Value-Priced Uncertainty: A Family of PPO-Compatible Exploration Bonuses

Qili Shen[1,2] **Xuanhong Chen**[3]* **Ang He**[4] **Dake Zhang**[3]* **Kairui Feng**[1,2]*

[1]Shanghai Research Institute for Intelligent Autonomous Systems &
State Key Laboratory of Autonomous Intelligent Unmanned Systems,
Tongji University, Shanghai 201210, China
[2]Shanghai Innovation Institute, Shanghai, China
[3]Shanghai Jiao Tong University, Shanghai, China
[4]Fudan University, Shanghai, China

Abstract

Proximal Policy Optimization (PPO) is value-driven only after experience has been collected: Generalized Advantage Estimation reinforces sampled actions according to their bootstrapped advantage, but the exploration process that produced those samples is still governed mainly by policy-space noise and entropy regularization. As a result, PPO does not explicitly ask, before sampling, which uncertain transitions would most improve its advantage estimates. We argue that PPO exploration should be formulated as *value-priced uncertainty*: uncertainty should be explored when it is both large and able to change the advantage. We propose VALU (**V**alue-**A**ware **L**atent **U**ncertainty), a family of PPO-compatible exploration bonuses of the form $b(s,a) = P(s,a)U(s,a)$. The price $P(s,a) = \|\nabla_{z'}\tilde{V}_\psi(\hat{z}')\|$ measures how strongly uncertainty in the predicted next latent can perturb the advantage, while $U(s,a)$ is an uncertainty quantity instantiated by non-parametric effective counts or learned prediction errors such as RND. Our theory derives the value-pricing term from a unified advantage-uncertainty view: an uncertainty coordinate induces an optimistic advantage correction through a dual norm. This yields count-based UCB and MCR members, corresponding respectively to optimism over latent-transition uncertainty and one-sample shrinkage of the optimistic advantage envelope. Across six standard MuJoCo continuous-control tasks, VALU substantially improves PPO's sample efficiency and final return, and achieves the best PPO-compatible performance on most tasks while remaining competitive on the others, compared with strong exploration baselines including RND, ICM, RE3, VCSE, and OPPO. Ablations over uncertainty estimators, schedules, latent encoders, and bonus components show that the gains come from pricing uncertainty by local value sensitivity, rather than from uncertainty or value sensitivity alone.

1 Introduction

Proximal Policy Optimization (PPO) [Schulman et al., 2017] is widely used in continuous-control reinforcement learning. In each on-policy iteration, PPO uses Generalized Advantage Estimation (GAE) [Schulman et al., 2016] to update the actor with bootstrapped advantage estimates, making the update value-driven after sampling. During rollout collection, however, exploration is still driven mainly by policy noise and entropy regularization. Such exploration increases behavioral diversity but is not aligned with the critic's latent geometry: PPO may sample value-irrelevant novelty while

*Corresponding authors: chenxuanhong@sjtu.edu.cn, dk.zhang@sjtu.edu.cn, kelvinfkr@tongji.edu.cn

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

missing low-coverage regions where small next-latent errors would strongly change the advantage estimate.

Exploration bonuses are a natural interface for adding such a signal to PPO, because they can be used to modify the advantage targets without changing the clipped objective. Existing intrinsic rewards have explored several useful objectives. Prediction-error methods such as ICM [Pathak et al., 2017] and RND [Burda et al., 2019] reward transitions that are hard to predict. Count- and density-based methods reward rarely visited regions [Bellemare et al., 2016, Ostrovski et al., 2017]. State-entropy methods, including RE3 [Seo et al., 2021], APT [Liu and Abbeel, 2021], and Proto-RL [Yarats et al., 2021], encourage broad coverage in a learned or fixed representation space. Other methods bring value information into exploration through scalar value uncertainty [O'Donoghue et al., 2018, Morere and Ramos, 2018, Flennerhag et al., 2020], value-conditional entropy [Kim et al., 2023], or value-aware model learning [Farahmand et al., 2017, Voelcker et al., 2022]. These methods provide useful exploration pressure, but they do not directly price state–action uncertainty by how much it can perturb the advantage.

We propose to regularize PPO exploration from first principles, by asking how local uncertainty propagates into the advantage estimate that PPO actually optimizes. Let ξ denote a local uncertainty coordinate associated with a transition (s, a). To first order, ξ perturbs the advantage as $A_k(s, a; \xi) \approx \hat{A}_k(s, a) + \langle G_k(s, a), \xi \rangle$, where $G_k(s, a)$ is the local sensitivity of the advantage to ξ. Taking the supremum of this perturbed advantage over a confidence set gives, by duality, an optimistic correction whose bonus is an uncertainty radius times a dual sensitivity norm. This norm-bound template is the starting point of our analysis: under different choices of ξ, it recovers OPPO [Cai et al., 2020] as a feature/parameter-space correction under a least-squares confidence norm.

We instantiate the template in latent transition space. Take ξ to be the next-latent perturbation δ: if the predicted next latent is $\hat{z}'$ and the true next latent is $\hat{z}' + \delta$, the one-step advantage changes by $\gamma \nabla_{z'} \tilde{V}_\psi(\hat{z}')^\top \delta$ to first order, and the dual sensitivity norm becomes the value-gradient norm $\|\nabla_{z'} \tilde{V}_\psi(\hat{z}')\|$. This yields the central principle of our method: uncertainty is useful for exploration only when it is also value-sensitive, and the value-gradient norm prices exactly that sensitivity. Based on this principle, we propose VALU (**V**alue-**A**ware **L**atent **U**ncertainty), a family of PPO-compatible exploration bonuses:

$$b(s, a) = P(s, a)U(s, a), \qquad P(s, a) = \|\nabla_{z'} \tilde{V}_\psi(\hat{z}')\|. \tag{1.1}$$

The same template prescribes how the uncertainty quantity $U(s, a)$ should be set. VALU-UCB takes the supremum over a count-based latent confidence set, giving an effective-count schedule; VALU-MCR rewards actions whose samples most shrink the optimistic advantage envelope, giving a marginal confidence reduction schedule. Learned estimators such as RND [Burda et al., 2019] do not change the value-gradient price; they provide an alternative empirical estimate of $U(s, a)$ in the same template. The latent representation supplying $\hat{z}'$ can be learned by JEPA [LeCun, 2022, Assran et al., 2023], by a VAE [Kingma and Welling, 2014], or by a frozen-random encoder [Seo et al., 2021]. Different uncertainty estimators and latent interfaces can therefore be plugged into the same value-pricing principle, making VALU a method family rather than a single fixed bonus.

Figure 1 summarizes the framework. In implementation, VALU adds the resulting bonus after rollout collection as an advantage-level correction, $\hat{A}_t^{\mathrm{VALU}} = \hat{A}_t + \beta b_t$, with the corresponding critic target $\hat{R}_t^{\mathrm{VALU}} = V_\xi(s_t) + \hat{A}_t^{\mathrm{VALU}}$. PPO then uses the same clipped surrogate, entropy term, and optimizer as before. Thus PPO remains unchanged as an optimizer, but its exploration is guided toward latent state–action regions where uncertainty is both high and value-sensitive.

We summarize the main contributions of this paper:

- We propose VALU, a family of PPO-compatible exploration bonuses that regularize latent-space exploration by the geometry of the value function. VALU weights local state–action uncertainty by the value-gradient norm, turning value-blind novelty into value-relevant exploration pressure.
- We derive VALU from a unified advantage-uncertainty view. UCB-style corrections arise by taking the supremum of a perturbed advantage over a confidence set; VALU-UCB instantiates this template in latent transition space, while VALU-MCR follows from the one-sample shrinkage of the optimistic advantage envelope.

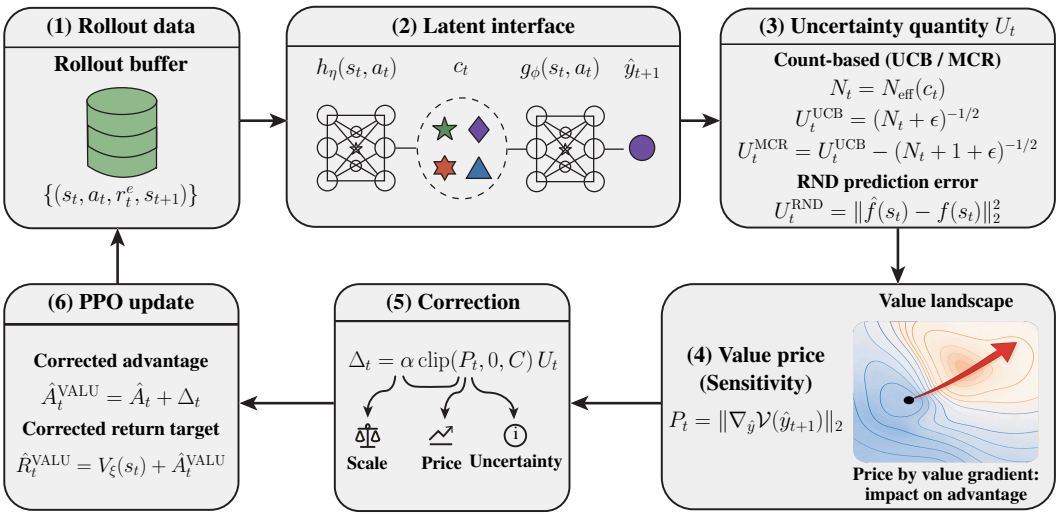


Figure 1: VALU framework. An on-policy rollout supplies the latent interface, uncertainty quantity U_t, and local value price P_t. Their product gives the post-GAE correction Δ_t, added once to both the actor advantage and critic return target.

- We validate VALU on six MuJoCo continuous-control tasks, where it improves PPO and outperforms strong PPO-compatible exploration baselines on most tasks. Ablations over uncertainty estimators, schedules, encoders, and bonus components confirm that both uncertainty quantity and value sensitivity are important.

2 Related work

Uncertainty quantity without value pricing. A common way to improve exploration is to add randomness in policy space, such as ϵ-greedy exploration [Sutton and Barto, 2018], parameter-space noise [Plappert et al., 2018], NoisyNets [Fortunato et al., 2018], or entropy regularization [Mnih et al., 2016, Haarnoja et al., 2018]. These methods increase behavioral diversity, but they do not measure whether the sampled transitions are useful for improving the critic. Bonus-based methods instead add an exploration signal to the policy update, often through rewards or advantage targets. Prediction-error methods, including ICM [Pathak et al., 2017] and RND [Burda et al., 2019], reward transitions that are hard to predict; pseudo-count and density methods reward rarely visited regions [Bellemare et al., 2016, Ostrovski et al., 2017, Tang et al., 2017]; and state-entropy methods encourage broad coverage in representation space [Seo et al., 2021, Liu and Abbeel, 2021, Yarats et al., 2021]. VALU shares the goal of estimating where exploration is needed, but differs in how the estimate is used. In the count-based members, $N_{\text{eff}}(c)$ over coverage points c (embeddings of state–action pairs used to measure local visitation) provides a non-parametric uncertainty quantity. In the RND-based member, the RND prediction error provides a learned empirical uncertainty quantity. The original RND baseline uses this prediction error directly as an intrinsic bonus, whereas VALU-RND uses it as U and multiplies it by the value-gradient price. Thus VALU asks not only whether a region is novel or under-covered, but whether uncertainty in that region can change the advantage.

Value information without latent uncertainty pricing. Several methods incorporate value information into exploration. UBE [O'Donoghue et al., 2018], EMU-Q [Morere and Ramos, 2018], and TDU [Flennerhag et al., 2020] use value uncertainty or temporal-difference uncertainty as exploration signals. VCSE [Kim et al., 2023] conditions a state-entropy bonus on value similarity. OPPO [Cai et al., 2020] is closest to our UCB-level view: it adds an upper-confidence correction to policy optimization under a linear feature/parameter confidence model. In our theory, OPPO and VALU-UCB share the same optimistic-advantage template, but place the confidence set in different spaces. OPPO places uncertainty on linear value parameters and obtains a Mahalanobis feature norm; VALU-UCB places uncertainty on the latent transition and prices it by the value-gradient norm.

VALU also connects to UCB-style optimism [Auer et al., 2002, Srinivas et al., 2010] and marginal value-of-information ideas [Frazier et al., 2008, Frazier, 2018]. However, our MCR schedule is not introduced as a standalone heuristic. It is derived as the one-sample shrinkage of the optimistic advantage envelope under the same latent-transition uncertainty model. The closest use of value gradients appears in value-aware model learning, where VAML [Farahmand et al., 2017] and VaGraM [Voelcker et al., 2022] weight model-prediction errors by local value sensitivity. VALU uses the same principle on the exploration side: rather than shaping the model loss, it asks PPO to collect data where latent uncertainty is both high and value-sensitive.

3 Preliminaries

3.1 PPO with advantage correction

PPO [Schulman et al., 2017] is an on-policy actor–critic method. At each iteration, it collects rollouts $\{(s_t, a_t, r_t^e, s_{t+1})\}$ using the current policy $\pi_{\theta_{\text{old}}}$, estimates advantages with GAE [Schulman et al., 2016], and updates the actor with the clipped surrogate

$$\mathcal{L}^{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(\rho_t(\theta)\hat{A}_t,\ \text{clip}(\rho_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t\right)\right], \quad \rho_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}. \tag{3.1}$$

The advantage estimate is computed from temporal-difference residuals

$$\hat{A}_t = \sum_{l\geq 0}(\gamma\lambda)^l \delta_{t+l}, \qquad \delta_t = r_t^e + \gamma V_\xi(s_{t+1}) - V_\xi(s_t), \tag{3.2}$$

where V_ξ is PPO's critic. VALU leaves the clipped surrogate, entropy term, optimizer, and rollout collection procedure unchanged. It modifies the actor advantage and the corresponding critic regression target:

$$\hat{A}_t^{\text{VALU}} = \hat{A}_t + \beta b_t, \qquad \hat{R}_t^{\text{VALU}} = V_\xi(s_t) + \hat{A}_t^{\text{VALU}}, \tag{3.3}$$

where $b_t = b(s_t, a_t)$ is the VALU bonus and β is its scale. This makes the implementation match the object analyzed in Section 4: VALU turns local uncertainty into an optimistic or reducible advantage correction and adds that correction directly to PPO's advantage estimate.

3.2 Latent interface and uncertainty quantities

VALU requires a latent interface that exposes two quantities per transition (s_t, a_t, s_{t+1}): a state–action coverage point $c_t = h_\eta(s_t, a_t)$ (an embedding of a state–action pair used to measure local coverage) and a predicted next object $\hat{y}_{t+1} = g_\phi(s_t, a_t)$ paired with a value function $\mathcal{V}$, giving the value sensitivity $P_t = \|\nabla_{\hat{y}}\mathcal{V}(\hat{y}_{t+1})\|_2$. The coverage point is state–action rather than state-only, so it distinguishes different actions taken in the same state—the object relevant to an advantage correction. Both c_t and $\hat{y}_{t+1}$ are read off the same learned latent space, which can be instantiated by a JEPA-style action-conditioned predictor [LeCun, 2022, Assran et al., 2023], a VAE-style state–action bottleneck, or a frozen-random encoder. The auxiliary predictor is trained on PPO's own rollouts and is detached from the PPO policy loss; a frozen-random encoder itself is not updated. Predictive training of learned interfaces encourages dynamics-similar pairs to have nearby coverage points. This motivates k-nearest-neighbour (kNN) counting in latent space. VALU therefore keeps a fixed-capacity memory bank $\mathcal{B} = \{c_i\}$, a sliding collection of recent coverage points, and uses $N_{\text{eff}}(c) = \sum_{c_i\in\mathcal{B}} K_h(c, c_i)$, truncated to the k nearest neighbours in practice, as a kNN-style coverage estimate [Kozachenko and Leonenko, 1987, Kraskov et al., 2004]; the UCB and MCR schedules built from N_{eff} are defined in Section 5.1. As a learned alternative, RND [Burda et al., 2019] replaces N_{eff} with the prediction error $U_{\text{RND}}(s_t) = \|\hat{f}(s_t) - f(s_t)\|_2^2$, where $\hat{f}$ is trained to match a fixed random target f; the value-gradient price P_t is unchanged, so count-based and RND-based members share the same value-pricing interface and differ only in how U is measured.

4 Theory: advantage uncertainty and value-priced exploration

We now formalize the exploration principle behind VALU. Section 3.1 showed that PPO updates its policy from advantage estimates, which are uncertain in under-covered regions of state–action space. We argue that exploration should target this uncertainty in proportion to its effect on the advantage, and we derive the bonus form used in Section 5 from a unified advantage-uncertainty template.

4.1 A unified advantage-uncertainty template

Let ξ denote a local uncertainty coordinate associated with (s, a). We model its effect on the advantage to first order as

$$A_k(s, a; \xi) \approx \hat{A}_k(s, a) + \langle G_k(s, a), \xi \rangle, \tag{4.1}$$

where $G_k(s, a)$ is the local sensitivity of the advantage to ξ. Given a confidence set $\mathcal{U}_k(s, a)$ for the unknown perturbation, two first-principle perspectives arise.

UCB perspective. Taking the supremum of Eq. (4.1) over $\mathcal{U}_k(s, a)$ gives an upper-confidence advantage

$$A_k^+(s, a) = \hat{A}_k(s, a) + \sup_{\xi \in \mathcal{U}_k(s,a)} \langle G_k(s, a), \xi \rangle = \hat{A}_k(s, a) + r_k(s, a) \, \|G_k(s, a)\|_*, \tag{4.2}$$

where $r_k(s, a)$ is the radius of $\mathcal{U}_k(s, a)$ and $\| \cdot \|_*$ is the dual of the norm defining the set. The exploration bonus is therefore a product of an uncertainty radius and a dual sensitivity norm.

MCR perspective. Writing $R_k(N; s, a) = r_k(s, a) \, \|G_k(s, a)\|_*$ as a function of an effective sample count N, the one-sample shrinkage of the optimistic advantage envelope is

$$\mathcal{I}_k(s, a) := R_k(N; s, a) - R_k(N + 1; s, a). \tag{4.3}$$

This is the marginal value of one more sample at (s, a) for reducing optimistic-advantage uncertainty.

4.2 VALU-UCB: optimism over latent-transition uncertainty

VALU-UCB places the confidence set in latent transition space:

$$\mathcal{U}_k^{\mathrm{VALU}}(s, a) = \{\delta z' : \ \|\delta z'\|_{\mathcal{B}} \le u_k(N_{\mathrm{eff}}(z, a))\}, \tag{4.4}$$

so that $r_k(s, a) = u_k(N_{\mathrm{eff}}(z, a))$ and $G_k(s, a) = \gamma \, \nabla_{z'} \tilde{V}_k(\hat{z}')$. Substituting into Eq. (4.2) yields

$$A_k^{\text{VALU-UCB}}(s, a) = \hat{A}_k(s, a) + \gamma \, u_k(N_{\mathrm{eff}}(z, a)) \, \|\nabla_{z'} \tilde{V}_k(\hat{z}')\|_{\mathcal{B}^*}. \tag{4.5}$$

The Euclidean norm together with $u_k(N) = 1/\sqrt{N + \epsilon}$ gives the count-based UCB bonus implemented in Section 5.1. VALU-UCB is therefore optimism over the possible advantage under latent transition uncertainty.

Relation to OPPO. OPPO [Cai et al., 2020] is Eq. (4.2) with a different uncertainty coordinate. In the linear value setting $Q_w(s, a) = \phi(s, a)^\top w$, the perturbation is the parameter error $\xi = \Delta w$ with $G_k(s, a) = \phi_k(s, a)$, and the confidence set is the least-squares ellipsoid $\|\Delta w\|_{\Lambda_k} \le \beta_k$. This gives $A_k^{\mathrm{OPPO}}(s, a) = \hat{A}_k(s, a) + \beta_k \|\phi_k(s, a)\|_{\Lambda_k^{-1}}$. OPPO and VALU-UCB therefore share the dual-norm structure of Eq. (4.2) but place uncertainty in different spaces: OPPO on linear value parameters, VALU-UCB on the latent transition.

4.3 VALU-MCR: optimism over reducible advantage uncertainty

VALU-MCR adopts the second perspective. Rather than asking how large the current optimistic advantage could be, it asks how much one more sample reduces that uncertainty. Formally, this is an information-constrained selection problem $\max_a \hat{A}_k(s, a)$ subject to $\mathcal{I}_k(s, a) \ge \eta$, whose Lagrangian is equivalent, up to constants independent of a, to $\max_a [\hat{A}_k(s, a) + \beta \, \mathcal{I}_k(s, a)]$. With the VALU latent confidence radius $R_k(N; s, a) = \gamma \, u_k(N) \, \|\nabla_{z'} \tilde{V}_k(\hat{z}')\|_{\mathcal{B}^*}$, Eq. (4.3) becomes

$$\mathcal{I}_k(s, a) = \gamma \, [u_k(N) - u_k(N + 1)] \, \|\nabla_{z'} \tilde{V}_k(\hat{z}')\|_{\mathcal{B}^*}, \qquad N = N_{\mathrm{eff}}(z, a). \tag{4.6}$$

With $u_k(N) = 1/\sqrt{N + \epsilon}$, Eq. (4.6) gives the MCR schedule used in Section 5.1. VALU-MCR thus rewards actions whose samples most shrink the local optimistic advantage envelope.

Learned uncertainty quantities. The derivations above specify how an uncertainty quantity should be *priced* by advantage sensitivity, but leave open how that quantity is *measured*. The count-based confidence radius $u_k(N_{\text{eff}}(c))$ is a non-parametric choice. A learned estimator such as RND can be plugged into the same template by replacing $u_k(N_{\text{eff}}(c))$ with an empirical prediction-error estimate $U_{\text{RND}}(s)$: the value-gradient price is unchanged, only the estimator of U differs. We exploit this modularity in Section 5, where UCB, MCR, and RND-based members all share the same value-pricing mechanism.

5 Method

The theory in Section 4 prescribes an advantage-level correction of the form $\beta\, b(s,a)$, with $b(s,a) = P(s,a)\, U(s,a)$ a value-priced uncertainty bonus. We now describe the implementation: how the value price P_t is computed from the latent interface, how the uncertainty quantity U_t is supplied by either non-parametric coverage or a learned estimator, and how the resulting bonus is integrated into PPO without changing the optimizer.

5.1 Value-priced latent uncertainty

The value price is the gradient norm at the predicted next object,$P_t = \|\nabla_{\hat{y}} \mathcal{V}(\hat{y}_{t+1})\|_2$, which is the Euclidean instance of the dual sensitivity norm in Eq. (4.2). The latent interface (Section 3.2) determines $\hat{y}_{t+1}$ and $\mathcal{V}$: in a JEPA-style state-latent implementation, $\hat{y}_{t+1} = \hat{z}_{t+1}$ and $\mathcal{V} = \tilde{V}$ is a latent value head; in a state-space surrogate implementation, $\hat{y}_{t+1} = \hat{s}_{t+1}$ and $\mathcal{V} = V_\xi$ is PPO's critic. In either case, P_t measures how strongly local prediction uncertainty can perturb the advantage estimate. The gradient is taken with respect to the predicted next object, where advantage sensitivity is $\gamma \nabla_{\hat{y}} \mathcal{V}$; action gradients of Q or A concern a different perturbation.

For the count-based members, U_t is read off the schedules derived in Section 4. With $u_k(N) = 1/\sqrt{N+\epsilon}$, Eqs. (4.5) and (4.6) reduce to

$$U_t^{\text{UCB}} = \frac{1}{\sqrt{N_{\text{eff}}(c_t)+\epsilon}}, \qquad U_t^{\text{MCR}} = \frac{1}{\sqrt{N_{\text{eff}}(c_t)+\epsilon}} - \frac{1}{\sqrt{N_{\text{eff}}(c_t)+1+\epsilon}}, \tag{5.1}$$

where $N_{\text{eff}}(c_t)$ is the kNN-truncated effective count. Hence VALU-UCB rewards under-covered transitions, while VALU-MCR rewards transitions whose marginal sample most reduces the optimistic envelope. Following the modularity remark in Section 4.3, the count-based quantity can be replaced by a learned RND-style prediction error $U_t^{\text{RND}} = \left\| \hat{f}(s_t) - f(s_t) \right\|_2^2$, where f is a fixed random target network and $\hat{f}$ is a predictor trained on observations from the agent's rollouts; in practice this error is normalized (Appendix C.1). Given P_t and U_t, the VALU bonus is $b_t = \sigma_0 \,\text{clip}(P_t, 0, C)\, U_t$, where C is a numerical clipping threshold. We use the single effective scale $\alpha = \beta\sigma_0$, giving $\Delta_t := \beta b_t = \alpha \,\text{clip}(P_t, 0, C) U_t$; the original symbols distinguish their roles in the derivation.

5.2 Integration with PPO

VALU is added after each rollout has been collected and the standard GAE advantages have been computed. Following Eq. (3.3), we form

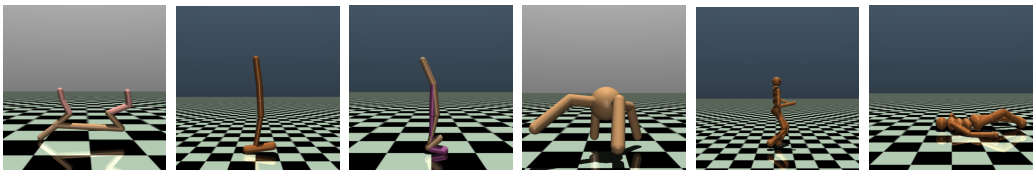
$$\hat{A}_t^{\text{VALU}} = \hat{A}_t + \Delta_t, \qquad \hat{R}_t^{\text{VALU}} = V_\xi(s_t) + \hat{A}_t^{\text{VALU}}, \tag{5.2}$$

The actor uses $\hat{A}_t^{\text{VALU}}$ in the clipped surrogate, and the critic is fitted to $\hat{R}_t^{\text{VALU}}$. The correction is added once after standard GAE, preserving task-reward bootstrapping without propagating future bonuses through the reward sequence. The optimizer, clipping rule, entropy term, and rollout collection are unchanged. Because PPO is on-policy, very old coverage points can become stale, so for count-based members we keep a fixed-capacity sliding memory bank $\mathcal{B}$, and N_{eff} is a recent-occupancy estimate rather than a lifetime visitation count.

Algorithm 1 summarizes the resulting training loop. Relative to vanilla PPO, VALU adds three operations: training the auxiliary latent interface, estimating the uncertainty quantity, and computing the value-priced advantage correction. The latent encoder (JEPA or VAE-style bottleneck) and the uncertainty estimator (effective counts, RND error) are independent design axes; we treat them as ablation variables in Section 6.

Algorithm 1 VALU+PPO

Initialize policy π_θ, critic V_ξ, latent interface $(h, g, \mathcal{V})$, optional memory bank $\mathcal{B}$, uncertainty estimator U, and effective scale α.
for iteration $= 1, 2, \ldots$ **do**
  Collect rollout $\{(s_t, a_t, r_t^e, s_{t+1})\}_{t=1}^T$ using π_θ.
  Compute standard GAE advantages $\hat{A}_t$ and value estimates $V_\xi(s_t)$.
  Train the auxiliary latent interface on the same rollout.
  Compute coverage points $c_t = h(s_t, a_t)$.
  Estimate uncertainty quantities U_t: use $N_{\text{eff}}(c_t)$ from $\mathcal{B}$ for count-based UCB/MCR, or $U_t^{\text{RND}} = \|\hat{f}(s_t) - f(s_t)\|_2^2$ for VALU-RND.
  Compute the value price $P_t = \|\nabla_{\hat{y}} \mathcal{V}(\hat{y}_{t+1})\|_2$.
  Form $\Delta_t = \alpha \,\text{clip}(P_t, 0, C) U_t$.
  Set $\hat{A}_t^{\text{VALU}} = \hat{A}_t + \Delta_t$ and $\hat{R}_t^{\text{VALU}} = V_\xi(s_t) + \hat{A}_t^{\text{VALU}}$.
  Update π_θ with the clipped surrogate using $\hat{A}_t^{\text{VALU}}$; fit V_ξ to $\hat{R}_t^{\text{VALU}}$.
  Update the uncertainty estimator: append c_t to $\mathcal{B}$ for count-based members, or train the RND predictor $\hat{f}$ on s_t.
end for



HALFCHEETAH HOPPER WALKER2D ANT HUMANOID HSTANDUP

Figure 2: Representative MuJoCo continuous-control environments used in our main experiments, covering locomotion and high-dimensional humanoid control.

6 Experiments

We design our experiments to evaluate VALU as a PPO-compatible exploration technique for continuous control. The goal is to test whether value-priced uncertainty improves PPO's sample efficiency and final performance, and whether the improvement comes from the two factors in VALU: an uncertainty quantity and a value-gradient price. We compare VALU with vanilla PPO and representative PPO-compatible exploration baselines, and then use ablations to study the structure of the method. Dense-reward MuJoCo [Todorov et al., 2012] tests exploration-budget allocation when informative rewards are available. Additional results on A2C, partially observed control, and Atari [Bellemare et al., 2013], together with independent validation and sparse-reward evaluations, are reported in Appendix C.

6.1 Experimental setup

Our main evaluation uses six standard MuJoCo continuous-control tasks: HALFCHEETAH-V4, HOPPER-V4, WALKER2D-V4, ANT-V4, HUMANOID-V4, and HUMANOIDSTANDUP-V4, covering locomotion and high-dimensional humanoid control. Figure 2 shows representative renderings of these environments. We use 1M environment steps for all tasks except HUMANOID-V4, where we use 2M steps.

We compare VALU with vanilla PPO and five PPO-compatible exploration baselines: RND [Burda et al., 2019], ICM [Pathak et al., 2017], RE3 [Seo et al., 2021], VCSE [Kim et al., 2023], and OPPO [Cai et al., 2020]. For each environment, all PPO-based methods use the same PPO hyperparameters, based on RL-Zoo [Raffin, 2020]; only method-specific exploration parameters are tuned. The VALU column reports a *family envelope*: the highest-mean member per task within the main-comparison candidate set, spanning effective-count UCB/MCR and RND-based uncertainty with the latent interfaces in Section 5. Selection and reporting use the same runs; this summary is not an independent estimate for a single fixed variant. Evaluation uses deterministic policies and extrinsic reward only. Experiments were run on NVIDIA RTX 4090 GPUs with 15 CPU cores per

Table 1: Final return on six standard MuJoCo continuous-control tasks. Each cell reports mean ± standard deviation over 10 random seeds. Bold marks the largest reported mean. VALU is the per-task family envelope within the main-comparison candidate set, selected on these same runs. Independent, fixed-variant results are reported in Table C.3.

Environment	VALU family envelope	PPO	RND	ICM	RE3	VCSE	OPPO
HALFCHEETAH	4378±280	4210±84	**4446±673**	3015±251	2209±132	3696±310	3964±347
HOPPER	**1197±421**	575±315	378±131	991±917	365±26	309±56	373±111
WALKER2D	**1553±448**	1259±636	612±170	451±162	420±23	512±141	588±59
ANT	**2800±376**	2259±168	1488±289	1048±227	1401±123	2107±146	2695±139
HUMANOID	**1037±159**	871±187	614±31	383±36	381±36	640±32	706±126
HUMANOIDSTANDUP	107323±21579	89909±10526	110786±36063	91278±30367	73511±11802	124824±19854	**138954±11749**

run. Additional implementation details, configuration comparisons, and experimental results are reported in Appendix C.

6.2 Main PPO results

Table 1 reports final evaluation return on the six main MuJoCo tasks. The selected VALU family members improve on vanilla PPO in mean return on all reported environments and attain the largest reported means in four of the six tasks: HOPPER, WALKER2D, ANT, and HUMANOID. On HALFCHEETAH, VALU is a close second to RND with less than half of its across-seed standard deviation. On HUMANOIDSTANDUP, VALU is not the top method, but it still substantially improves over PPO. Overall, the results show that value-priced uncertainty is a strong exploration signal for PPO across both medium-dimensional locomotion and high-dimensional humanoid control.

VALU is evaluated here as a method family rather than a single fixed bonus. On HALFCHEETAH, HOPPER, WALKER2D, ANT, and HUMANOID, the selected member is the count-based MCR configuration used as the reference in Table 3; these gains support the central mechanism of pricing uncertainty by its effect on the advantage. RND-based members apply the same price to a learned uncertainty estimate: unlike the RND baseline, which rewards prediction error directly as value-blind novelty, VALU-RND uses that error only as U_t and prices it by P_t.

Independent validation. To evaluate fixed variants separately from family selection, we selected α on three development seeds for each task–variant pair, froze it, and evaluated four variants and a rerun PPO baseline on ten fresh, disjoint validation seeds in a rebuilt software environment (Appendix C.4). JEPA–kNN has a higher mean than rerun PPO on all six tasks; other variants have task-dependent gains and losses. The appendix reports all four variants separately, together with sparse-reward tests and price diagnostics.

Figure 3 shows the learning curves on the same six tasks. The curves complement the final-return results in Table 1: VALU not only improves the final policy on most tasks, but also tends to reach useful behaviors earlier, especially on ANT and HUMANOIDSTANDUP. On HUMANOIDSTANDUP, OPPO obtains the highest final return, while VALU still learns rapidly.

6.3 Sample efficiency and ablations

We next evaluate sample efficiency against PPO's final return R_{PPO} in Table 1. If a method's final return is at least R_{PPO}, Table 2 reports the speedup T/t_m, where T is the training budget and t_m is the first evaluation step at which the method's mean curve reaches R_{PPO}. Otherwise, it reports in parentheses the reverse score t_{PPO}/T, where t_{PPO} is the first evaluation step at which PPO's mean curve reaches that method's final return; a smaller reverse score means that PPO matches that return earlier.

VALU reaches the PPO final-return threshold earlier than PPO on all six tasks and is the fastest method on five of them. The largest accelerations appear on ANT and HUMANOIDSTANDUP, where value-sensitive uncertainty helps the agent identify useful behaviors substantially earlier. On HALFCHEETAH, RND is slightly faster at reaching the PPO threshold. This suggests that raw prediction-error novelty can accelerate early learning in some environments, while value pricing gives a more consistent final outcome on HALFCHEETAH.

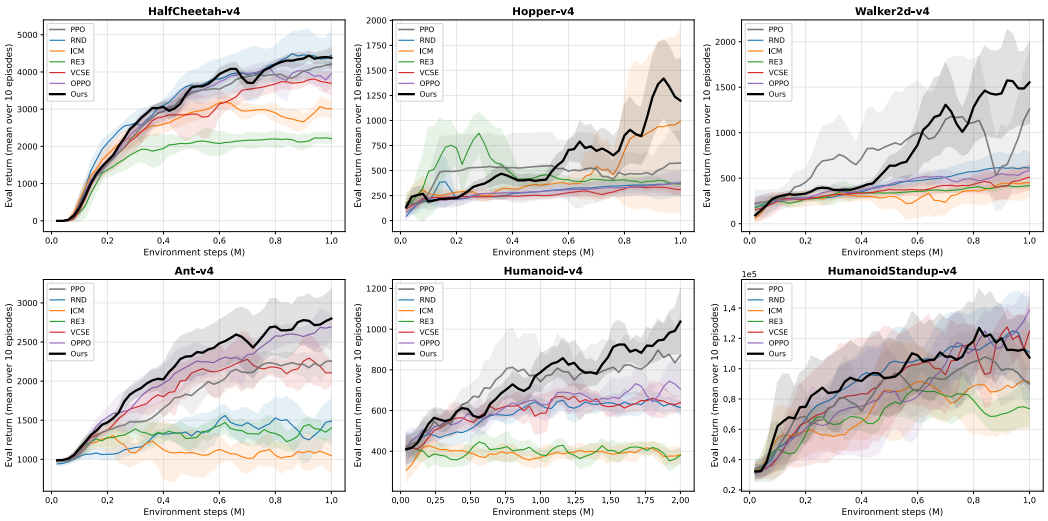


Figure 3: Learning curves on the six main MuJoCo tasks. Solid lines show mean evaluation return, and shaded regions show ± 1 standard deviation across seeds. VALU denotes the selected family member for each task in the main comparison, as in Table 1; independent validation runs are reported separately in Appendix C.4.

Table 2: Sample efficiency relative to vanilla PPO. Entries are speedups T/t_m; entries in parentheses are reverse scores t_{PPO}/T for methods whose final return is below PPO's (Section 6.3). Bold and underline mark the largest and second-largest speedups.

Environment	VALU	RND	ICM	RE3	VCSE	OPPO
HOPPER	**1.72**×	(0.14)	<u>1.25×</u>	(0.12)	(0.12)	(0.12)
WALKER2D	**1.43**×	(0.24)	(0.20)	(0.20)	(0.22)	(0.24)
ANT	**2.17**×	(0.34)	(0.10)	(0.24)	(0.64)	<u>1.67×</u>
HUMANOID	**1.32**×	(0.22)	(0.02)	(0.02)	(0.24)	(0.30)
HUMANOIDSTANDUP	**2.94**×	<u>2.63×</u>	1.67×	(0.28)	2.08×	1.52×
HALFCHEETAH	<u>1.22×</u>	**1.25**×	(0.42)	(0.30)	(0.66)	(0.90)

We ablate the structure of VALU in Table 3. This table focuses on the count-based subset of the VALU family, isolating the product structure, the UCB/MCR schedules, and the latent encoder choices. Each alternative row changes one factor relative to Reference, which combines the product bonus, MCR, and the environment-specific anchor encoder. Comparing Reference with density-only, gradient-only, and additive variants, the product form is strongest on every task except HALFCHEETAH, where it is on par with density-only, showing that the gain comes from pricing local uncertainty by value sensitivity rather than from either factor alone. The UCB and Alternative encoder rows test the schedule and representation choices. MCR is stronger than UCB on all five tasks, while JEPA and VAE each win on different environments. Together with the RND-based results in Appendix C.4, these ablations support the modular view of VALU: the uncertainty estimator supplies U_t, the value-gradient term supplies the price P_t, and the product determines how exploration pressure is assigned.

7 Discussion

Limitations and future directions. VALU relies on the latent geometry used to estimate local state–action coverage, and our ablations show that different encoders can be preferable in different environments. A promising direction is to couple value-priced exploration with stronger predictive world models, so that the latent space captures longer-horizon controllability and value-relevant

Table 3: Ablations on PPO for the count-based subset of VALU, with each shared reference shown once. Reference combines Product, MCR, and the anchor encoder: JEPA for HalfCheetah, Ant, and Humanoid, and VAE for Hopper and Walker2d. Alternative encoder uses the other encoder. Each other row changes one factor. Cells report mean ± standard deviation over random seeds.

Configuration	HALFCHEETAH	HOPPER	WALKER2D	ANT	HUMANOID
Reference	4378±280	1197±421	1553±448	2800±376	1037±159
Density-only	4390±159	504±183	1417±1306	2383±234	827±72
Gradient-only	4078±41	337±68	533±192	1818±694	805±123
Additive	4139±420	386±144	733±321	2167±509	753±53
UCB	4239±446	956±619	870±198	2247±254	787±26
Alternative encoder	4359±198	1004±1167	926±278	2800±432	820±77

dynamics beyond one-step transition similarity. On the theory side, our analysis derives VALU as a local advantage-uncertainty correction, but does not yet provide a global regret or sample-complexity guarantee. OPPO suggests one possible route: its regret analysis relies on a confidence set, an optimistic correction, and a cumulative bonus control argument. Extending this style of analysis to learned latent representations, non-parametric coverage estimates, and value-gradient pricing is an important open problem. VALU preserves PPO's discount, rollout horizon, and GAE convention, so local value sensitivity does not structurally shorten its decision horizon. The kNN member requires an explicit memory bank and can lose distance discrimination in high-dimensional spaces, motivating RND-based uncertainty for pixel observations. Finally, although this paper focuses on PPO, our additional A2C results suggest that value-priced uncertainty may apply more broadly to on-policy actor–critic methods and to higher-dimensional or partially observed tasks where latent dynamics models play a larger role.

Conclusion. We introduced VALU, a family of PPO-compatible exploration bonuses based on value-priced uncertainty. The contribution is conceptual as well as algorithmic: starting from a first-order advantage perturbation and a supremum over a confidence set, a single dual-norm template prescribes how any local uncertainty coordinate should be priced for exploration. OPPO falls out of this template under a feature/parameter confidence norm, and VALU-UCB and VALU-MCR are its latent-transition instances, where the dual sensitivity norm becomes the value-gradient norm. The same template accommodates a learned uncertainty estimator such as RND with the value-gradient price unchanged, and different latent encoders (JEPA and VAE) under the same bonus structure, making VALU a modular method family rather than a single fixed bonus. Independent validation on six dense-control tasks gives JEPA–kNN higher mean returns than rerun PPO, and component comparisons support combining uncertainty with value sensitivity. A pre-fixed JEPA–RND configuration yields higher mean returns than RND on three sparse tasks. We hope this perspective helps make intrinsic exploration in deep reinforcement learning more selective, stable, and value-relevant.

Acknowledgments and Disclosure of Funding

Kairui Feng and Qili Shen were supported by the National Natural Science Foundation of China (Grant No. T2600685), the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (Grant No. JYB2025XDXM908), and the Xiaomi Foundation. Dake Zhang was supported by the National Natural Science Foundation of China (Grant No. 72603219) and the Shanghai Pujiang Program (Grant No. 25PJC049). We also thank the Shanghai Innovation Institute. The authors declare no competing interests.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pages 29304–29320, 2021.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15619–15629, 2023.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, 2002.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The Arcade Learning Environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, volume 29, pages 1471–1479, 2016.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pages 1283–1294. PMLR, 2020.

Amir-massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1486–1494. PMLR, 2017.

Sebastian Flennerhag, Jane X. Wang, Pablo Sprechmann, Francesco Visin, Alexandre Galashov, Steven Kapturowski, Diana L. Borsa, Nicolas Heess, Andre Barreto, and Razvan Pascanu. Temporal difference uncertainties as a signal for exploration. *arXiv preprint arXiv:2010.02255*, 2020.

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Matteo Hessel, Ian Osband, Alex Graves, Volodymyr Mnih, Rémi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration. In *International Conference on Learning Representations*, 2018.

Peter I. Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Peter I. Frazier, Warren B. Powell, and Savas Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439, 2008.

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*, pages 1861–1870. PMLR, 2018.

Dongyoung Kim, Jinwoo Shin, Pieter Abbeel, and Younggyo Seo. Accelerating reinforcement learning with value-conditional state entropy exploration. In *Advances in Neural Information Processing Systems*, volume 36, pages 31811–31830, 2023.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

L. F. Kozachenko and N. N. Leonenko. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(6):066138, 2004.

Yann LeCun. A path towards autonomous machine intelligence. OpenReview, 2022. URL `https://openreview.net/forum?id=BZ5a1r-kVsf`.

Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. In *Advances in Neural Information Processing Systems*, volume 34, pages 18459–18473, 2021.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937. PMLR, 2016.

Philippe Morere and Fabio Ramos. Bayesian RL for goal-only rewards. In *Conference on Robot Learning*, pages 386–398. PMLR, 2018.

Brendan O'Donoghue, Ian Osband, Rémi Munos, and Volodymyr Mnih. The uncertainty Bellman equation and exploration. In *International Conference on Machine Learning*, pages 3839–3848. PMLR, 2018.

Georg Ostrovski, Marc G. Bellemare, Aäron van den Oord, and Rémi Munos. Count-based exploration with neural density models. In *International Conference on Machine Learning*, pages 2721–2730. PMLR, 2017.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, pages 2778–2787. PMLR, 2017.

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. In *International Conference on Learning Representations*, 2018.

Antonin Raffin. RL Baselines3 Zoo. `https://github.com/DLR-RM/rl-baselines3-zoo`, 2020.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22 (268):1–8, 2021.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. State entropy maximization with random encoders for efficient exploration. In *International Conference on Machine Learning*, pages 9443–9454. PMLR, 2021.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*, pages 1015–1022, 2010.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. #Exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, pages 2753–2762, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

Claas Voelcker, Victor Liao, Animesh Garg, and Amir-massoud Farahmand. Value gradient weighted model-based reinforcement learning. In *International Conference on Learning Representations*, 2022.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pages 11920–11931. PMLR, 2021.

A Table of notations

The following table provides a reference for the key symbols and notations used throughout this paper.

Table A.1: Notation reference.

Symbol	Description
MDP and rollout data	
$s,\ a,\ s'$	State, action, next state
$s_t,\ a_t,\ s_{t+1}$	State, action, and next state at rollout step t
r_t^e	Extrinsic environment reward at step t
γ	Discount factor
λ	GAE bootstrapping coefficient
T	Rollout horizon / total environment steps
$t,\ k,\ l$	Rollout step index, training-iteration index, GAE summation index
Policy, critic, and PPO	
π_θ	Stochastic policy with parameters θ
$\pi_{\theta_{\text{old}}}$	Behaviour policy used to collect the current rollout
V_ξ	PPO critic with parameters ξ
$\rho_t(\theta)$	PPO probability ratio $\pi_\theta(a_t \mid s_t)/\pi_{\theta_{\text{old}}}(a_t \mid s_t)$
$\mathcal{L}^{\text{CLIP}}(\theta)$	PPO clipped surrogate objective
δ_t	Temporal-difference residual $r_t^e + \gamma V_\xi(s_{t+1}) - V_\xi(s_t)$
$\hat{A}_t$	GAE advantage estimate at step t
$\hat{A}_k(s,a)$	Nominal one-step advantage at iteration k, $r(s,a) + \gamma\tilde{V}_k(\hat{z}') - \tilde{V}_k(z)$
$\mathbb{E}_t[\cdot]$	Empirical expectation over rollout steps
Latent interface and encoders	
$z,\ z_t$	Latent state of s and s_t
g_ϕ	Action-conditioned next-latent / next-state predictor with parameters ϕ
$\hat{z}',\ \hat{z}_{t+1}$	Predicted next latent $\hat{z}_{t+1} = g_\phi(z_t, a_t)$
$\hat{y}_{t+1}$	Generic predicted next object: a next latent $\hat{z}_{t+1}$ or a next state $\hat{s}_{t+1}$
$\tilde{V}_\psi,\ \tilde{V}_k$	Latent value head with parameters ψ (subscript k = its iterate-k snapshot)
$\mathcal{V}$	Generic value function used to price uncertainty: either the latent head $\tilde{V}$ or PPO's critic V_ξ
Coverage points and uncertainty quantities	
h_η	State–action coverage map; $c_t = h_\eta(s_t, a_t)$
$c,\ c_t,\ c_i$	State–action coverage points stored in the memory bank
$\mathcal{B}$	Fixed-capacity sliding memory bank $\mathcal{B} = \{c_i\}_{i=1}^N$
$K_h(\cdot,\cdot)$	kNN-truncated kernel of bandwidth h
h	kNN kernel bandwidth (set to the median pairwise latent distance)
$N_{\text{eff}}(c),\ N_{\text{eff}}(z,a)$	Effective sample count around c in $\mathcal{B}$
ϵ	Small constant inside $\sqrt{N_{\text{eff}} + \epsilon}$ (distinct from the PPO clip ε)
$U_t,\ U(s,a)$	Generic uncertainty quantity priced by the value gradient
U_t^{UCB}	UCB count-based uncertainty $1/\sqrt{N_{\text{eff}}(c_t) + \epsilon}$
U_t^{MCR}	MCR count-based uncertainty schedule
f	Fixed random target network used by RND
$\hat{f}$	Trainable RND predictor matching f on the agent's observations
$U_t^{\text{RND}},\ U_{\text{RND}}(s)$	RND prediction-error uncertainty $\|\hat{f}(s) - f(s)\|_2^2$
$\text{norm}(\cdot)$	Per-batch L^2 or running-std normalisation of the RND error
Value pricing and the VALU bonus	
$P_t,\ P(s,a)$	Value-gradient price $\|\nabla_{\hat{y}}\mathcal{V}(\hat{y}_{t+1})\|_2$
$\nabla_{z'}\tilde{V}_\psi(\hat{z}'),\ \nabla_{\hat{y}}\mathcal{V}(\hat{y}_{t+1})$	Local value gradient at the predicted next latent / next object
C	Numerical clipping threshold applied to P_t
$\text{clip}(P_t, 0, C)$	Element-wise clipping of P_t to $[0, C]$

Symbol	Description (continued)
b_t, $b(s,a)$	VALU exploration bonus $\sigma_0 \operatorname{clip}(P_t, 0, C)\, U_t$
σ_0	Bonus prior scale absorbed into b_t
β	Correction weight in the derivation; used through $\alpha = \beta\sigma_0$
α	Single effective bonus scale, $\beta\sigma_0$
Δ_t	Post-GAE correction, $\alpha \operatorname{clip}(P_t, 0, C) U_t$
$\hat{A}_t^{\mathrm{VALU}}$	VALU-corrected advantage $\hat{A}_t + \beta b_t$
$\hat{R}_t^{\mathrm{VALU}}$	VALU critic target $V_\xi(s_t) + \hat{A}_t^{\mathrm{VALU}}$
Theory: confidence sets, perturbations, and norms	
ξ	Generic local uncertainty coordinate in the unified template
$\delta z'$, δ	Latent transition perturbation, $z' = \hat{z}' + \delta z'$
$A_k(s,a;\xi)$	First-order perturbed advantage $\hat{A}_k(s,a) + \langle G_k(s,a), \xi\rangle + o(\|\xi\|)$
$G_k(s,a)$	Sensitivity vector in the unified template
$G_k^{\mathrm{VALU}}(s,a)$	VALU-UCB sensitivity $\gamma \nabla_{z'} \tilde{V}_k(\hat{z}')$
$\mathcal{U}_k(s,a)$, $\mathcal{U}_k^{\mathrm{VALU}}(s,a)$	Local confidence set $\{\xi : \|\xi\|_{\mathcal{B}} \le r_k(s,a)\}$ (resp. for latent perturbations)
$r_k(s,a)$	Confidence radius of $\mathcal{U}_k(s,a)$
$u_k(N)$	Count-based confidence-radius schedule, here $u_k(N) = 1/\sqrt{N+\epsilon}$
$A_k^+(s,a)$	Optimistic advantage $\sup_{\xi \in \mathcal{U}_k} A_k(s,a;\xi)$
$A_{k,N}^+(s,a)$	Optimistic advantage at effective count N
$R_k(N;s,a)$	Optimism radius $\gamma u_k(N) \|\nabla_{z'} \tilde{V}_k(\hat{z}')\|_{\mathcal{B}^*}$
$\mathcal{I}_k(s,a)$	One-sample shrinkage of the optimistic envelope
η	Constraint level in the MCR Lagrangian formulation ; distinct from encoder parameters η
$\|\cdot\|_{\mathcal{B}}$	Confidence norm defining $\mathcal{U}_k$
$\|\cdot\|_{\mathcal{B}^*}$	Dual norm of $\|\cdot\|_{\mathcal{B}}$ used in the optimistic correction
$\|\cdot\|_2$, $\|\cdot\|_\infty$	Euclidean and supremum norms
$\|\cdot\|_\Lambda$	Mahalanobis norm $\sqrt{(\cdot)^\top \Lambda (\cdot)}$
OPPO-specific symbols	
w, w_k	Linear value parameters and their iterate-k estimate, $Q_w(s,a) = \phi(s,a)^\top w$
Δw	Parameter error $w - w_k$ (the OPPO uncertainty coordinate)
$\phi(s,a)$, $\phi_k(s,a)$	Linear feature map and its iterate-k evaluation
Λ_k	Least-squares Gram matrix $\lambda I + \sum_{i=1}^n \phi_i \phi_i^\top$
β_k	OPPO confidence-ellipsoid radius $\|\Delta w\|_{\Lambda_k} \le \beta_k$
$G_k^{\mathrm{OPPO}}(s,a)$	OPPO sensitivity, $\phi_k(s,a)$

B Additional theory details

This appendix gives derivations omitted from Section 4. We show how UCB-style optimistic advantage corrections, OPPO, and VALU schedules arise from a unified advantage-uncertainty view.

B.1 Unified optimistic advantage correction

Let ξ denote a local uncertainty coordinate and assume $A_k(s,a;\xi) = \hat{A}_k(s,a) + \langle G_k(s,a), \xi\rangle + o(\|\xi\|)$. For a confidence set $\mathcal{U}_k(s,a) = \{\xi : \|\xi\|_{\mathcal{B}} \le r_k(s,a)\}$, the optimistic advantage is

$$A_k^+(s,a) = \sup_{\xi \in \mathcal{U}_k(s,a)} A_k(s,a;\xi) \approx \hat{A}_k(s,a) + r_k(s,a)\|G_k(s,a)\|_{\mathcal{B}^*}, \tag{B.1}$$

by dual normity. Examples: $\|\xi\|_2$ gives $\sup = r\|G\|_2$, $\|\xi\|_\infty$ gives $\sup = r\|G\|_1$, and a Mahalanobis ball $\|\xi\|_\Lambda \le r$ gives $\sup = r\sqrt{G^\top \Lambda^{-1} G}$.

B.2 VALU-UCB derivation

VALU-UCB places the uncertainty coordinate in latent transition space. Let $z' = \hat{z}' + \delta z'$. The one-step advantage under this perturbation is $A_k(s,a;\delta z') = r(s,a) + \gamma \tilde{V}_k(\hat{z}' + \delta z') - \tilde{V}_k(z)$. Define the nominal advantage $\hat{A}_k(s,a) = r(s,a) + \gamma \tilde{V}_k(\hat{z}') - \tilde{V}_k(z)$. A first-order expansion of $\tilde{V}_k$ around $\hat{z}'$ gives

$$\tilde{V}_k(\hat{z}' + \delta z') = \tilde{V}_k(\hat{z}') + \nabla_{z'} \tilde{V}_k(\hat{z}')^\top \delta z' + o(\|\delta z'\|). \tag{B.2}$$

Thus,

$$A_k(s,a;\delta z') = \hat{A}_k(s,a) + \gamma \nabla_{z'} \tilde{V}_k(\hat{z}')^\top \delta z' + o(\|\delta z'\|). \tag{B.3}$$

Therefore the sensitivity vector in the unified template is $G_k^{\mathrm{VALU}}(s,a) = \gamma \nabla_{z'} \tilde{V}_k(\hat{z}')$. Let the latent confidence set be $\mathcal{U}_k^{\mathrm{VALU}}(s,a) = \{\delta z' : \|\delta z'\|_{\mathcal{B}} \le u_k(N_{\mathrm{eff}}(z,a))\}$. Using the unified correction, $A_k^{\mathrm{VALU-UCB}}(s,a) = \hat{A}_k(s,a) + \gamma u_k(N_{\mathrm{eff}}(z,a))\|\nabla_{z'} \tilde{V}_k(\hat{z}')\|_{\mathcal{B}^*}$. If $\|\cdot\|_{\mathcal{B}} = \|\cdot\|_2$, the dual norm is also L_2, and

$$A_k^{\mathrm{VALU-UCB}}(s,a) = \hat{A}_k(s,a) + \gamma u_k(N_{\mathrm{eff}}(z,a))\|\nabla_{z'} \tilde{V}_k(\hat{z}')\|_2. \tag{B.4}$$

Choosing $u_k(N) = \frac{1}{\sqrt{N+\epsilon}}$ gives the UCB-level schedule $b_{\mathrm{UCB}}(s,a) = \frac{\|\nabla_{z'} \tilde{V}_k(\hat{z}')\|_2}{\sqrt{N_{\mathrm{eff}}(z,a)+\epsilon}}$, up to constants absorbed into the bonus scale.

B.3 OPPO as a parameter-space instance

OPPO can be written as the same optimistic advantage correction with a different uncertainty coordinate. In a linear value or Q-function model, $Q_w(s,a) = \phi(s,a)^\top w$. Let w_k be the current parameter estimate and let $\Delta w = w - w_k$ be the parameter error. The perturbed value object has the first-order form $A_k(s,a;\Delta w) \approx \hat{A}_k(s,a) + \phi_k(s,a)^\top \Delta w$. Thus,$G_k^{\mathrm{OPPO}}(s,a) = \phi_k(s,a)$.

OPPO uses a least-squares confidence ellipsoid:

$$\mathcal{U}_k^{\mathrm{OPPO}} = \{\Delta w : \|\Delta w\|_{\Lambda_k} \le \beta_k\}, \tag{B.5}$$

where $\|\Delta w\|_{\Lambda_k} = \sqrt{\Delta w^\top \Lambda_k \Delta w}$, $\Lambda_k = \lambda I + \sum_{i=1}^n \phi_i \phi_i^\top$. The dual norm of $\|\cdot\|_{\Lambda_k}$ is $\|\cdot\|_{\Lambda_k^{-1}}$. Therefore,$\sup_{\|\Delta w\|_{\Lambda_k} \le \beta_k} \phi_k(s,a)^\top \Delta w = \beta_k \|\phi_k(s,a)\|_{\Lambda_k^{-1}}$. Hence,

$$A_k^{\mathrm{OPPO}}(s,a) = \hat{A}_k(s,a) + \beta_k \|\phi_k(s,a)\|_{\Lambda_k^{-1}}. \tag{B.6}$$

This shows that OPPO and VALU-UCB share the same optimistic-advantage template. OPPO places the confidence set on linear value parameters and uses a Mahalanobis confidence norm determined by least-squares covariance. VALU-UCB places the confidence set on the latent transition and uses a non-parametric local uncertainty radius together with a value-gradient sensitivity norm.

B.4 MCR from an information-constrained control problem

UCB-style optimism answers the question: *how large could the current advantage be under the uncertainty set?* MCR answers a different question: *how much can one more sample reduce the uncertainty in the optimistic advantage?*

Let $\mathcal{I}_k(s,a)$ denote the information gain from sampling (s,a). We formulate action selection as

$$\max_a \hat{A}_k(s,a) \quad \text{s.t.} \quad \mathcal{I}_k(s,a) \geq \eta. \tag{B.7}$$

The Lagrangian is $\mathcal{L}(a,\beta) = \hat{A}_k(s,a) + \beta(\mathcal{I}_k(s,a) - \eta)$. Since $-\beta\eta$ does not depend on a, maximizing the Lagrangian over a is equivalent to $\max_a \left[\hat{A}_k(s,a) + \beta\mathcal{I}_k(s,a)\right]$. Thus, the information gain appears as an advantage correction.

We define $\mathcal{I}_k(s,a)$ as the one-sample shrinkage of the optimistic advantage envelope: $\mathcal{I}_k(s,a) = A^+_{k,N}(s,a) - A^+_{k,N+1}(s,a)$, where $N = N_{\text{eff}}(z,a)$. Let $A^+_{k,N}(s,a) = \hat{A}_k(s,a) + R_k(N;s,a)$. Then $\mathcal{I}_k(s,a) = R_k(N;s,a) - R_k(N+1;s,a)$. For the VALU latent transition confidence set, $R_k(N;s,a) = \gamma u_k(N)\|\nabla_{z'}\tilde{V}_k(\hat{z}')\|_{\mathcal{B}^*}$. Therefore, $\mathcal{I}_k(s,a) = \gamma\left[u_k(N) - u_k(N+1)\right]\|\nabla_{z'}\tilde{V}_k(\hat{z}')\|_{\mathcal{B}^*}$. With $u_k(N) = \frac{1}{\sqrt{N+\epsilon}}$, we obtain $\mathcal{I}_k(s,a) = \gamma\left[\frac{1}{\sqrt{N+\epsilon}} - \frac{1}{\sqrt{N+1+\epsilon}}\right]\|\nabla_{z'}\tilde{V}_k(\hat{z}')\|_{\mathcal{B}^*}$.

For the Euclidean norm, this gives

$$b_{\text{MCR}}(s,a) = \|\nabla_{z'}\tilde{V}_k(\hat{z}')\|_2\left[\frac{1}{\sqrt{N_{\text{eff}}(z,a)+\epsilon}} - \frac{1}{\sqrt{N_{\text{eff}}(z,a)+1+\epsilon}}\right], \tag{B.8}$$

up to constants absorbed into the bonus scale.

Thus MCR is not simply another hand-designed schedule. It is the Lagrangian bonus induced by requiring the selected action to provide enough shrinkage of the optimistic advantage envelope.

C Experimental supplementary

This appendix provides implementation details and additional experimental results for Section 6. We first describe the shared training protocol and hyperparameter choices. We then compare critic-side stability across exploration methods and evaluate whether the PPO-side conclusions transfer to A2C. Finally, we report additional experiments: independent dense-control validation, sparse-reward and observation-structure evaluations, and local price diagnostics. These experiment sets are reported separately from the main comparison and are not pooled.

C.1 Implementation and hyperparameters

The PPO and A2C runs in the main comparison use hyperparameters based on RL-Zoo [Raffin, 2020] for the corresponding environment. We keep these base hyperparameters fixed across vanilla agents and exploration baselines, and only tune method-specific exploration parameters. The latent interface uses either a JEPA-style encoder or a VAE-style bottleneck encoder (Section 5). For the count-based VALU members in the main comparison, we search the effective scale $\alpha \in \{0.01, 0.05, 0.1, 0.5\}$.

For the RND member, the uncertainty quantity is the normalized prediction error $U_t = \text{norm}(\|\hat{f}(s_t) - f(s_t)\|_2^2)$, where f is a fixed random network and $\hat{f}$ is a learned predictor [Burda et al., 2019]. We search prediction-error multipliers $\{0.05, 0.1, 0.5\}$ with either per-batch L^2 or running-standard-deviation normalization, selecting a family member per environment. In the diagnostic tables, VALU+MCR denotes the count-based MCR member and VALU+RND the prediction-error member. The kNN variants of the independent validation are reported separately in Appendix C.4.

The kNN coverage estimator uses $k = 10$, and the kernel bandwidth h is set to the median pairwise distance of latent points in the current rollout buffer. Each run uses 4 vectorized environments (DUMMYVECENV from Stable-Baselines3 [Raffin et al., 2021]) and is trained for 1M environment steps, except HUMANOID-V4, which uses 2M steps. Evaluation is performed with deterministic

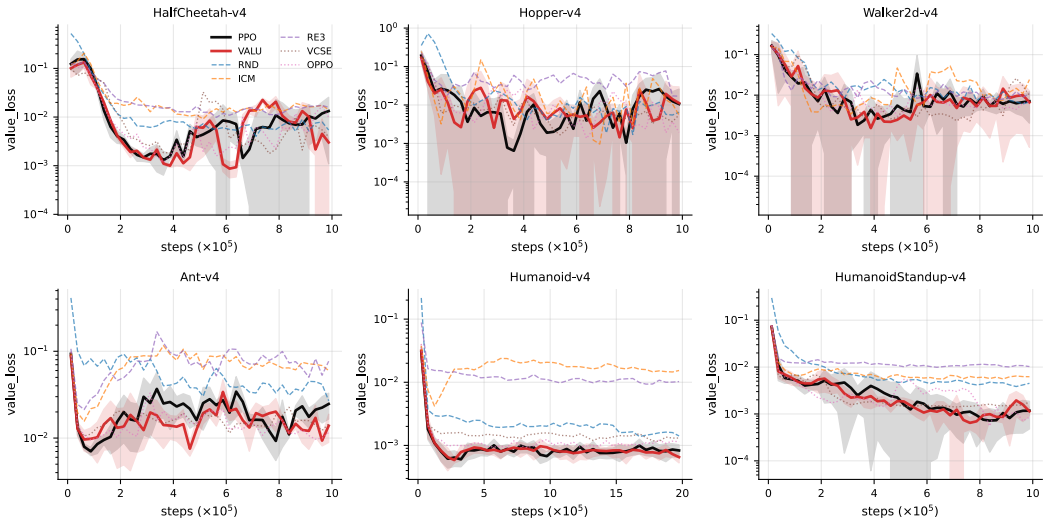


Figure C.1: Per-rollout `value_loss` for PPO (black solid) and VALU (red solid) versus reward-augmenting baselines RND, ICM, RE3 (dashed) and advantage-side baselines VCSE, OPPO (dotted) on six MuJoCo environments. VALU's critic-loss scale remains relatively close to PPO; the reward-augmenting methods have larger training losses, most visibly on ANT, HUMANOID, and HUMANOIDSTANDUP. Quantitative ratios are given in Table C.1.

policies and extrinsic reward only, using 10 evaluation episodes per checkpoint and approximately 50 checkpoints per run. Unless otherwise noted, each configuration in the main comparison is evaluated over 10 random seeds. Experiments are run on a single NVIDIA RTX 4090 with 15 CPU cores per run; a typical 1M-step VALU run takes 5–7 hours depending on the environment.

C.2 Critic-side stability

A natural concern when adding a multiplicative exploration bonus to PPO is whether the bonus disturbs critic learning, either by inflating returns or by injecting non-stationary value targets. VALU adds the current transition's correction once to both the actor advantage and critic return target after extrinsic-reward GAE (Section 5.2). We examine this by comparing PPO's training-side critic statistics across all six exploration methods (Figure C.1 and Table C.1); the evolution of the value price itself is reported in Appendix C.7.

The qualitative picture in Figure C.1 is made quantitative by the time-averaged ratios in Table C.1. VALU+MCR stays within $\pm 23\%$ of vanilla PPO and has a lower training loss on four of six environments; VALU+RND ranges from $0.56\times$ to $1.51\times$. RND, ICM, and RE3 range from approximately $1.5\times$ to $9.8\times$. These ratios describe training-loss scales: the methods use different targets, so the ratios alone do not measure accuracy against a common extrinsic-value target. VALU's post-GAE correction changes the critic target locally without propagating a sequence of intrinsic rewards through GAE.

C.3 Generalization to A2C

VALU is derived as a local optimistic-advantage correction (Section 4) and does not depend on PPO's clipped surrogate. We therefore expect it to extend to other on-policy actor–critic methods that use an advantage estimate. To verify this we run the same VALU implementation on A2C with the only change being the underlying optimizer loop (A2C hyperparameters based on RL-Zoo, otherwise identical). Results on the seven A2C-applicable MuJoCo tasks are reported in Table C.2.

The PPO-side improvement over the base agent broadly transfers: VALU has a higher mean than vanilla A2C on six tasks and ties it on INVERTEDDOUBLEPENDULUM. Among the compared methods, VALU has the highest mean on ANT ($+47\%$ over A2C and $+28\%$ over OPPO) and ties for

Table C.1: Time-averaged `train/value_loss` relative to vanilla PPO on the six MuJoCo environments. Ratios compare each method's own training loss with vanilla PPO's; they do not imply identical regression targets. VALU+MCR has a lower training loss than vanilla PPO on four environments.

Environment	VALU+MCR	VALU+RND	RND	ICM	RE3	VCSE	OPPO
HALFCHEETAH	$0.98\times$	$1.04\times$	$2.35\times$	$2.04\times$	$1.97\times$	$1.19\times$	$0.99\times$
HOPPER	$1.12\times$	$1.51\times$	$4.11\times$	$2.03\times$	$4.13\times$	$0.96\times$	$1.55\times$
WALKER2D	$1.00\times$	$1.51\times$	$3.07\times$	$1.79\times$	$1.46\times$	$1.19\times$	$0.82\times$
ANT	$0.77\times$	$0.56\times$	$3.48\times$	$2.92\times$	$3.55\times$	$1.10\times$	$0.75\times$
HUMANOID	$0.94\times$	$1.07\times$	$4.30\times$	$9.75\times$	$8.33\times$	$1.15\times$	$1.13\times$
HUMANOIDSTANDUP	$0.87\times$	$0.88\times$	$2.98\times$	$2.01\times$	$3.30\times$	$1.11\times$	$1.04\times$

Table C.2: Final return on MuJoCo continuous-control tasks under *A2C*. Bold marks the best A2C-compatible method.

Environment	VALU	A2C	RND	ICM	RE3	VCSE	OPPO
ANT	**2150±687**	1467±275	1228±105	1104±218	894±193	1463±297	1675±329
HALFCHEETAH	3034±651	2822±1706	2096±1635	−252±164	2137±1160	**3222±1146**	2733±1225
HOPPER	538±237	197±23	142±35	315±170	372±69	**826±249**	428±256
HUMANOID	629±43	584±59	**662±124**	453±39	436±26	493±52	509±15
INVERTEDDOUBLEPENDULUM	**9360±0**	**9360±0**	9359±0	220±100	9359±1	**9360±0**	**9360±0**
PUSHER	−52±1	−54±3	**−50±3**	−64±17	**−50±4**	−54±3	−52±3
WALKER2D	561±407	352±119	401±197	220±64	282±39	**563±415**	470±234

the highest on INVERTEDDOUBLEPENDULUM. VCSE has the highest mean on HALFCHEETAH, HOPPER, and WALKER2D; RND leads on HUMANOID, and RND/RE3 tie on PUSHER. We did not evaluate A2C on HUMANOIDSTANDUP. These results support applying value-priced uncertainty to other on-policy actor–critic methods, with task-dependent relative performance.

C.4 Independent dense-control validation

For each environment–variant pair, we selected α by mean return on three development seeds and froze it before evaluating on ten fresh, disjoint validation seeds. The four variants $\{\mathrm{JEPA}, \mathrm{VAE}\} \times \{\mathrm{kNN}, \mathrm{RND}\}$ were fixed before validation, with PPO rerun under the same protocol. Table C.3 reports every variant separately; validation returns were not used for selection. These validation runs, including PPO, were run from scratch in a rebuilt software environment and are separate from the main comparison in Table 1, Figure 3, and Table 2. Absolute returns, including PPO's, therefore differ between the two sets, and methods are compared only within each set.

JEPA–kNN has a higher mean than rerun PPO on all six tasks; VAE–kNN has the highest mean on Ant. Other variants show task-dependent gains and losses. JEPA–kNN's validation cross-task IQM [Agarwal et al., 2021] relative to PPO is 1.15 (95% confidence interval $[1.11, 1.20]$).

Per-task tests. Table C.4 reports unadjusted, two-sided Welch tests for JEPA–kNN versus rerun PPO.

C.5 Sparse-reward evaluation

JEPA–RND with $\alpha = 0.05$ was fixed before observing sparse-domain results and applied to all three tasks without per-environment tuning. Gravitar uses 10M training steps; SparseHalfCheetah and SparseHopper use 1M. Table C.5 reports five-seed results.

VALU yields slightly higher mean returns than RND, with modest margins relative to seed variation. These results support compatibility with prediction-based novelty in sparse settings, without establishing a statistical ranking over RND.

Table C.3: Independent validation on the six dense-control tasks. Each column uses ten validation seeds; four variants are reported without a family envelope. Entries are mean ± standard deviation across training seeds. Bold indicates the largest mean within a row.

Environment	PPO	JEPA–kNN	JEPA–RND	VAE–kNN	VAE–RND
HALFCHEETAH	3964 ±159	**4466 ±376**	4210 ±182	4270 ±223	4044 ±249
HOPPER	461 ±115	**974 ±448**	408 ±129	486 ±167	569 ±278
WALKER2D	570 ±119	**1341 ±1074**	958 ±381	1174 ±876	1156 ±576
ANT	2725 ±231	2865 ±116	2756 ±113	**2944 ±206**	2683 ±193
HUMANOID	791 ±28	**883 ±79**	864 ±70	753 ±71	760 ±80
HUMANOID STANDUP	123574 ±14049	**141925 ±14647**	119512 ±8636	132501 ±10569	115751 ±17817

Table C.4: Unadjusted two-sided Welch tests for JEPA–kNN versus the rerun PPO baseline, using the ten-seed validation results.

Environment	t	p
HALFCHEETAH	3.88	0.002
HOPPER	3.50	0.006
WALKER2D	2.26	0.050
ANT	1.71	0.110
HUMANOID	3.46	0.005
HUMANOID STANDUP	2.86	0.010

C.6 Partially observed and visual control

The NoVel evaluations remove velocity observations. Returns improve from 794 ± 141 to 1081 ± 344 on HopperNoVel-v4 and from 701 ± 111 to 830 ± 263 on Walker2dNoVel-v4 (approximately 36% and 18%); $\pm$ denotes standard deviation across training seeds.

Table C.6 reports Alien and MsPacman results with five seeds and 10M steps per game. The VALU column reports per-game family envelopes. These tasks test visual control; the fixed JEPA–RND evaluation on Gravitar above addresses sparse rewards.

C.7 Value-gradient diagnostics

Finite-difference probe. Every ten rollouts, we sample 256 predicted next latents from on-policy data and compare finite-difference critic changes with gradient-based estimates at perturbation scales $\kappa \in \{0.01, 0.1, 0.5\}$:

$$D_{\mathrm{FD}} = |\tilde{V}(z+\Delta) - \tilde{V}(z)|, \qquad D_{\mathrm{price}} = \|\nabla_z \tilde{V}(z)\|_2 \|\Delta\|_2, \qquad \Delta = \kappa\, \sigma_z \odot \nu. \tag{C.1}$$

Here $\tilde{V}$ is the learned latent critic, with scaled perturbation $\sigma_z \odot \nu$. Across 72 cells (six environments, two VALU members, three scales, and early/late training), Pearson correlations range from 0.18 to 0.57. At the two smaller scales, all twelve environment–member combinations improve from early to late training.

Price activity. Table C.7 shows nonzero median prices in early and late training, with task-dependent evolution.

Table C.5: Sparse-domain results with one pre-fixed JEPA–RND configuration, $\alpha = 0.05$, and five seeds. Entries are mean $\pm$ standard deviation across training seeds.

Environment (budget)	PPO	RND	VALU JEPA–RND
Gravitar (10M)	690 ± 159	801 ± 74	853 ± 145
SparseHalfCheetah (1M)	627 ± 253	841 ± 290	875 ± 182
SparseHopper (1M)	84 ± 38	90 ± 52	97 ± 17

Table C.6: Visual-control results at 10M steps: mean $\pm$ standard deviation across five training seeds. The last column is a per-game family envelope.

Game	PPO	RND	ICM	RE3	OPPO	VALU envelope
Alien	1638 ±347	1757 ±566	1522 ±309	1862 ±476	1664 ±572	1944 ±704
MsPacman	1774 ±475	2021 ±512	1907 ±735	1778 ±895	2318 ±983	2340 ±308

Table C.7: Median value prices P_{50} in early and late training. Arrows indicate early $\rightarrow$ late.

Environment	VALU–RND	VALU–MCR
HalfCheetah	$0.661 \rightarrow 0.239$	$0.476 \rightarrow 0.204$
Hopper	$1.150 \rightarrow 0.798$	$0.945 \rightarrow 0.685$
Walker2d	$1.190 \rightarrow 1.010$	$1.450 \rightarrow 0.763$
Ant	$0.823 \rightarrow 0.993$	$0.784 \rightarrow 0.959$
Humanoid	$0.209 \rightarrow 0.162$	$0.214 \rightarrow 0.154$
HumanoidStandup	$0.459 \rightarrow 0.576$	$0.445 \rightarrow 0.617$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main methodological claim of VALU as a PPO-compatible value-priced uncertainty bonus and describe its theoretical derivation from advantage uncertainty.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We address our limitations in the Discussion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Section 4 states the modelling assumptions (a first-order expansion of the advantage and a norm-ball confidence set) and derives each bonus from them. Full derivations are provided in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The method section specifies the full VALU training loop, the PPO integration, the effective-count memory bank, the value-gradient price, and the UCB/MCR schedules. The experimental section and appendix report the MuJoCo tasks, baselines, ablations, and implementation details needed to reproduce the main conclusions.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The experiments use standard MuJoCo benchmark environments and the algorithmic details are described in the paper.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper describes the benchmark suite, PPO-compatible baselines, VALU variants, ablation axes, and the main implementation choices. Additional hyperparameters and implementation details are provided in the supplemental material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The experimental results are reported over multiple random seeds with shaded error regions or error bars.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We address this point in the Experimental section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work studies a reinforcement-learning exploration algorithm on standard simulated continuous-control benchmarks. It does not involve human subjects, private data, web-scraped data, or high-risk deployed systems.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

Justification: This is foundational research on exploration in reinforcement learning, evaluated only on simulated benchmarks. It is not tied to a particular application or deployment, and we do not foresee a direct path to negative societal impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper poses no such risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites the existing algorithms, benchmark environments, and representation-learning methods used for comparison or implementation.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method does not use LLMs as an algorithmic component, experimental tool, or source of scientific results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.