GENERATIVE BELLMAN FLOW: MITIGATING THE CVAR GRADIENT BOTTLENECK IN RISK-SENSITIVE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Distributional reinforcement learning provides the expressivity to model full return distributions, yet this has not consistently translated into reliable tail-risk policy optimization in continuous control. We identify a CVaR actor-gradient bottleneck: implicit-quantile critics can represent the lower tail, but actor gradients derived from re-sampled tail queries can become noisy when catastrophic and non-catastrophic outcome modes induce well-separated action gradients. To address this, we introduce GBF-SAC, a Soft Actor-Critic instantiation of Generative Bellman Flow. Its conditional flow-matching critic models mean-policy continuation returns and generates a shared sample set for each state-action pair, from which GBF-SAC amortizes the mean and lower-tail statistics into differentiable Q-heads. These heads provide the mean and risk-sensitive actors with smoother scalar objectives rather than repeatedly re-sampled tail estimates. On a suite of risk-sensitive MuJoCo continuous-control tasks with a velocity-dependent crash wrapper, GBF-SAC improves the reward-crash trade-off on the main MuJoCo suite while maintaining competitive mean return. An IQN+Q-Head ablation further isolates the mechanism, indicating that the reward-safety gains arise from the interaction between stable tail representations and the Q-head bridge. Because the critic retains a full learned return distribution, the same trained model also supports diagnostic post-hoc risk scoring at deployment without retraining.

1 INTRODUCTION

Consider a high-speed locomotion controller that earns large reward most of the time but occasionally terminates in a crash. A mean-value critic averages crashes into the expected return; a distributional critic exposes them through the lower tail. The open question is whether that information can be turned into a reliable improvement signal for the policy.

Categorical, quantile, and implicit-quantile critics make return modeling increasingly expressive (Bellemare et al., 2017; Dabney et al., 2018b;a; Yang et al., 2019). An actor still has to turn that distribution into an optimization signal. With finitely many resampled lower-tail queries, successive updates depend on which return modes are sampled. We show that their additional variance scales with squared mode-gradient separation and decreases with the query budget. This identifies an actor-interface problem even when the return critic is accurate.

Generative Bellman Flow (GBF) uses generated return samples to learn persistent actor objectives. A conditional flow-matching critic (Lipman et al., 2023) supplies a common sample set for mean and lower-tail statistics, which supervise differentiable Q-heads. The mean actor defines the reference continuation, and the risk actor uses the tail head to improve action choices under this return model. Actor updates then reuse the learned objective surface. The same distribution remains available for diagnostic risk queries.

Controlled experiments with neural critics support this mechanism: tail-query variance follows the predicted scaling with mode-gradient separation and query budget. In the five-seed HalfCheetah control, adding a Q-head to IQN raises risk-policy reward from -139 to 1834 at the same 0% observed crash rate; GBF-SAC reaches 2611.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

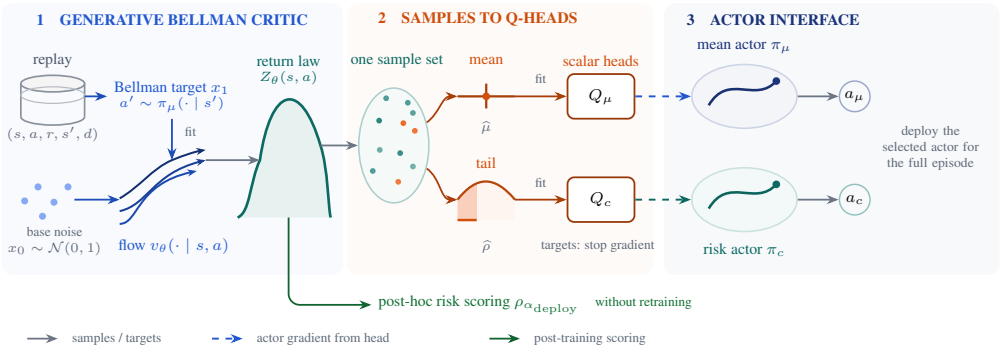


Figure 1: GBF-SAC actor interface. The flow critic models mean-policy continuation Z^{π_μ}. One generated sample set supplies stopped-gradient mean and lower-tail targets for the Q-heads. The actors differentiate through the heads; the return distribution remains available for diagnostic scoring.

In the four-task MuJoCo benchmark, GBF-SAC's mean actor achieves the highest average return on three tasks, and its risk actor records zero observed crashes on three tasks. In the fixed-α study, deploying the risk actor instead of its paired mean actor reduces crashes in all five seeds on HalfCheetah and Ant under both execution modes. On Ant, this change also improves trajectory CVaR in every seed.

2 Related Work

Distributional RL models random returns using categorical (Bellemare et al., 2017), quantile (Dabney et al., 2018b; Yang et al., 2019), and implicit representations (Dabney et al., 2018a; Yue et al., 2020). QR uses fixed quantile locations, whereas IQN samples quantile fractions and can query selected return regions. D4PG and DSAC incorporate distributional critics into continuous-control actor–critic learning (Barth-Maron et al., 2018; Ma et al., 2025); GMAC fits Gaussian-mixture return distributions using a sample-based Cramér discrepancy (Nam et al., 2021). TQC truncates upper quantiles to control overestimation (Kuznetsov et al., 2020). These methods make different choices about the statistics extracted from a return model. Rowland et al. (2019) formalize the relationship between learned statistics and distribution samples. GBF uses generated samples as regression targets for persistent mean and tail heads. Its actor differentiates the fitted statistic, removing fresh tail-query noise from that update.

CVaR is a coherent risk measure (Artzner et al., 1999; Rockafellar & Uryasev, 2000) with sampling-based policy-gradient estimators (Tamar et al., 2015). Risk-sensitive RL studies CVaR through robust control (Chow et al., 2015), optimistic exploration (Keramati et al., 2020), and offline policy learning (Urpí et al., 2021). GBF learns a lower-tail action objective under the mean-policy continuation and evaluates its risk actor over complete episodes. Safety-constrained methods instead specify an explicit cost criterion: CPO constrains expected cost (Achiam et al., 2017), while WCSAC uses a cost distribution to impose a tail-risk constraint (Yang et al., 2021). This makes WCSAC a relevant reward–safety comparison for GBF's return-based risk objective.

Generative return models extend distributional critics beyond fixed categorical or quantile outputs. Diffusion models generate samples by iterative denoising (Ho et al., 2020; Song et al., 2021); neural ODEs and flow matching parameterize continuous transports (Chen et al., 2018; Lipman et al., 2023; Tong et al., 2024). Diffusion-QL and IDQL use diffusion policies for offline RL (Wang et al., 2023; Hansen-Estruch et al., 2023), while Diffuser generates trajectories for planning (Janner et al., 2022). Recent flow-based value-learning methods include floq (Agrawalla et al., 2026) and Value Flows (Dong et al., 2026). FlowCritic models value distributions with flow matching and generates samples for value estimation (Zhong et al., 2025). GBF-SAC uses flow matching for online lower-tail actor optimization. A common return batch trains its mean and risk heads, which supply scalar objectives for subsequent actor updates. The IQN+Q-head control examines the benefit of this interface within a quantile critic, alongside the comparison between return representations.

3 PRELIMINARIES

For a Markov decision process with discount γ, distributional policy evaluation keeps the random return

$$Z^\pi(s,a) \overset{D}{=} R(s,a) + \gamma Z^\pi(S', A'), \quad S' \sim P(\cdot \mid s,a), \quad A' \sim \pi(\cdot \mid S'). \tag{1}$$

Larger returns are preferred. Lower-tail CVaR is $\mathrm{CVaR}_\alpha(Z) = \alpha^{-1} \int_0^\alpha F_Z^{-1}(u)\, du$: α specifies a tail probability mass. Conditional flow matching fits a vector field transporting Gaussian noise x_0 to return samples x_1. Along $x_t = (1-t)x_0 + tx_1$, its target velocity is $x_1 - x_0$ (Lipman et al., 2023).

4 METHOD: GENERATIVE BELLMAN FLOW

4.1 MEAN AND RISK OBJECTIVES

GBF-SAC builds on Soft Actor-Critic (Haarnoja et al., 2018) and uses one return model to compare mean and risk-sensitive action choices. The mean actor π_μ defines the entropy-regularized reference law

$$Z^{\pi_\mu}(s,a) \overset{D}{=} R_0 + \sum_{t \geq 1} \gamma^t [R_t - \lambda_\mu \log \pi_\mu(A_t \mid S_t)], \quad A_{t \geq 1} \sim \pi_\mu. \tag{2}$$

For a lower-tail functional ρ, the risk actor optimizes

$$J^\rho_{\mathrm{ref}}(\phi) = \mathbb{E}_{s \sim \mathcal{D},\, a \sim \pi_{c,\phi}} [\rho(Z^{\pi_\mu}(s,a)) - \lambda_c \log \pi_{c,\phi}(a \mid s)]. \tag{3}$$

During each actor update, the reference law is fixed. The mean actor optimizes its expected return, while the risk actor uses the lower tail of the same model. Separate networks allow the two roles to use independent objectives and entropy temperatures; a conditioned actor could parameterize both roles with the same mean-conditioned continuation.

The mean actor supplies the continuation actions in critic backups. The risk actor learns from the tail head and selects actions at every step when deployed for a complete episode (Section 6.3).

4.2 FLOW CRITIC AND SHARED-SAMPLE HEADS

GBF uses twin conditional flows $v_{\theta_i}(x_t, t \mid s, a)$ and target networks. For a replay transition (s, a, r, s', d), draw $a' \sim \pi_\mu(\cdot \mid s')$ and return samples from both target critics. Select the distribution with the smaller sample mean and form

$$\begin{aligned} x_1 &= r + \gamma(1-d)[x' - \lambda_\mu \log \pi_\mu(a' \mid s')], \\ \mathcal{L}_{\mathrm{critic}} &= \mathbb{E}\|v_\theta(x_t, t \mid s, a) - (x_1 - x_0)\|^2. \end{aligned} \tag{4}$$

Euler integration produces N return samples per queried pair. One batch supervises both mean and risk heads: $Q_\mu^{(i)} \approx \mathbb{E}Z_{\theta_i}$ and $Q_c^{(i)} \approx \rho(Z_{\theta_i})$. Let $\widehat{\mu}_i$ and $\widehat{\rho}_i$ be the mean and selected tail statistic of critic i's sample batch. Their regression objective is

$$\mathcal{L}_{\mathrm{head}} = \sum_{i=1}^{2} \mathbb{E}_{\mathcal{D}} \left[(Q_\mu^{(i)} - \mathrm{sg}[\widehat{\mu}_i])^2 + (Q_c^{(i)} - \mathrm{sg}[\widehat{\rho}_i])^2 \right], \tag{5}$$

where sg stops gradients through the generated targets. Each head learns a state-action function that subsequent policy updates can query. The actors update by

$$\mathcal{L}_{\pi_\mu} = \mathbb{E}[\lambda_\mu \log \pi_\mu(a \mid s) - \min_i Q_\mu^{(i)}(s,a)], \tag{6}$$

$$\mathcal{L}_{\pi_c} = \mathbb{E}[\lambda_c \log \pi_c(a_c \mid s) - \min_i Q_c^{(i)}(s,a_c)]. \tag{7}$$

The two entropy temperatures are independently tuned. We fit the scalar heads to generated-return targets at replay state–action pairs and update the actors by backpropagating through these heads at policy-generated actions.

4.3 Tail Targets and Risk Parameters

The actor interface accepts a tail statistic as its regression target. The four-task benchmark uses

$$\widehat{\rho}_\kappa(z_{1:N}) = \sum_j \frac{\exp[-z_j/(\kappa\widehat{s})]}{\sum_\ell \exp[-z_\ell/(\kappa\widehat{s})]} z_j, \qquad \widehat{s} = \max\{\mathrm{Std}(z_{1:N}), 1\}, \quad \kappa = 0.5. \tag{8}$$

Here $\kappa = 0.5$ controls tail emphasis relative to the sample spread. Above the numerical floor, normalization makes the relative weights invariant to positive reward rescaling.

The fixed-α control, denoted GBF-α, centers its weights at an empirical quantile. For ordered samples, let $k = \lceil \alpha N \rceil$ and $q_\alpha = z_{(k)}$. Using sigmoid weights $w_j = \sigma((q_\alpha - z_j)/(\kappa_\alpha \widehat{s}))$, we form the target

$$\widehat{\rho}_{\alpha,\kappa_\alpha}(z_{1:N}) = \frac{\sum_j w_j z_j}{\sum_j w_j}. \tag{9}$$

The experiments use $\alpha = 0.1$ and $\kappa_\alpha = 0.05$. The former selects the empirical threshold; the latter sets its standardized transition width. The same generated samples supervise the mean head and this tail head, so changing the statistic does not introduce a separate actor-side sampling operation. Appendix A.2 gives the numerical target diagnostics.

For canonical CVaR, α expresses the fraction of low-return outcomes relevant to the application. It can be specified by that preference or selected on validation data using a declared reward/risk criterion. The shared sample set also supports bottom-k averaging and other tail summaries without changing the head interface. The weighted targets provide scalar training scores; their discrepancy from canonical CVaR enters the approximation analysis. Canonical CVaR queries at other levels leave the trained actor fixed.

Algorithm 1 GBF-SAC update

1: Sample replay transitions (s, a, r, s', d).
2: Sample $a' \sim \pi_\mu(\cdot \mid s')$ and twin target-flow returns.
3: Select the lower-mean target distribution; form x_1 and update flows by Eq. (4).
4: Every two updates, generate one return batch per critic.
5: Regress mean/risk heads onto that batch's mean and selected tail statistic.
6: Update π_μ, π_c by Eqs. (6)–(7); tune their entropy temperatures.
7: Polyak-update target critics.

5 Theoretical Analysis

An actor interface must provide both a stable update and an accurate objective. We quantify the fresh-query component of gradient variance, then bound the error introduced by learning a persistent head. The analysis fixes the reference policy and actor-update state distribution. Distributional foundations are given in Appendix A.3.

5.1 The CVaR Gradient Bottleneck

Reparameterize an actor action as $a = g_\phi(s, \xi)$. With an exact quantile critic, write $G(\tau; s, \xi) = \nabla_\phi F^{-1}(\tau, s, g_\phi(s, \xi))$. The following calculation isolates query randomness conditional on the state and actor noise.

Theorem 1 (Conditional tail-query variance)**.** *Assume differentiation and integration can be interchanged and G has finite second moment. For K independent queries $\tau_k \sim U[0, \alpha]$,*

$$\widehat{g}_K = \frac{1}{K}\sum_k G(\tau_k; s, \xi), \qquad \mathrm{Var}[\widehat{g}_K \mid s, \xi] = \frac{1}{K}\mathrm{Var}_\tau[G(\tau; s, \xi)]. \tag{10}$$

For a two-point return law with locally fixed crash mass $0 < p_- < \alpha$, differentiable mode values $v_- < v_+$, and $\Delta_g = \|\nabla_\phi v_+ - \nabla_\phi v_-\|$,

$$\mathrm{Var}[\widehat{g}_K \mid s, \xi] = \frac{p_-}{\alpha}\left(1 - \frac{p_-}{\alpha}\right)\frac{\Delta_g^2}{K}. \tag{11}$$

Vector variance denotes the expected squared centered norm. Fixed-grid quadrature and a frozen Q-head both have zero fresh-query variance. GBF implements the latter through scalar-head regression. Section 6.1 measures the predicted scaling with learned critics.

For a frozen critic or head, the law of total variance separates the two sources:

$$\mathrm{Var}[\widehat{g}] = \mathbb{E}_{s,\xi}\,\mathrm{Var}[\widehat{g} \mid s, \xi] + \mathrm{Var}_{s,\xi}(\mathbb{E}[\widehat{g} \mid s, \xi])\,. \tag{12}$$

Theorem 1 quantifies the first term for resampled tail queries; the second is variation across actor states and action noise. Amortization lets the actor reuse a scalar surface at a fixed state and action, with generated samples supplying supervision for that surface. The next result relates its fidelity to the reference action objective.

For a minibatch $\mathcal{B}$ of B examples with independent query sets, define $\bar{\sigma}_F^2 = \mathbb{E}_{s,\xi}\,\mathrm{Var}_\tau[G(\tau; s, \xi)]$. The query component of the averaged update is

$$\mathbb{E}_{\mathcal{B},\xi}\,\mathrm{Var}[\widehat{g}_{\mathcal{B}} \mid \mathcal{B}, \xi] = \frac{\bar{\sigma}_F^2}{BK}. \tag{13}$$

Gradient variance enters standard nonconvex stochastic-optimization bounds (Ghadimi & Lan, 2013); increasing K reduces the query component by $1/K$. A persistent head supplies its gradient directly, with its accuracy governed by the following decomposition.

5.2 From Head Fidelity to Action Improvement

For $q_\alpha(s, a) = \mathrm{CVaR}_\alpha(Z^{\pi_\mu}(s, a))$, we bound the objective gap from optimizing its learned head.

Theorem 2 (Reference-objective approximation)**.** *Fix π_μ, the state distribution $\mathcal{D}$, and λ_c. Let Π be a policy class and Ω contain the support of $(s, a) \sim \mathcal{D}\pi$ for every $\pi \in \Pi$. Suppose regression, sampling, and target errors are uniformly bounded on Ω by $\varepsilon_{\mathrm{head}}, \varepsilon_{\mathrm{MC}}, \varepsilon_{\mathrm{target}}$, and define*

$$\varepsilon_Z = \sup_{(s,a)\in\Omega} W_1(Z_\theta(s, a), Z^{\pi_\mu}(s, a)) < \infty.$$

Then

$$\|\widehat{Q}_c - q_\alpha\|_{\infty,\Omega} \le \varepsilon_\alpha := \varepsilon_{\mathrm{head}} + \varepsilon_{\mathrm{MC}} + \varepsilon_{\mathrm{target}} + \varepsilon_Z/\alpha. \tag{14}$$

Let $\pi^* \in \Pi$ maximize $J_{\mathrm{ref}}^{CVaR_\alpha}$ and let $\widehat{\pi} \in \Pi$ maximize its head approximation up to residual δ_{opt}. Then

$$J_{\mathrm{ref}}^{CVaR_\alpha}(\pi^*) - J_{\mathrm{ref}}^{CVaR_\alpha}(\widehat{\pi}) \le 2\varepsilon_\alpha + \delta_{\mathrm{opt}}. \tag{15}$$

Equation (15) converts head error into an actor-objective gap of at most $2\varepsilon_\alpha + \delta_{\mathrm{opt}}$. Equation (14) assigns this error to regression, sampling, target choice, and return modeling; its $1/\alpha$ factor quantifies tail sensitivity.

Appendix A.6 relates value accuracy to action-gradient accuracy under additional smoothness assumptions. Section 6 evaluates the resulting policies.

6 Experiments

We evaluate how return-sample amortization affects actor gradients and deployed policy risk on four MuJoCo locomotion tasks with a velocity-dependent crash wrapper (Todorov et al., 2012).

The full benchmark uses Eq. (8), three common seeds, and 30 final episodes per seed; GBF/IQN/WCSAC also have five-seed extensions. The fixed-α study uses five separately trained policies per method and 1,000 common episodes per seed and mode.

6.1 Understanding the Actor Interface

The IQN+Q-head control changes risk-policy reward from -139 ± 133 to 1834 ± 540, compared with GBF's 2611 ± 139, at 0% observed crash (Appendix Table 10). The result isolates a persistent scalar interface while GBF additionally changes the return representation and tail target.

To test the query-noise mechanism, we train neural critics on a two-mode return law using three seeds, four mode separations, and query budgets $K \in \{8, 16, 32, 64\}$. The measured IQN gradient variance

closely follows the predicted Δ_g^2/K scaling (Figure 2). At $K = 64$, the fitted slope is 0.003688, against 0.003906 theoretically, with $R^2 = 0.9975$. The maximum fresh-query gradient variance of the frozen heads is 3.20×10^{-14}. Both IQN and flow heads remove that conditional component; their approximation errors remain distinct (Appendix Table 13).

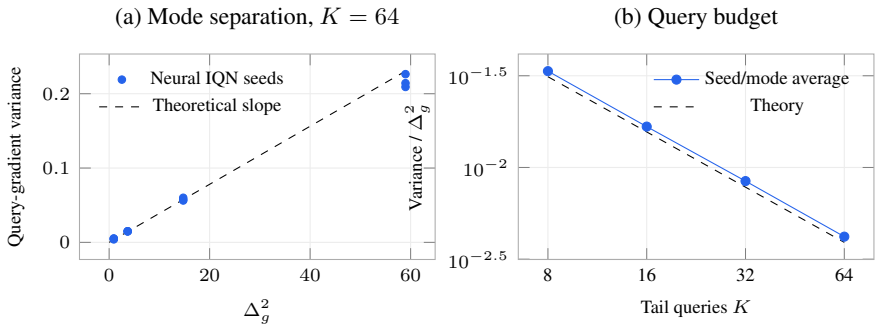


Figure 2: Learned-neural evidence for the query-gradient mechanism. Left: individual training-seed measurements at $K = 64$ and the theoretical slope. Right: variance normalized by squared mode separation, averaged across seeds and separations. Dashed curves are the theoretical predictions.

Full 1M-step target controls separate the head interface from its regression statistic. The shared sample set accepts bottom-k and benchmark-score targets without changing the critic or actor interface; complete paired outcomes and budgets are reported in Appendix F.3.

6.2 Control Performance

Table 1: Reward–crash operating points across four tasks: three common training seeds, 1M steps, deterministic actors, and 30 final episodes per seed. Returns are mean ± population SD. The GBF risk head uses Eq. (8); the panels report mean-actor return and risk-policy reward / observed crash rate.

Agent	HalfCheetah	Hopper	Walker2d	Ant
Mean actor reward higher is better				
GBF-SAC	**3138 ± 350**	**2573 ± 93**	2276 ± 503	**1938 ± 436**
IQN-SAC	2014 ± 314	2167 ± 316	2220 ± 313	1060 ± 124
DSAC	2332 ± 749	1554 ± 490	2180 ± 132	873 ± 56
QR-SAC	1723 ± 355	2006 ± 163	1320 ± 221	599 ± 242
Ens-SAC-CVaR	2363 ± 614	2553 ± 47	**2591 ± 349**	944 ± 331
WCSAC	1708 ± 47	610 ± 255	1325 ± 857	1356 ± 403
SAC	2930 ± 162	1611 ± 683	1793 ± 792	1397 ± 394
Risk-policy reward / observed crash rate higher reward and lower crash are better				
GBF-SAC	2708 ± 91 / **0%**	1449 ± 748 / 7%	1083 ± 1037 / **0%**	882 ± 347 / **0%**
IQN-SAC	−99 ± 129 / 0%	322 ± 212 / 3%	638 ± 371 / 1%	84 ± 158 / 0%
DSAC	990 ± 911 / 32%	436 ± 235 / 0%	395 ± 455 / 2%	−114 ± 146 / 0%
QR-SAC	2487 ± 420 / 63%	265 ± 121 / 0%	1769 ± 204 / 76%	−832 ± 472 / 0%
Ens-SAC-CVaR	2738 ± 13 / 69%	2543 ± 93 / 20%	2451 ± 571 / 17%	1330 ± 67 / 34%
WCSAC	1710 ± 46 / **0%**	611 ± 254 / **0%**	1425 ± 888 / 12%	1364 ± 400 / **0%**
SAC	2833 ± 47 / 60%	1572 ± 781 / 20%	1878 ± 753 / 32%	1200 ± 136 / 37%

Table 1 reports the reward–crash operating points. GBF's mean actor has the highest average return on three tasks, and its risk actor reaches 0% observed crash on three tasks. Figure 3 presents the joint operating points.

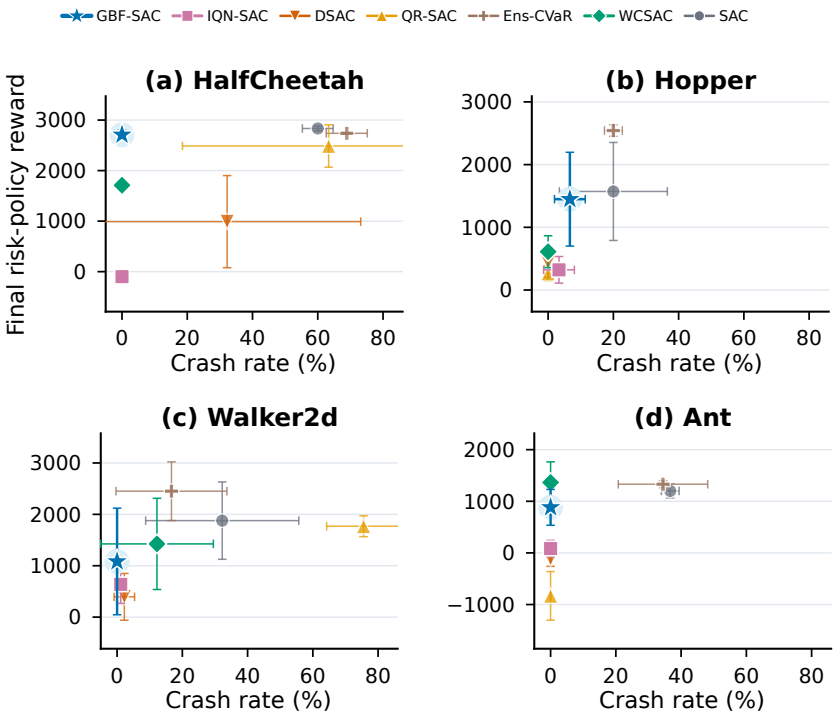


Figure 3: Risk-policy reward–crash operating points for the common three-seed benchmark. Points are seed means; error bars are seed standard deviations. The upper-left region combines higher reward with fewer observed crashes.

The five-seed analysis uses training runs as independent units. Table 2 summarizes reward effects supported by paired t tests after Holm adjustment across eight comparisons: HalfCheetah versus IQN, $+2750$ $[2593, 2908]$, HalfCheetah versus WCSAC, $+916$ $[726, 1106]$, and Ant versus IQN, $+818$ $[363, 1272]$. Each is favorable in all five seeds. Appendix Table 12 gives the complete comparison family. Appendix Figure 6 gives the four-task reward and crash learning curves.

Table 2: Five-seed risk-policy reward effects passing Holm-adjusted paired t tests at 0.05 across eight comparisons. Intervals are unadjusted paired seed 95% intervals; wins count favorable seeds. All eight comparisons appear in Appendix Table 12.

Task	Comparator	GBF reward effect [95% CI]	Wins
HalfCheetah	IQN	$+2750$ $[2593, 2908]$	5/5
HalfCheetah	WCSAC	$+916$ $[726, 1106]$	5/5
Ant	IQN	$+818$ $[363, 1272]$	5/5

6.3 Risk-Sensitive Deployment

The two actors permit a direct measurement of what risk-sensitive action selection contributes within each trained GBF-α agent. We execute both actors for 1,000 common episodes per seed. Across HalfCheetah and Ant, the risk actor reduces observed crashes in every training seed under both deterministic and stochastic execution (Figure 4).

On HalfCheetah, the crash reductions are 35.9 $[28.2, 43.7]$ percentage points deterministically and 38.4 $[25.2, 51.6]$ stochastically. On Ant they are 41.4 $[18.9, 63.9]$ and 42.9 $[28.1, 57.8]$ points. Deterministic HalfCheetah has a paired reward change of $+6.6$ $[-403, 416]$. Figure 4 displays all reward changes alongside the crash reductions; Table 3 gives the complete values.

Table 3: GBF-α risk actor relative to its paired mean actor: five training seeds and 1,000 common episodes per seed and mode. Intervals use training seeds; crash reduction is in percentage points.

Task	Mode	Reward change [95% CI]	Crash reduction [95% CI]	Crash wins
HalfCheetah	Det.	$+6.6\ [-403.0, 416.2]$	$35.9\ [28.2, 43.7]$	5/5
HalfCheetah	Stoch.	$-398.9\ [-997.3, 199.4]$	$38.4\ [25.2, 51.6]$	5/5
Ant	Det.	$-137.2\ [-1028.6, 754.2]$	$41.4\ [18.9, 63.9]$	5/5
Ant	Stoch.	$-119.7\ [-949.1, 709.6]$	$42.9\ [28.1, 57.8]$	5/5

Each pair uses two actors from the same trained checkpoint and the same evaluation scenarios. The comparison measures the deployment choice within one learned agent. All five seeds agree on the crash-reduction direction across both tasks and execution modes. The reward intervals and trajectory CVaR quantify the accompanying change in task performance.

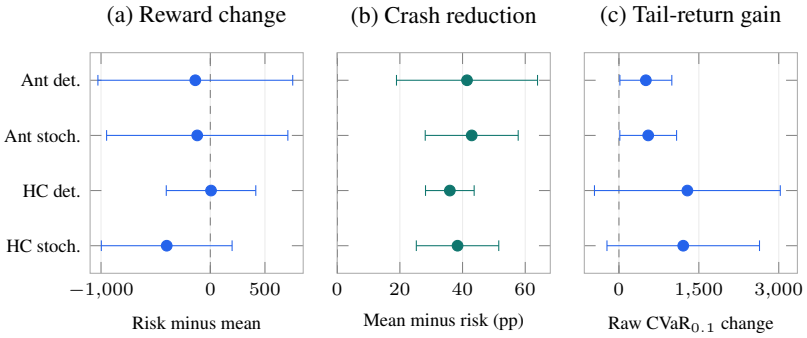


Figure 4: Deploying the GBF-α risk actor instead of its paired mean actor. All panels use the same five training seeds and 1,000 common complete episodes per seed and mode. Points and intervals are paired seed means and 95% intervals. Reward and CVaR are risk minus mean; crash reduction is mean minus risk in percentage points. HC denotes HalfCheetah.

Complete-trajectory $\text{CVaR}_{0.1}$ averages the lowest 100 raw returns. On Ant, the risk actor improves this quantity in all five seeds in both modes, with effects $+505\ [18, 993]$ and $+550\ [17, 1082]$. HalfCheetah has positive effects in four seeds, with means $+1285$ and $+1207$ and wider intervals (Table 4). These episode outcomes measure deployment directly, complementing the reference-continuation objective used during training.

Table 4: Complete-trajectory raw-return $\text{CVaR}_{0.1}$ of GBF-α: 1,000 common episodes per seed, five seeds. Effects are risk minus mean actor, with paired seed 95% intervals.

Task	Mode	Risk	Mean	Paired effect [95% CI]
Ant	Det.	270	-235	$+505\ [18, 993]$
Ant	Stoch.	329	-220	$+550\ [17, 1082]$
HalfCheetah	Det.	1282	-2	$+1285\ [-459, 3029]$
HalfCheetah	Stoch.	1244	37	$+1207\ [-225, 2639]$

The restored-state experiment isolates the effect of the subsequent policy. Holding the first action fixed, recursive risk continuation increases entropy-regularized discounted CVaR by $155.7\ [91.8, 219.6]$ on Ant and $115.9\ [24.3, 207.6]$ on HalfCheetah, with positive shifts in all ten trained policies.

6.4 Comparison with Dense Quantile Critics

Table 5: Fixed-α comparison: five seeds, 1M steps and 1,000 episodes per seed. Entries are reward (mean $\pm$ sample SD) / crash rate. GBF-α uses the quantile-centered target; these are separate trained policies from Table 1.

Task	Method	Deterministic	Stochastic
HalfCheetah	GBF-α	3034 ± 260 / 16.4%	2671 ± 539 / 8.7%
HalfCheetah	QR256	2547 ± 956 / 16.4%	2338 ± 1024 / 7.6%
HalfCheetah	QR512	5 ± 600 / 0.0%	-42 ± 554 / 0.0%
Ant	GBF-α	1564 ± 624 / 22.0%	1563 ± 536 / 17.7%
Ant	QR256	8 ± 798 / 0.0%	-6 ± 659 / 0.0%

Dense fixed-grid critics provide a performance comparison to the sampled interface. At $\alpha = 0.1$, GBF-α reaches reward 3034 on HalfCheetah at 16.44% observed crashes, compared with QR256's 2547 at 16.42%. The paired reward effect is $+487$ $[-778, 1752]$; under stochastic execution it is $+334$ $[-1106, 1773]$. Thus the point estimates place GBF-α at the higher-reward point for similar average crash levels, while the paired intervals quantify seed-to-seed variation.

On Ant, GBF-α retains higher reward in every seed, while QR256 occupies a zero-crash, lower-reward operating point. The QR512 HalfCheetah and QR256 Ant rows make these distinct operating points explicit (Table 5).

Across all four task/mode settings, GBF-α achieves higher seed-averaged trajectory $\text{CVaR}_{0.1}$ than QR256 (Appendix Table 16). For comparison within QR256, Table 15 reports the reward and crash changes from deploying its risk actor instead of its mean actor.

QR256/512 use midpoint grids and 64 target quantiles. Their fixed-level controls match GBF-α's risk-actor entropy target, exploration ratio, and clipping threshold. Appendix F.4 specifies the settings. The four-task benchmark and fixed-α study use the target choices defined in Section 4.3.

Computational cost. At batch size 256 and 32 return samples, a CPU critic-only measurement takes 14.9 ms with four-step flow sampling and 95.3 ms with 20-step DDIM sampling, a $6.4\times$ difference at these solver configurations. GBF uses one generated batch to supervise both actor objectives and refreshes the heads every two updates. Actor gradients subsequently traverse the scalar heads. The NFE and sample-count studies in Appendix E report the accompanying control-performance trade-offs.

Configuration summary. The benchmark uses 32 generated returns in training, 30 at evaluation, and four Euler flow steps. Heads refresh every two updates from shared mean/tail samples. Fixed-α controls use $\alpha = 0.1$, $\kappa_\alpha = 0.05$, five policies, and 1,000 paired episodes per seed and mode.

7 Conclusion

GBF-SAC fits mean and tail Q-heads to generated returns, allowing actor updates to reuse these estimates without fresh tail queries. Adding a Q-head to IQN raises HalfCheetah risk-policy reward from -139 to 1834 without changing the critic's quantile representation. In the fixed-α study, deploying GBF's risk actor reduces crashes in every seed on both tasks and improves trajectory CVaR on Ant. Amortizing generated returns into persistent value heads removes fresh tail-query noise from actor updates and gives risk-sensitive policies a reusable scalar objective.

AI USE STATEMENT

We used generative AI tools for literature search and manuscript language editing. We reviewed all AI-assisted text and take responsibility for the final manuscript, including its claims and conclusions.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/achiam17a.html.

Bhavya Kumar Agrawalla, Michal Nauman, Khush Agrawal, and Aviral Kumar. floq: Training critics via flow-matching for scaling compute in value-based RL. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=m14YNdmPAh.

Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999. doi: https://doi.org/10.1111/1467-9965.00068. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-9965.00068.

Gabriel Barth-Maron, Matthew W. Hoffman, David Budden, Will Dabney, Dan Horgan, Dhruva TB, Alistair Muldal, Nicolas Heess, and Timothy Lillicrap. Distributional policy gradients. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SyZipzbCb.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 449–458. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/bellemare17a.html.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html.

Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. Risk-sensitive and robust decision-making: a cvar optimization approach. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/64223ccf70bbb65a3a4aceac37e21016-Paper.pdf.

Will Dabney, Georg Ostrovski, David Silver, and Remi Munos. Implicit quantile networks for distributional reinforcement learning. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1096–1105. PMLR, 10–15 Jul 2018a. URL https://proceedings.mlr.press/v80/dabney18a.html.

Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018b. doi: 10.1609/aaai.v32i1.11791. URL https://ojs.aaai.org/index.php/AAAI/article/view/11791.

Perry Dong, Chongyi Zheng, Chelsea Finn, Dorsa Sadigh, and Benjamin Eysenbach. Value flows. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=2VyNYUVF2k.

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013. doi: 10.1137/120880811. URL https://doi.org/10.1137/120880811.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. IDQL: Implicit Q-learning as an actor-critic method with diffusion policies, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9902–9915. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/janner22a.html.

Ramtin Keramati, Christoph Dann, Alex Tamkin, and Emma Brunskill. Being optimistic to be conservative: Quickly learning a cvar policy. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):4436–4443, Apr. 2020. doi: 10.1609/aaai.v34i04.5870. URL https://ojs.aaai.org/index.php/AAAI/article/view/5870.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5556–5566. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/kuznetsov20a.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xiaoteng Ma, Junyao Chen, Li Xia, Jun Yang, Qianchuan Zhao, and Zhengyuan Zhou. DSAC: Distributional soft actor-critic for risk-sensitive reinforcement learning. *Journal of Artificial Intelligence Research*, 83, 2025. doi: 10.1613/jair.1.17526. URL https://www.jair.org/index.php/jair/article/view/17526.

Daniel W Nam, Younghoon Kim, and Chan Y Park. Gmac: A distributional perspective on actor-critic framework. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7927–7936. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/nam21a.html.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2(3):21–41, 2000. doi: 10.21314/JOR.2000.038.

Mark Rowland, Robert Dadashi, Saurabh Kumar, Remi Munos, Marc G. Bellemare, and Will Dabney. Statistics and samples in distributional reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5528–5536. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/rowland19a.html.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Aviv Tamar, Yonatan Glassner, and Shie Mannor. Optimizing the cvar via sampling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1):2993–2999, 2015. doi: 10.1609/aaai.v29i1.9561. URL https://ojs.aaai.org/index.php/AAAI/article/view/9561.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

Núria Armengol Urpí, Sebastian Curi, and Andreas Krause. Risk-averse offline reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=TBIzh9b5eaz.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=AHvFDPi-FA.

Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tie-Yan Liu. Fully parameterized quantile function for distributional reinforcement learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f471223d1a1614b58a7dc45c9d01df19-Paper.pdf.

Qisong Yang, Thiago D. Simão, Simon H. Tindemans, and Matthijs T. J. Spaan. Wcsac: Worst-case soft actor critic for safety-constrained reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):10639–10646, May 2021. doi: 10.1609/aaai.v35i12.17272. URL https://ojs.aaai.org/index.php/AAAI/article/view/17272.

Yuguang Yue, Zhendong Wang, and Mingyuan Zhou. Implicit distributional reinforcement learning. *Advances in Neural Information Processing Systems*, 33:7135–7147, 2020.

Shan Zhong, Shutong Ding, He Diao, Xiangyu Wang, Kah Chan Teh, and Bei Peng. FlowCritic: Bridging value estimation with flow matching in reinforcement learning, 2025.

A OBJECTIVES, TARGETS, AND PROOFS

A.1 RETURN AND EVALUATION DEFINITIONS

The critic evaluates the entropy-regularized discounted reference law in Eq. (2). Its Bellman continuation action is sampled from π_μ. The risk actor's minibatch state distribution is the replay distribution, held fixed within an actor update. At deployment, the selected actor is executed recursively for the whole episode.

Complete-episode performance uses the original task rewards, summed without entropy bonuses or discounting. Its empirical $\mathrm{CVaR}_{0.1}$ is the mean of the lowest 100 returns among 1,000 episodes. Restored-state continuation experiments instead compare entropy-regularized discounted returns under specified continuation policies. These definitions separate the trained score, a controlled change in continuation, and actual episode outcomes.

A.2 TAIL TARGETS AND THE MEANING OF THE RISK PARAMETERS

The benchmark target is Eq. (8). Its population counterpart is

$$\rho_\kappa(Z) = \frac{\mathbb{E}[Z \exp(-Z/(\kappa s_Z))]}{\mathbb{E}[\exp(-Z/(\kappa s_Z))]}, \qquad s_Z = \max\{\mathrm{Std}(Z), 1\}.$$

The parameter κ specifies relative tail emphasis; this target does not impose a prescribed tail mass.

GBF-α uses the quantile-centered target in Eq. (9), with $\alpha = 0.1$ and $\kappa_\alpha = 0.05$. The threshold is an empirical quantile, and the finite width produces a weighted lower-tail score. Its difference from population CVaR_α belongs to the target and sample terms of Eq. (14).

Table 6: Quantile-centered target diagnostics at $\alpha = 0.1$, $N = 32$, across the stored controlled-law family. Bias is absolute population deviation from canonical CVaR, normalized by the law's standard deviation; ESS is mean finite-sample effective sample size.

κ_α	Mean bias / SD	Max. bias / SD	Mean ESS
0.02	0.029	0.118	4.35
0.05	0.160	0.549	5.79
0.1	0.461	1.144	8.84
0.2	0.852	1.883	13.45
0.3	1.065	2.230	16.23
0.5	1.338	2.462	20.12

The numerical target study uses normal and Student-t_3 laws and crash-mixture settings with masses 0.02, 0.05, and 0.10. Table 6 reports the recorded bias and effective sample size as the width changes. At the evaluated width 0.05, the mean population deviation is 0.160 law standard deviations and the mean finite-sample ESS is 5.79. These diagnostics quantify the target component of the approximation trade-off in the controlled distributions; the uniform error assumption in Theorem 2 is stated separately.

The order-statistic control uses

$$\widehat{\rho}_k = \frac{1}{k} \sum_{j=1}^{k} z_{(j)}, \qquad k = \lceil \alpha N \rceil. \tag{16}$$

For $N = 32, \alpha = 0.1$, it averages four samples. This corresponds to empirical tail mass $4/32$; exact empirical CVaR at a fractional αN uses the fractional mass of the boundary sample. The population CVaR in the theory is defined by the quantile integral.

For canonical CVaR, the application supplies the lower-tail mass of interest. This parameter is not a constraint on a separately defined crash event. A learned reference-return distribution can be scored at additional tail levels without retraining that critic. A changed scoring level does not itself retrain the head or actor.

A.3 DISTRIBUTIONAL FOUNDATIONS

For fixed π, λ, define

$$(\mathcal{T}^{\pi,\lambda} Z)(s,a) \overset{D}{=} R(s,a) + \gamma[Z(S', A') - \lambda \log \pi(A' \mid S')]. \tag{17}$$

Proposition 1 (Reference-law contraction)**.** *On a class with finite uniform first moments and integrable entropy corrections,*

$$\sup_{s,a} W_1(\mathcal{T}^{\pi,\lambda} Z_1, \mathcal{T}^{\pi,\lambda} Z_2) \le \gamma \sup_{s,a} W_1(Z_1, Z_2). \tag{18}$$

Proof. Couple the transition, continuation action, and reward identically in both backups. Conditional on each successor pair, couple the two returns optimally. The common reward and entropy correction cancel, leaving an expected absolute difference bounded by $\gamma W_1(Z_1(S', A'), Z_2(S', A'))$. Averaging and taking the supremum gives the claim. □

Proposition 2 (Risk Lipschitz bounds)**.** $|\mathbb{E}Z_1 - \mathbb{E}Z_2| \le W_1(Z_1, Z_2)$ *and* $|\mathit{CVaR}_\alpha(Z_1) - \mathit{CVaR}_\alpha(Z_2)| \le W_1(Z_1, Z_2)/\alpha$.

Proof. The mean bound follows from an optimal coupling. For CVaR, use $\mathrm{CVaR}_\alpha(Z) = \sup_\nu\{\nu - \alpha^{-1}\mathbb{E}[(\nu - Z)_+]\}$. Each integrand is $1/\alpha$-Lipschitz in the return. Apply the coupling bound at fixed ν and then $|\sup f - \sup g| \le \sup |f - g|$. □

The term ε_Z measures the discrepancy between the generated return law and the reference law, accounting for model fitting and numerical integration.

A.4 PROOF OF THE TAIL-QUERY VARIANCE RESULT

Proof of Theorem 1. Condition on (s, ξ) and the critic. The interchange assumption gives

$$\mathbb{E}_{\tau \sim U[0,\alpha]} G(\tau; s, \xi) = \nabla_\phi \left[\frac{1}{\alpha} \int_0^\alpha F^{-1}(u, s, g_\phi(s, \xi))\, du \right].$$

Independence of the K queries eliminates centered cross terms, so the variance of their average is $1/K$ times the variance of one query.

For the two-point law, a tail query selects v_- with probability p_-/α and v_+ otherwise. With locally fixed mixture mass, the two gradient values are $\nabla_\phi v_-$ and $\nabla_\phi v_+$. The variance of this Bernoulli mixture is $\frac{p_-}{\alpha}(1 - \frac{p_-}{\alpha})\Delta_g^2$. Dividing by K proves Eq. (11). □

For Eq. (13), independent query sets give conditional variance $B^{-2} \sum_{i=1}^B \sigma_F^2(s_i, \xi_i)/K$ for the batch average. Taking expectations under the common example distribution gives $\bar{\sigma}_F^2/(BK)$.

Averaging K independent tail queries reduces the conditional variance by $1/K$. A fixed head supplies a deterministic gradient for the same state and actor-noise realization.

A.5 PROOF OF THE REFERENCE-OBJECTIVE BOUND

Proof of Theorem 2. Let $\widehat{\rho}_N$ be the finite target and $\rho(Z_\theta)$ its population target functional. Insert these quantities and canonical CVaR of the learned distribution:

$$\begin{aligned} |\widehat{Q}_c - \mathrm{CVaR}_\alpha(Z^{\pi_\mu})| \le\ & |\widehat{Q}_c - \widehat{\rho}_N| + |\widehat{\rho}_N - \rho(Z_\theta)| \\ & + |\rho(Z_\theta) - \mathrm{CVaR}_\alpha(Z_\theta)| + |\mathrm{CVaR}_\alpha(Z_\theta) - \mathrm{CVaR}_\alpha(Z^{\pi_\mu})|. \end{aligned} \tag{19}$$

The first three terms are head, sampling, and target errors; Proposition 2 bounds the last by ε_Z/α. Taking the supremum over Ω with the assumed component bounds proves Eq. (14). The same bound holds for the minimum of twin heads when both satisfy it around the common target.

Write $\widehat{J}$ for the head objective and $J = J_{\mathrm{ref}}^{\mathrm{CVaR}_\alpha}$, with π_μ, $\mathcal{D}$, and λ_c fixed as in the theorem. For every $\pi \in \Pi$, the support condition and cancellation of entropy terms give $|\widehat{J}(\pi) - J(\pi)| \le \varepsilon_\alpha$. Therefore

$$J(\pi^*) \le \widehat{J}(\pi^*) + \varepsilon_\alpha \le \widehat{J}(\widehat{\pi}) + \delta_{\mathrm{opt}} + \varepsilon_\alpha \le J(\widehat{\pi}) + \delta_{\mathrm{opt}} + 2\varepsilon_\alpha.$$

□

A.6 VALUE AND GRADIENT FIDELITY

Suppose $e(a) = \widehat{Q}_c(a) - q_\alpha(a)$ satisfies $|e| \le \varepsilon_\alpha$ and $\|\nabla_a^2 e\|_{\mathrm{op}} \le H$ on a neighborhood containing $a + hu$ for every unit u and $0 < h \le r$. Taylor's theorem gives

$$|u^\top \nabla_a e(a)| \le \frac{2\varepsilon_\alpha}{h} + \frac{Hh}{2}.$$

If $H > 0$ and $2\sqrt{\varepsilon_\alpha / H} \le r$, then $\|\nabla_a e(a)\| \le 2\sqrt{H\varepsilon_\alpha}$. Multiplying by the actor Jacobian bound controls the corresponding reparameterized gradient error. Thus removing fresh-query noise is accompanied by an explicit approximation requirement.

B HYPERPARAMETERS

Table 7: GBF-SAC hyperparameters for the main experiments (1M steps).

Parameter	Value	Notes
Batch size	256	
Learning rate	3×10^{-4}	Adam (Kingma & Ba, 2015)
Adam (β_1, β_2)	$(0.9, 0.999)$	PyTorch defaults
Discount γ	0.99	
Soft update τ	0.005	
Initial entropy λ	0.2	Auto-tuned
Flow N (train)	32	Samples per update
Target backup N_t	4	Samples per transition
Flow N (eval)	30	Samples for evaluation
ODE steps (NFE)	4	Selected from ablation
Tail-score scale κ	0.5	Eq. (8)
Q-head update freq	2	Amortized updates
Actor updates / env step	1	Both actors
Critic updates / env step	1	Both flow critics
Replay buffer	500K	
Warmup steps	5000	Random actions
Hidden dim	256	All MLPs
Cond dim	128	Flow conditioning

Table 8: Diffusion-SAC hyperparameters. The optimizer, discount, target update, buffer, warmup, and hidden dimensions follow the benchmark settings.

Parameter	Value	Notes
Diffusion steps T	100	DDPM forward process
DDIM sampling steps	20	Minimum for stable convergence
Noise schedule	Cosine	$\bar{\alpha}_t = \cos^2(\frac{t/T+s}{1+s} \cdot \frac{\pi}{2})$
Timestep embedding	64-dim	Sinusoidal + linear
Noise predictor	3-layer MLP	256 hidden, ReLU
Target backup N_t	4	Samples per transition
Diffusion N (eval)	30	Samples for evaluation

Table 9: IQN-SAC hyperparameters (matching GBF-SAC where possible for fair comparison).

Parameter	Value	Notes
Batch size	256	Same as GBF-SAC
Learning rate	3×10^{-4}	Adam
Discount γ	0.99	
Soft update τ	0.005	
Initial entropy λ	0.2	Auto-tuned
N_τ (tau samples)	64	Per forward pass
Cosine embedding dim	64	$\cos(\pi i \tau)$ basis
CVaR α	0.1	Bottom 10%
Quantile Huber κ	1.0	
Explore CVaR ratio	0.3	Same as GBF-SAC
CVaR entropy scale	0.5	Same as GBF-SAC
Gradient clip	1.0	Same as GBF-SAC
Replay buffer	500K	
Warmup steps	5000	Random actions
Hidden dim	256	All MLPs

C Environment Details

We report four MuJoCo locomotion tasks. The MuJoCo tasks use v4 (Todorov et al., 2012) with a shared risk-sensitive wrapper that introduces a velocity-dependent crash mechanism in a spatial danger zone.

HalfCheetah. State dim: 17. Action dim: 6. Danger zone: $x > 5.0$m. Velocity threshold: 4.0 m/s. Crash probability coefficient: 0.003. Crash penalty: -500. Max episode length: 1000 steps.

Hopper. State dim: 11. Action dim: 3. Danger zone: $x > 3.0$m. Velocity threshold: 2.0 m/s. Crash probability coefficient: 0.005. Crash penalty: -500.

Walker2d. State dim: 17. Action dim: 6. Danger zone: $x > 3.0$m. Velocity threshold: 2.5 m/s. Crash probability coefficient: 0.005. Crash penalty: -500.

Ant. State dim: 27. Action dim: 8. Danger zone: $x > 4.0$m. Velocity threshold: 2.0 m/s. Crash probability coefficient: 0.004. Crash penalty: -500.

D Additional Experimental Details

Bandit validation. 2-armed bandit: action 0 yields deterministic reward 5; action 1 yields bimodal reward $\{0, 10\}$ with equal probability. GBF correctly recovers both a Dirac delta and bimodal distribution.

Corridor validation. 3-state linear MDP with discounted bimodal terminal rewards. GBF preserves distributional structure across multi-step Bellman backups.

Seeds and reproducibility. The complete MuJoCo table reports the common seeds $\{42, 123, 456\}$ for all baselines. Random seeds control environment reset, action sampling, network initialization, and replay buffer sampling. Because HalfCheetah is the clearest collapse case and Hopper was the highest-variance case, we additionally ran seeds $\{789, 1024\}$ for GBF-SAC, IQN-SAC, IQN+Q-head, and WCSAC on both tasks; this targeted subset is reported in Table 10. We also completed targeted 5-seed comparisons on Walker2d and Ant for GBF-SAC, IQN-SAC, and WCSAC (Table 11).

Table 10: Targeted IQN+Q-head bridge comparisons on the five-seed mechanism subset. On HalfCheetah, the cleanest collapse case, the scalar bridge partially recovers reward but does not match GBF-SAC's reward-preserving behavior. On Hopper, all four agents are within seed noise of each other, so this row supports the high-variance reading rather than a strict ranking. WCSAC remains conservative but lower-reward on both tasks.

Setting	Agent	CVaR Reward	CVaR Crash
HalfCheetah	**GBF-SAC**	$\mathbf{2611 \pm 139}$	**0%**
HalfCheetah	IQN-SAC	-139 ± 133	0% (passive)
HalfCheetah	IQN-SAC+Q-head	1834 ± 540	0%
HalfCheetah	WCSAC	1695 ± 41	0%
Hopper	**GBF-SAC**	$\mathbf{1042 \pm 809}$	9%
Hopper	IQN-SAC	344 ± 170	2%
Hopper	IQN-SAC+Q-head	819 ± 995	**1%**
Hopper	WCSAC	601 ± 325	**0%**

E Additional Experiments and Analysis

This appendix contains supplementary experiments, ablations, and analysis referenced in the main text.

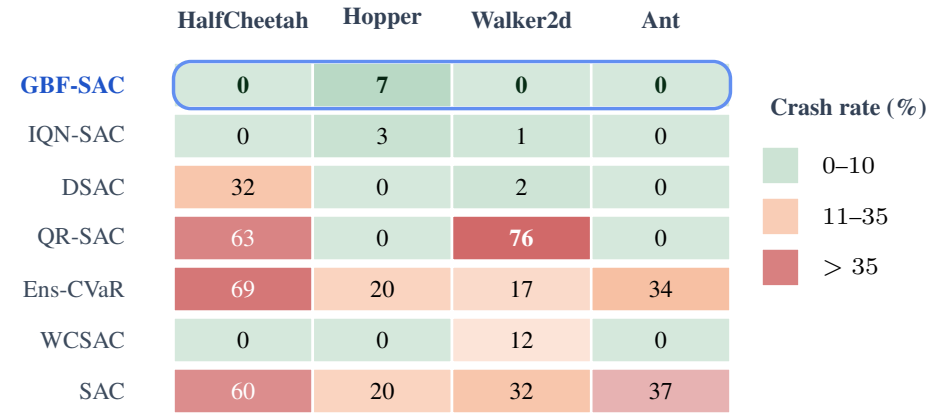


Figure 5: Risk-policy observed crash rates on the main MuJoCo suite. The heatmap makes the reward and safety pattern visually explicit: GBF-SAC lies in the band with low crash rates on all four tasks, while several baselines obtain low crashes only with much lower reward or crash heavily on selected environments.

Tail-target ablation. The 300K-step comparison uses the benchmark target in Eq. (8), bottom-k averaging, and EVT/GPD tail fitting on HalfCheetah and Hopper (three seeds). EVT achieves risk-policy reward 2719 with 1% crash on HalfCheetah; bottom-k achieves 1668 with 8% crash on Hopper; the benchmark target achieves 0% observed crash on HalfCheetah. Target preference is task-dependent. The full 1M-step comparison is reported in Appendix F.3.

NFE ablation. In the 50K-step ablation, NFE=4 provides the clearest risk differentiation: the mean actor pushes performance (5% crash) while the CVaR actor maintains 0% observed crash. At NFE=1–2, both actors remain conservative; at NFE=8, both become aggressive (15% crash).

Flow sample count. Increasing N from 16 to 64 keeps Walker2d CVaR observed crash near zero ($\sim$1% across 5 seeds) while Hopper CVaR observed crash remains in the same high-variance band ($\sim$13%), at modest mean performance cost.

Q-head bridge ablation. Removing the Q-head bridge and passing gradients directly through ODE samples changes risk-policy observed crashes from 0% to 1–3% in the recorded ablation.

Gradient diagnostic. A 30K-step HalfCheetah run (seed 42) recomputes the risk-actor loss every 50 updates and records consecutive-gradient cosine similarity. Over the final 100 logged updates, the cosine averages are 0.0126 for GBF-SAC and -0.0049 for IQN-SAC. The recorded actor losses are -15.7 and -12.0 for their respective objectives. The learned-neural experiment in Section 6.1 directly measures conditional query variance.

Table 11: Targeted 5-seed comparison on Walker2d and Ant for GBF-SAC, IQN-SAC, and WCSAC (1M steps). Cells report seed mean $\pm$ population SD / crash rate. Mean and Risk denote the actor used for evaluation; both report average episode return, not trajectory CVaR. The table combines the common seeds $\{42, 123, 456\}$ with additional seeds $\{789, 1024\}$ for this three-agent subset; Table 1 remains the complete comparison on common seeds across all baselines.

Agent	Walker2d Mean	Walker2d Risk	Ant Mean	Ant Risk
GBF-SAC	$\mathbf{2477 \pm 529}$ / 23%	1207 ± 927 / **1%**	$\mathbf{1789 \pm 389}$ / 61%	772 ± 306 / **0%**
IQN-SAC	1941 ± 571 / 59%	862 ± 419 / 2%	1171 ± 399 / 77%	-46 ± 236 / **0%**
WCSAC	1554 ± 744 / 47%	$\mathbf{1584 \pm 727}$ / 39%	1226 ± 351 / **0%**	$\mathbf{1235 \pm 348}$ / **0%**

F Extended Experimental Evidence

F.1 Training-seed inference

The independent unit is a complete training run. We pair final-checkpoint rewards by training seed across $\{42, 123, 456, 789, 1024\}$ and report mean reward differences, paired t confidence intervals, and favorable-seed counts. Holm adjustment is applied jointly to the eight task/comparator combinations.

The common-three-seed benchmark and original five-seed summary tables use population standard deviations. The fixed-level and external-task tables use sample standard deviations. Episode counts describe the measurement precision of each trained policy.

Table 12: Complete five-seed inference family for the benchmark risk policies. Intervals are paired seed t intervals; p_{Holm} is adjusted jointly over the eight comparisons.

Task	Comparator	GBF reward effect [95% CI]	p_{Holm}	Wins
HalfCheetah	IQN	$+2750$ $[2593, 2908]$	8.71×10^{-6}	5/5
HalfCheetah	WCSAC	$+916$ $[726, 1106]$	0.00127	5/5
Hopper	IQN	$+697$ $[-352, 1747]$	0.556	4/5
Hopper	WCSAC	$+441$ $[-932, 1813]$	0.846	4/5
Walker2d	IQN	$+345$ $[-737, 1427]$	0.846	3/5
Walker2d	WCSAC	-377 $[-848, 94]$	0.451	1/5
Ant	IQN	$+818$ $[363, 1272]$	0.045	5/5
Ant	WCSAC	-463 $[-1188, 261]$	0.556	1/5

F.2 LEARNED-NEURAL INTERFACE EXPERIMENT

Table 13: Learned two-mode interface diagnostics at $K = 64$, averaged over the three trained seeds and four mode separations. Gradient MAE is relative to the experiment's oracle.

Interface	Mean query variance	Mean gradient MAE
IQN direct	0.0735	0.183
IQN + head	1.81×10^{-14}	0.0118
Flow direct	1.94	1.27
Flow + head	1.22×10^{-14}	1.13

The experiment crosses three critic-training seeds, four mode separations $\{1, 2, 4, 8\}$, and query budgets $\{8, 16, 32, 64\}$. The plot uses the actor-gradient separation recorded for each mode setting. At each setting the critic and head are held fixed while query draws are repeated. The mean gradient error and conditional query variance are measured separately.

Head amortization reduces the measured fresh-query variance to numerical precision for both critic families. Table 13 reports gradient MAE separately to characterize the fitted objectives.

F.3 FULL-HORIZON TARGET CONTROLS

Table 14: Full 1M-step target controls on three paired seeds. Rewards are seed means. Crash counts follow seeds 42/123/456. HalfCheetah uses 30 evaluation episodes per seed; Hopper uses 1,000 per seed and mode.

Task	Mode	Bottom-k reward	Benchmark-target reward	Crash counts: bottom-k / benchmark
HalfCheetah	Det.	1818	2708	0/4/0 / 0/0/0
Hopper	Det.	2712	677	283/0/25 / 0/0/202
Hopper	Stoch.	2694	335	311/14/39 / 0/1/24

The paired reward effects (bottom-k minus benchmark target) are -891 $[-5019, 3238]$ for HalfCheetah, $+2035$ $[269, 3801]$ for deterministic Hopper, and $+2359$ $[1997, 2721]$ for stochastic Hopper. Hopper crash-effect intervals contain zero. The target controls are separately trained runs with matched within-seed runtimes; their means are not re-evaluations of Table 1. Hopper seeds 42/123 use matched CUDA training and seed 456 matched CPU training.

F.4 BASELINE AND FIXED-LEVEL COMPARISON DETAILS

The benchmark QR64 uses twin midpoint quantile critics, a quantile-Huber loss with threshold 1, mean-policy distributional Bellman targets, and the lowest six outputs for its risk actor. Ensemble-SAC-CVaR uses twin five-member scalar ensembles and their minimum. The benchmark uses batch size 256, Adam at 3×10^{-4}, discount 0.99, Polyak coefficient 0.005, a 500K replay buffer, 5K warmup steps, and width 256.

For the dense-grid study, QR256 and QR512 each regress every midpoint against 64 fixed target quantiles, using $64M$ pairwise losses for M outputs. The actor averages the lowest $\lfloor 0.1M \rfloor$ outputs. GBF-α uses Eq. (9). These finite estimators share the requested level, while their finite-sample target errors differ.

The dense-grid controls match GBF-α's risk-actor entropy target $-0.5\,\mathrm{action_dim}$, exploration ratio 0.3, and gradient-clipping threshold 1.0. The mean actor defines their continuation targets. Evaluation scenarios and runtimes are paired within each training seed. The five-seed reward/crash comparison in Table 5 reports both policy modes, including low-return dense-grid settings.

F.5 COMPLETE-TRAJECTORY DEPLOYMENT

Table 4 compares the two frozen actors trained together in each GBF-α run. Both are executed for entire episodes under common scenarios. This endpoint concerns the realized trajectories, whereas the learned critic models the reference law in Eq. (2).

Table 3 combines reward changes with crash reductions for the same policies. All five seeds favor the GBF-α risk actor in crash frequency on both tasks and under both execution modes. Reward effects have wider seed intervals. The paired quantities measure the performance change associated with deploying the risk actor within each trained agent.

Table 15: QR256 risk actor relative to its paired mean actor under the same five-seed evaluation. These are within-method effects; Table 5 compares the methods' deployed risk policies.

Task	Mode	Reward change [95% CI]	Crash reduction [95% CI]
HalfCheetah	Det.	$+245$ $[-1072, 1561]$	52.2 $[7.2, 97.2]$
HalfCheetah	Stoch.	-21 $[-1476, 1434]$	56.6 $[14.3, 98.9]$
Ant	Det.	-423 $[-879, 33]$	0.0 $[0.0, 0.0]$
Ant	Stoch.	-447 $[-921, 27]$	0.0 $[0.0, 0.0]$

The corresponding QR256 effects are reported in Table 15. On HalfCheetah, its risk actor also reduces crashes in every seed; both QR256 actors have zero observed crashes on Ant. These within-agent results describe each design's actor roles.

Table 16: Raw-return $\text{CVaR}_{0.1}$ computed from the saved final-policy episode returns. Each method has five seeds and 1,000 episodes per seed. Effects are paired GBF-α minus QR256, with 95% seed t intervals.

Task	Mode	GBF-α	QR256	Paired effect [95% CI]	Wins
HalfCheetah	Det.	1282	865	$+418$ $[-2465, 3300]$	2/5
HalfCheetah	Stoch.	1244	1142	$+102$ $[-2161, 2365]$	2/5
Ant	Det.	270	-163	$+433$ $[-608, 1475]$	3/5
Ant	Stoch.	329	-152	$+481$ $[-362, 1324]$	3/5

For Table 16, we average the lowest 100 of each saved set of 1,000 raw episode returns and pair methods by training seed. GBF-α has higher mean CVaR in all four task/mode comparisons, while each paired interval includes zero. The five-seed variation is retained in the intervals and direction counts. These summaries use the same saved policies and episodes as the reward/crash comparison.

F.6 SAFETYPOINTGOAL1 UNDER REWARD–COST SCALARIZATION

Table 17: SafetyPointGoal1-v0: three 300K-step training seeds, 200 common evaluation scenarios per policy. Entries are seed mean $\pm$ sample SD. All methods use the same fixed reward-minus-cost scalarization. Native task reward and cost are reported separately.

Method	Native return	Episode cost	Any-cost rate
GBF-SAC	-0.902 ± 0.760	28.69 ± 11.74	$9.7 \pm 2.4\%$
IQN	0.054 ± 0.113	53.06 ± 7.25	$12.0 \pm 1.8\%$
WCSAC	-0.092 ± 0.184	31.91 ± 11.09	$6.5 \pm 2.3\%$

We evaluate GBF-SAC, IQN, and WCSAC on SafetyPointGoal1-v0 using the same reward–cost scalarization, three training seeds, and a 300K-step budget. Table 17 reports native return, episode cost, and any-cost rate over 200 common evaluation scenarios per policy.

F.7 Learning dynamics and computational cost

The benchmark's NFE, sample-count, target, and head ablations are reported alongside the original task details. A CPU critic-only timing measurement at batch 256 and 32 samples gives 14.9 ms for four-step flow sampling and 95.3 ms for 20-step DDIM sampling. These numbers compare the specified sampling configurations.

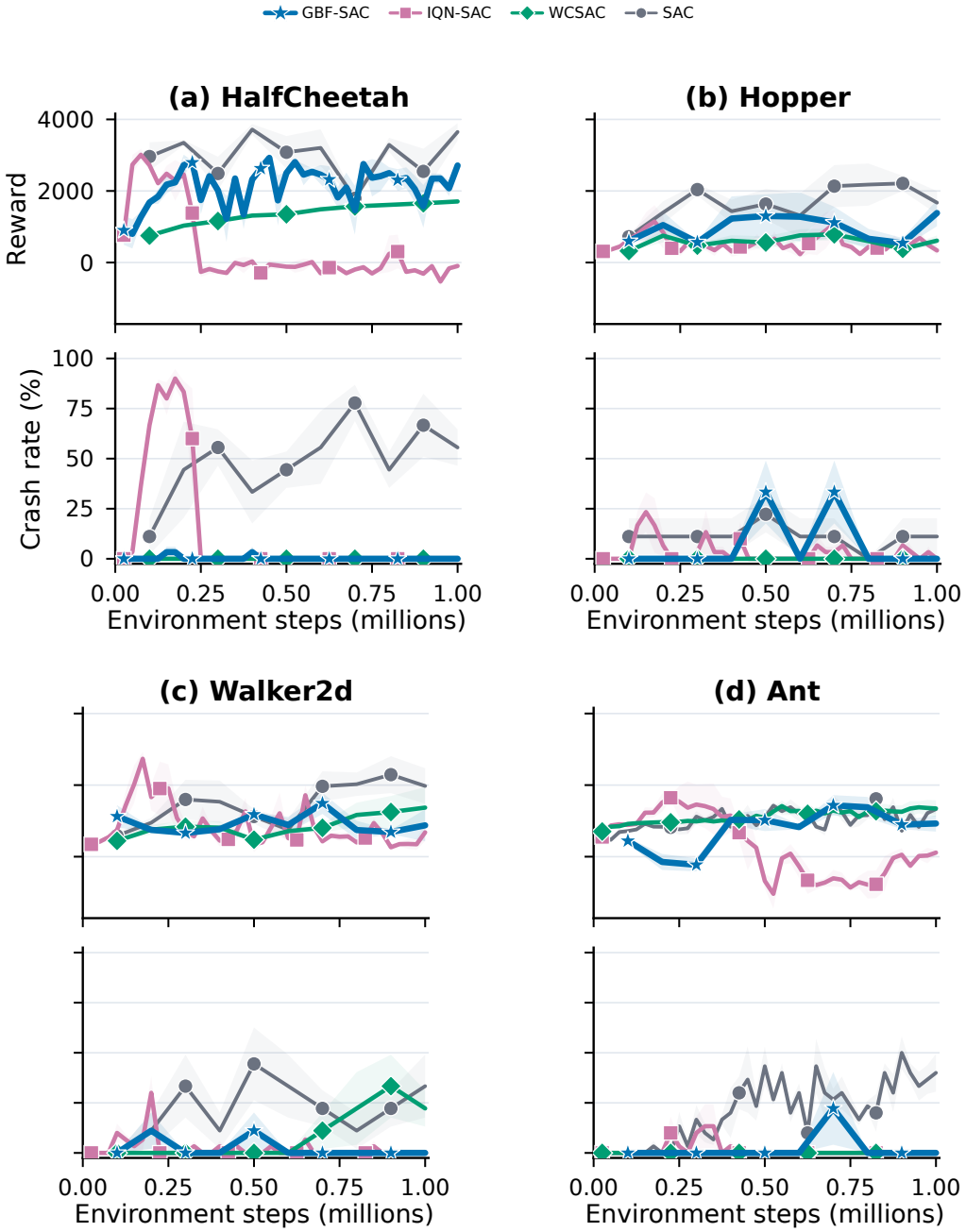


Figure 6: Four-task benchmark risk-policy learning curves, retaining both reward and observed crash frequency.