



Generative Actor-Critic with Soft Bridge Policies

Ke He¹ · Le He¹ · Shunpu Tang² · Yafei Wang³ · Lisheng Fan¹



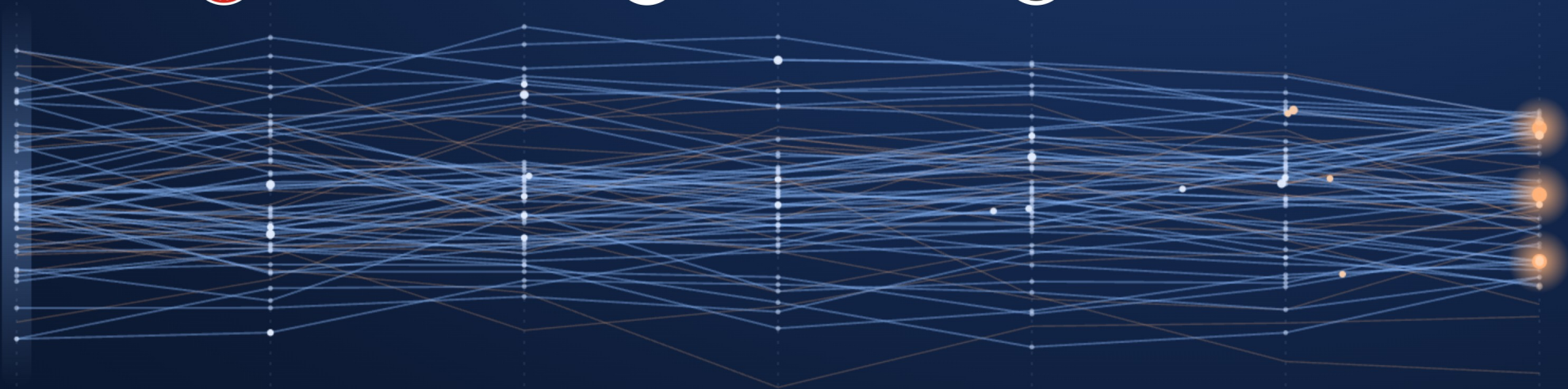
¹Guangzhou University



²Zhejiang University



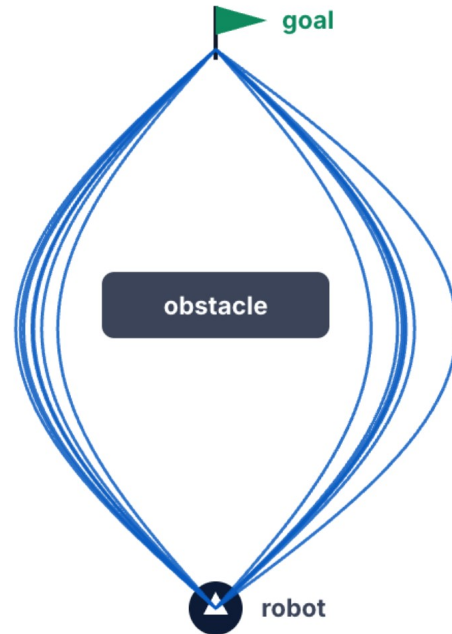
³Southeast University



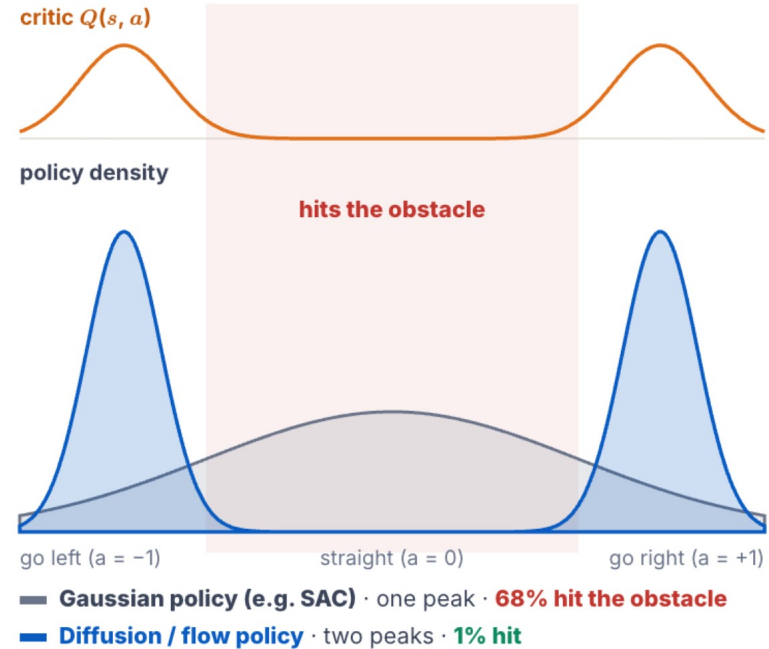
MOTIVATION

Why generative policies?

Example: a robot in front of an obstacle



Action distribution over the steering a



One peak aims **into the obstacle**. Two peaks keep **both good options**.

THE CATCH

Expressive, but with two problems

Diffusion and flow policies keep every good option. But used as actors in maximum-entropy RL, they run into two problems:

1 Cost vs. performance

Many-step policies are strong, but call the same network many times for every action. One-step generators are fast, but usually lose performance.

Usually: faster $\Rightarrow$ weaker

2 Intractable entropy

Any implicit generator, one step or many, is easy to sample but has no tractable density, so the entropy that MaxEnt RL needs cannot be computed.

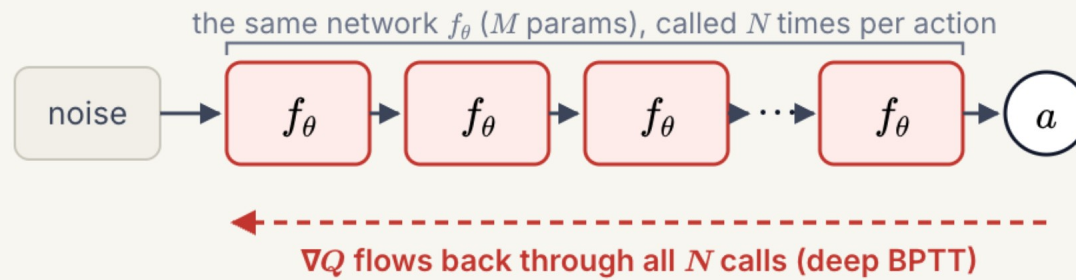
Entropy of the policy = ?

PROBLEM 1 OF 2

Cost vs. performance

Many-step diffusion / flow

N calls per action



One-step generator (black box)

1 call per action



COST VS. RETURN · QUALITATIVE

Many steps

cost / action



return



One step

cost / action



return



↓ returns usually drop

Usually: faster $\Rightarrow$ weaker

Many steps: strong, but N calls per action. One step: fast, but usually weaker. So usually, **faster means weaker**.

PROBLEM 2 OF 2

Intractable entropy

Any implicit generator

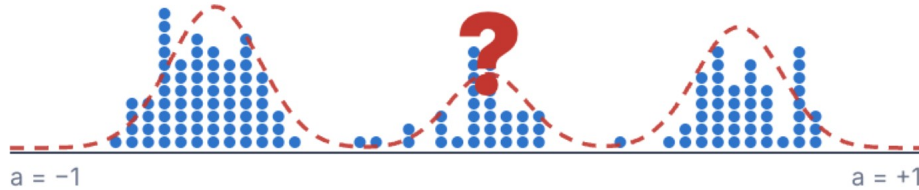
one step or many steps



samples of the action a

✓ sampling is easy

✗ density $\pi(a|s)$ cannot be computed



$$\pi_{\theta}(a | s) = \int p(z) \delta(a - f_{\theta}(z, s)) dz \quad \Rightarrow \quad \mathcal{H}(\pi_{\theta}) = ?$$

Approximate the entropy

INEXACT

Estimate it from samples, or bound it.

e.g. GMM fit (DACER) · Monte Carlo (MaxEntDP) · lower bound (DIME)

Replace it with a surrogate

INDIRECT

Regularize something easier instead.

e.g. kinetic energy (FLAC) · Wasserstein distance (FlowRL)

Neither guarantees the true MaxEnt policy (endpoint-equivalence).

THE QUESTION

**Can one policy be truly soft,
fast, and better, all at once?**

METHOD

Open the box: one call, same budget

SoftGAC: also one call, about the same M params

1 call per action



K small stochastic layers inside

Many-step actor

one network · M params

× N calls

SoftGAC actor

M/K

M/K

M/K

M/K

M/K

M/K

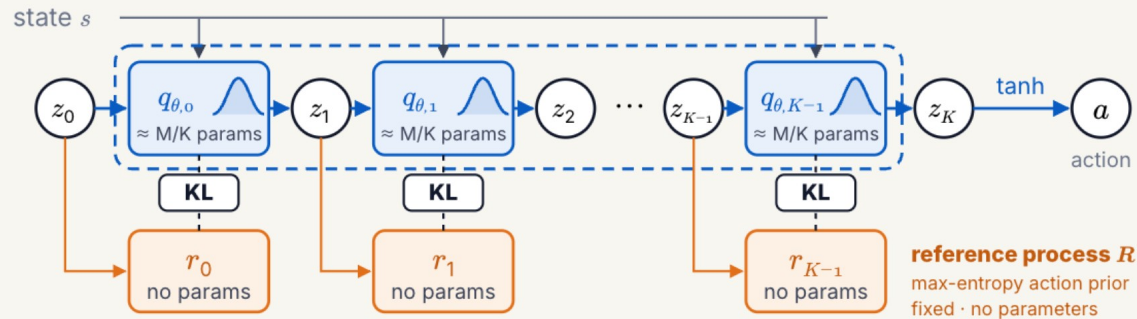
× 1 call

same total budget · actor params (avg. of 3 domains): SoftGAC 0.43M, DIME 0.44M

The many-step budget, one network of M params × N calls, becomes **K small layers of $\approx M/K$ params, each used once.**

METHOD

A soft bridge inside one call



CLOSED-FORM KL OF EACH LAYER (EQ. 24 OF THE PAPER)

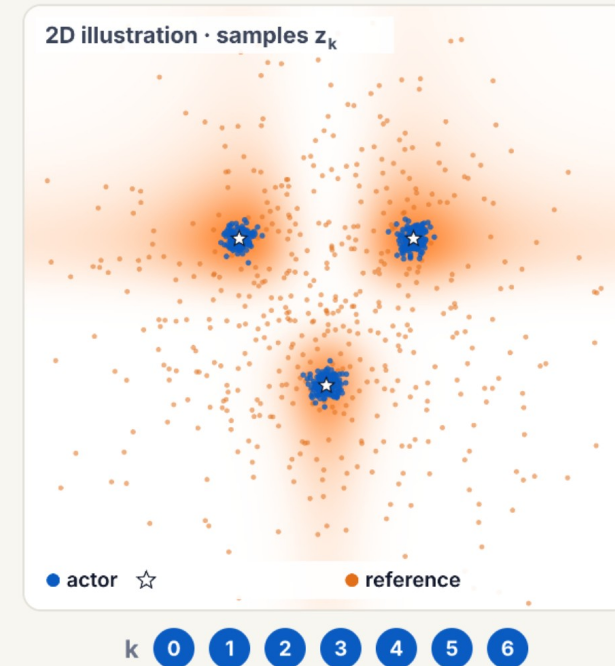
$$\mathcal{C}_k(s, z_k) = \text{KL}(q_{\theta,k} \parallel r_k) = \frac{1}{2} \sum_{i=1}^{d_a} \left[\sigma_{\theta,k,i}^2 + \frac{(\mu_{\theta,k,i} - \mu_{R,k,i})^2}{2h} - 1 - \log \sigma_{\theta,k,i}^2 \right]$$

$$\mu_{\theta,k} = z_k + h u_{\theta,k}(s, z_k), \quad \mu_{R,k} = z_k - 2h \tanh z_k, \quad h = 1/K$$

Control energy $\mathcal{C}_{\theta}(s, \tau) = \sum_k \text{KL}(q_{\theta,k} \parallel r_k)$



= the **exact path KL**, from the same forward pass $\rightarrow$ replaces entropy in the **actor** and **soft Bellman target**



Each Gaussian layer is paired with a max-entropy reference step (no parameters). Both are Gaussian, so every KL is closed-form, and their sum is **the exact soft regularizer, in the same forward pass**.

THEORY

Principled: MaxEnt in path space

REGULARIZE THE WHOLE GENERATION PATH

$$\mathcal{J}_{\text{path}} = \mathbb{E}_{\tau \sim P_{\theta}} [Q(s, \tanh z_K)] - \alpha D_{\text{KL}}(P_{\theta}(\cdot | s) \| R)$$

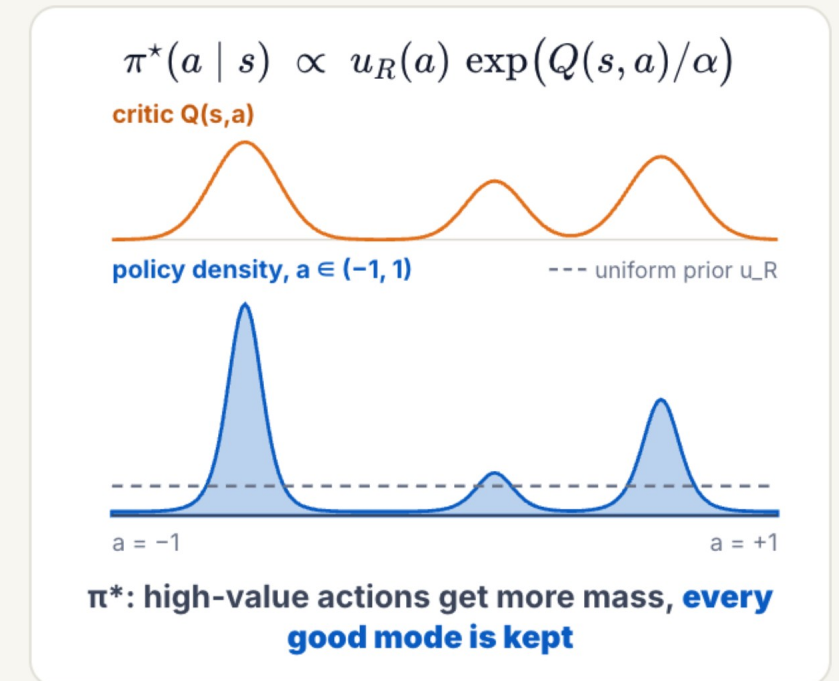
IT CONTAINS SAC'S ENTROPY TERM

$$D_{\text{KL}}(P_{\theta} \| R) = \underbrace{D_{\text{KL}}(\pi_{\theta} \| u_R)}_{\text{endpoint KL (SAC)}} + \underbrace{\mathbb{E}_{a \sim \pi_{\theta}} [D_{\text{KL}}(P_{\theta}(\cdot | a) \| R(\cdot | a))]}_{\text{how } a \text{ is generated}}$$

endpoint KL = SAC's entropy term

how a is generated

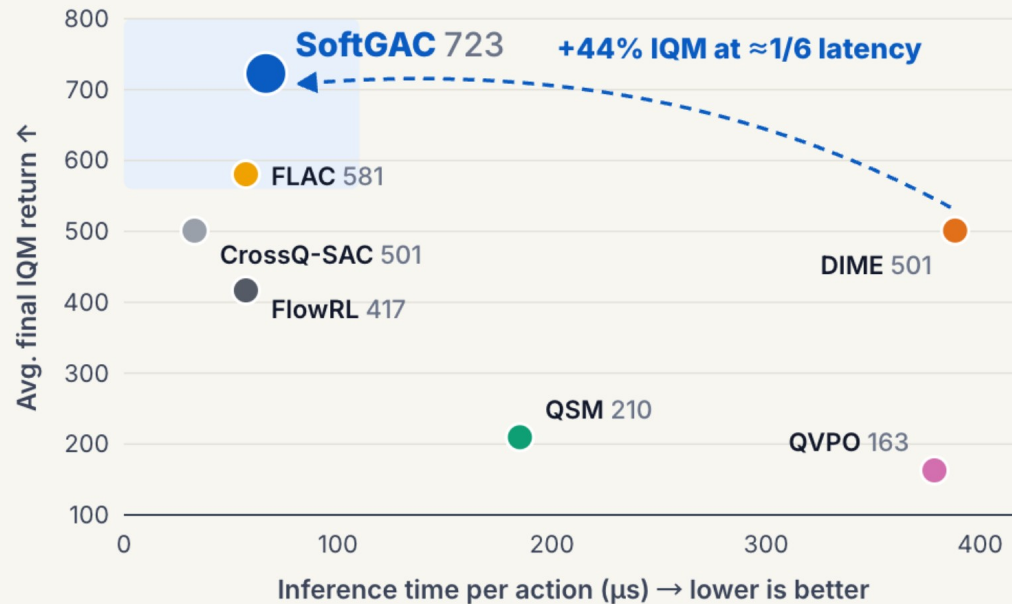
ENDPOINT-EQUIVALENT Uniform $u_R \Rightarrow$ exactly SAC's Boltzmann policy



At a fixed α , Q tilts the uniform prior: the optimum is **exactly SAC's Boltzmann policy** (endpoint-equivalence). No entropy estimates, no surrogates.

RESULTS

Faster and stronger



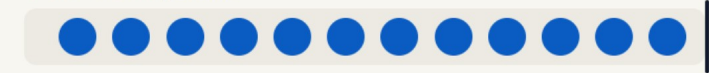
Latency race · H1 Hurdle

actions generated in 800 μs (CPU, one action at a time), similar actor sizes

SoftGAC · 63 μs

1 call · 0.34M params

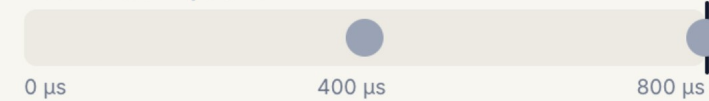
12 actions



20-step diffusion (DACER) · 400 μs

20 calls · 0.32M params

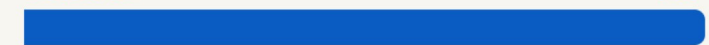
2 actions



Return (IQM) on H1 Hurdle

SoftGAC

737



20-step diffusion (DACER)

377



8 hard tasks: **+44% return** over the strongest many-step policy at **≈1/6** of its time per action. H1 Hurdle: **≈6×** faster and **≈2×** the return of 20-step diffusion.

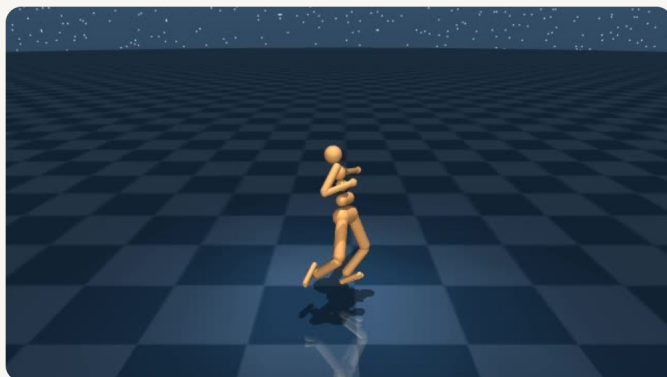
Usual belief: ~~Faster~~ $\Rightarrow$ ~~weaker~~

**Faster does not
have to mean weaker.**

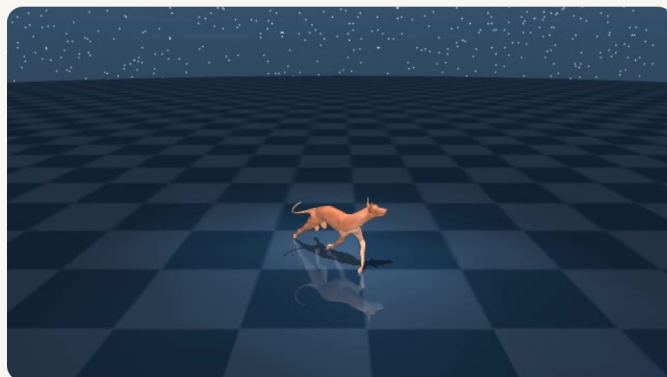
On these tasks, SoftGAC is both faster and stronger than the many-step policies we tested.

RESULTS

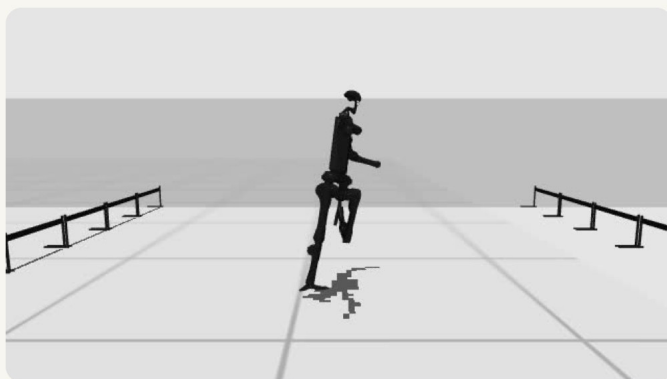
Trained policies in action



Humanoid Run · 21-D actions



Dog Run · 38-D actions



H1 Hurdle · 19-D actions



H1 Run · 19-D actions

1

network call per action

61–75 μ s per action on a laptop CPU

Task	IQM
Humanoid Run	701
Dog Run	755
H1 Hurdle	737
H1 Run	824

SOFTGAC IN THREE WORDS

01

Soft

Principled MaxEnt: an analytical path-KL regularizer, endpoint-equivalent to SAC.

02

Fast

One network call per action, with about the same parameters: $\approx 1/6$ the latency of many-step diffusion.

03

Better

+44% return over the strongest many-step policy, on 8 hard tasks.

One step outside, a **soft bridge** inside.



Generative Actor-Critic with Soft **Bridge** Policies

One step outside, a **soft bridge** inside.

Ke He · Le He · Shunpu Tang · Yafei Wang · Lisheng Fan

github.com/skypitcher/soft_gac_jax

skypitcher.github.io/soft_gac_home

[arXiv:2605.08733](https://arxiv.org/abs/2605.08733)



Scan for code

