=1}^{N}(Q_i - \bar{Q}) + N(\bar{Q} - y)^2 \tag{19}$$

$$= \sum_{i=1}^{N}(Q_i - \bar{Q})^2 + N(\bar{Q} - y)^2, \tag{20}$$

where the middle term vanishes because $\sum_{i=1}^{N}(Q_i - \bar{Q}) = 0$ by definition of $\bar{Q}$. Identifying the ensemble variance $\hat{E}(s,a) = \widehat{\mathrm{Var}}_i[Q_i] = \frac{1}{N}\sum_{i=1}^{N}(Q_i - \bar{Q})^2$ as the epistemic uncertainty defined in Section 3 and $(\bar{Q} - y)^2$ as the squared mean Bellman error, we obtain

$$\sum_{i=1}^{N}(Q_i - y)^2 = N\Big(\hat{E}(s,a) + (\bar{Q} - y)^2\Big). \tag{21}$$

Step 4: Combine. Substituting Eq.21 into Eq.16 and taking square roots gives the lower bound; substituting into Eq.18 gives the upper bound directly:

$$\|\nabla_\phi \mathcal{L}_Q\|^2 \;\geq\; \frac{4\,c_0(t)^2}{N}\Big(\hat{E}(s,a) + (\bar{Q} - y)^2\Big), \qquad \|\nabla_\phi \mathcal{L}_Q\| \;\leq\; 2\,C_0(t)\sqrt{\hat{E}(s,a) + (\bar{Q} - y)^2}. \tag{22}$$

Taking square roots in the first inequality and combining with the second yields Eq.5 with

$$c(t) = \frac{2\sqrt{\lambda_0(t)}\, L_f(t)}{\sqrt{N}} = \frac{2c_0(t)}{\sqrt{N}}, \qquad C(t) = 2\sqrt{\Lambda_0(t)}\, U_f(t) = 2C_0(t).$$

All terms involved are non-negative, so the epistemic and Bellman contributions enter both bounds additively. □

I Extended Results

Table 7: Comparison on 16 DeepMind Control Suite 100k tasks.

Task	UCRL-C	CURL
Finger, spin	**805** $\pm$ 85	767 $\pm$ 56
Cartpole, swingup	**715** $\pm$ 37	582 $\pm$ 146
Cartpole, balance	**944** $\pm$ 10	894 $\pm$ 179
Cartpole, balance sparse	994 $\pm$ 8	**1000** $\pm$ 0
Cartpole, swingup sparse	**110** $\pm$ 118	1 $\pm$ 3
Reacher, easy	**880** $\pm$ 64	538 $\pm$ 233
Reacher, hard	**557** $\pm$ 202	431 $\pm$ 120
Cheetah, run	170 $\pm$ 31	**299** $\pm$ 48
Walker, walk	**555** $\pm$ 114	403 $\pm$ 24
Walker, stand	**919** $\pm$ 47	765 $\pm$ 384
Ball in cup, catch	**952** $\pm$ 20	769 $\pm$ 43
Finger, turn easy	244 $\pm$ 99	**371** $\pm$ 203
Finger, turn hard	94 $\pm$ 93	**111** $\pm$ 58
Hopper, stand	**23** $\pm$ 35	4 $\pm$ 2
Hopper, hop	0 $\pm$ 0	0 $\pm$ 0
Pendulum, swingup	0 $\pm$ 0	**20** $\pm$ 44

I.1 Sensitivity Analysis

We sweep the prioritization coefficient $\alpha \in \{0.1, 0.3, 0.5, 0.7, 1.0\}$ and ensemble size $N \in \{2, 3, 5, 7, 10\}$ on Ball in Cup, Catch (3 seeds per configuration). To isolate the effect of N on the quality of the uncertainty estimate from its effect on the encoder's gradient (as a larger ensemble otherwise injects more gradient signal into the shared encoder), we apply stop-gradient on the additional critics throughout this analysis, following the setup in Section 5.5. Performance is stable across most of the grid (mean return 880–956), with our default configuration (N=5, α=0.7) performing near the top of this range (950 ± 27). Degradation is concentrated primarily at large ensembles (N=10), which show both lower mean return and substantially higher variance across most values of α (e.g., 880 ± 57 at α=0.7), suggesting that oversized ensembles produce a noisier or less reliable uncertainty estimate rather than improving it. Instability at α=1.0 is less consistent: it is pronounced at N=7 (891 ± 99) but not at N=5 (920 ± 7), indicating that full commitment to the uncertainty signal does not uniformly destabilize training and its effect depends on ensemble size. The corresponding heatmap is shown in Figure 15.

Table 8: Performance on Atari 100K. Relative gain is computed from the unrounded five-seed means as (UCRL-C - CURL) / |CURL| × 100. Bold indicates the better result; lower is better for Optimality Gap. Bold indicates the higher unrounded five-seed mean within each matched pair. IQM and Optimality Gap use all five seeds per game; brackets show 95% stratified-bootstrap confidence intervals. Lower is better for Optimality Gap.

Game	Human	Random	UCRL-C	CURL	Rel. Gain (%)
Alien	7127.7	227.8	**822.4**	711.2	+15.64
Amidar	1719.5	5.8	100.6	**150.2**	-33.05
Assault	742.0	222.4	**522.8**	520.3	+0.49
Asterix	8503.3	210.0	549.0	**620.0**	-11.45
Bank Heist	753.1	14.2	**65.2**	35.6	+83.15
Battle Zone	37187.5	2360.0	**9940.0**	8660.0	+14.78
Boxing	12.1	0.1	1.5	**5.8**	-74.48
Breakout	30.5	1.7	**2.8**	2.3	+22.41
Chopper Command	7387.8	811.0	**846.0**	828.0	+2.17
Crazy Climber	35829.4	10780.5	**20066.0**	14374.0	+39.60
Demon Attack	1971.0	152.1	442.6	**497.5**	-11.04
Freeway	29.6	0.0	**28.5**	28.2	+0.99
Frostbite	4334.7	65.2	**1842.0**	1832.2	+0.53
Gopher	2412.5	257.6	**421.2**	270.8	+55.54
Hero	30826.4	1027.0	**5974.8**	4099.1	+45.76
James Bond	302.8	29.0	291.0	**411.0**	-29.20
Kangaroo	3035.0	52.0	**2220.0**	744.0	+198.39
Krull	2665.5	1598.0	3570.2	**3652.6**	-2.26
Kung Fu Master	22736.3	258.5	**14364.0**	8134.0	+76.59
Ms. Pac-Man	6951.6	307.3	**935.4**	925.2	+1.10
Pong	14.6	-20.7	**-18.48**	-18.54	+0.32
Private Eye	69571.3	24.9	45.7	**100.0**	-54.30
Qbert	13455.0	163.9	700.0	**913.0**	-23.33
Road Runner	7845.0	11.5	**7216.0**	6258.0	+15.31
Seaquest	42054.7	68.4	**421.6**	277.6	+51.87
Up N Down	11693.2	533.4	2479.2	**2489.2**	-0.40
Mean HNS (%)↑	–	–	**33.91**	32.26	–
Median HNS (%)↑	–	–	**13.74**	12.33	–
IQM HNS (%) ↑	–	–	**15.45** [12.93, 18.19]	11.72 [9.74, 14.52]	–
Optimality Gap (%) ↓	–	–	**71.42** [69.22, 73.61]	74.13 [71.66, 76.50]	–
Mean Relative Gain (%) ↑	–	–	–	–	+14.81
Median Relative Gain (%) ↑	–	–	–	–	+1.05

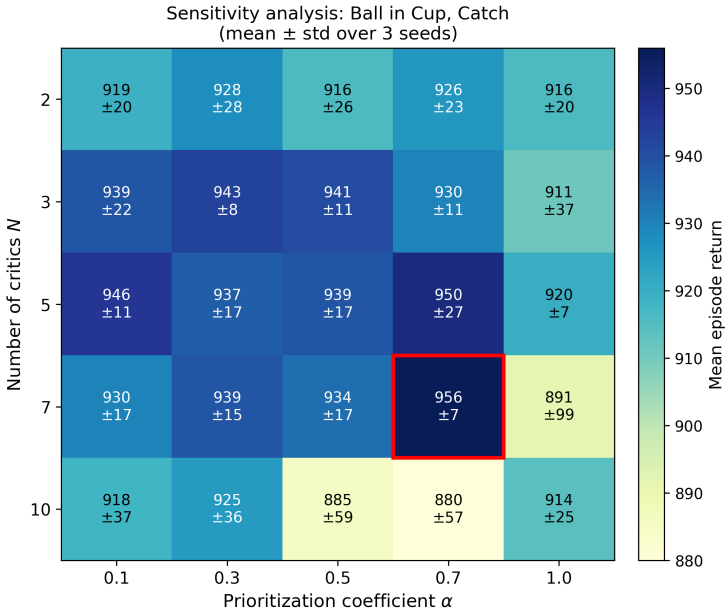


Figure 15: Sensitivity of UCRL to the number of ensemble critics N and the prioritization coefficient α on Ball in Cup, Catch (mean $\pm$ std over 3 seeds). Stop-gradient is applied to the additional critics throughout this analysis to isolate the effect of N on the uncertainty estimate from its effect on the shared encoder's gradient. The best-performing configuration is outlined in red.