Offline Reinforcement Learning with Generative Trajectory Policies

Xinsong Feng · Leshu Tang · Chenan Wang · Haipeng Chen

William & Mary · UCLA

ICML International Conference On Machine Learning

Project Page

Motivation: Expressiveness vs. Efficiency

Generative policies capture rich, **multi-modal** behavior for offline RL — but today's methods force a stark choice:

1. **Diffusion policies** — highly expressive, but **slow** iterative sampling is computationally expensive.
2. **Consistency policies** — remarkably **fast** (1–2 steps), but simplification **degrades** policy quality.

*Can one policy class achieve **both** expressiveness **and** efficiency?*

Contributions

We introduce **Generative Trajectory Policies (GTP)** — a policy that is **both** expressive and efficient, built on a unifying view of modern generative models.

- **Unified ODE view**: diffusion, flow-matching & consistency models are all instances of one continuous-time generative trajectory.
- **GTP paradigm**: learn the **entire** ODE **solution map** — neither slow iterative sampling nor lossy one-step shortcuts.
- **Two principled adaptations** make it practical for offline RL, giving **state-of-the-art on D4RL** — a **perfect 100.0** on `antmaze-umaze`.

A Unified ODE Framework

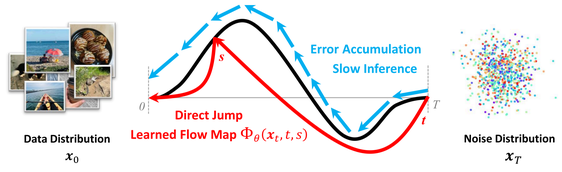
Reverse generation is a deterministic ODE

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = f(\boldsymbol{x}_t, t), \quad t \in [0, T].$$

The true **flow map** $\Phi(\boldsymbol{x}_t, t, s)$ maps any state at time t to time s:

$$\boldsymbol{x}_s = \Phi(\boldsymbol{x}_t, t, s) = \boldsymbol{x}_t + \int_t^s f(\boldsymbol{x}_\tau, \tau)\,\mathrm{d}\tau.$$

Diffusion, Consistency, CTMs, Shortcut & Mean Flows are all **special cases** — approximating aspects of the same Φ.



Two complementary objectives define the framework:

- **Instantaneous Flow Loss** (local anchor) — matches the infinitesimal step:

$$\lim_{s \to t} \varphi(\boldsymbol{x}_t, t, s) = \boldsymbol{x}_t - t f(\boldsymbol{x}_t, t)$$

- **Trajectory Consistency Loss** (global regulator) — multi-step self-consistency:

$$\Phi(\boldsymbol{x}_t, t, s) \approx \Phi(\Phi(\boldsymbol{x}_t, t, u), u, s)$$

GTP: Learning the Whole Trajectory

A **Generative Trajectory Policy** generates actions by learning the ODE solution map $\Phi_{\theta(s,a_t,t,\tau)}$. Making this practical for offline RL requires solving three challenges:

1. **Compute cost** — inner-loop ODE solving for every one of millions of updates.
2. **Instability** — learning from scratch, bad targets bootstrap bad updates.
3. **Misaligned objective** — pure generation = behavior cloning, no improvement.

Score Approximation — Efficient & Stable Training

Replace the self-referential learned field with a **closed-form surrogate** anchored to the offline sample:

$$\tilde{f}(\boldsymbol{x}_t, t) = \frac{\boldsymbol{x}_t - \boldsymbol{x}}{t}.$$

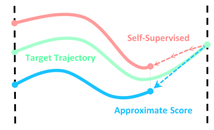
Intermediate points become a **one-step** perturbation, $\boldsymbol{x}_u = \boldsymbol{x} + u \cdot \boldsymbol{z}$, instead of a costly solver.

Theorem. For a p-th order zero-stable solver with max step h,

$$|\mathcal{L}_{\text{prac}}(\theta) - \mathcal{L}_{\text{ideal}}(\theta)| = O(h^p).$$

The surrogate objective coincides with the ideal one as $h \to 0$.

- **Efficient**: no multi-step integration.
- **Stable**: analytic target tied to data breaks error propagation.



Value Guidance — From Imitation to Improvement

Bridge imitation and improvement via the KL-regularized optimal policy

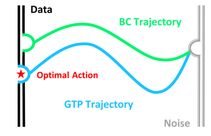
$$\pi^*(a|s) \propto \pi_{\text{BC}}(a|s) \exp(\eta A(s,a)).$$

Matching π_θ to π^* is **exactly** a weighted generative objective — no separate, unstable policy-gradient term:

$$\max_\theta \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\exp(\eta A(s,a)) \cdot \ell_{\text{gen}}(\pi_\theta; a|s)\right].$$

Stabilized weights (normalized, non-negative):

$$w(s,a) = \exp\left(\eta \cdot \frac{\max(0, A(s,a))}{\text{std}(A) + \varepsilon}\right).$$



Training GTP

Algorithm 1 Generative Trajectory Policy

Initialize actor Φ_θ, critic Q_φ, targets $\theta^- \leftarrow \theta$, $\varphi^- \leftarrow \varphi$
for iteration $i = 1$ **to** N_{iter} **do**
  Sample minibatch $(s, a, r, s') \sim \mathcal{D}$
  Update critic Q_φ by the TD error
  Compute advantage weights $w(s,a)$ from the critic
  Sample $t > u > \tau$; noise $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$
  Score approx.: $a_t = a + t\boldsymbol{z}$, $\tilde{a}_u = a + u\boldsymbol{z}$
  Update actor Φ_θ with w-weighted $\mathcal{L}_{\text{actor}} = \mathcal{L}_{\text{Consistency}} + \lambda_{\text{Flow}}\mathcal{L}_{\text{Flow}}$
  EMA-update targets θ^-, φ^-
end for

State-of-the-Art on D4RL — vs. Recent Generative Policies

Full actor-critic; normalized score, 5 seeds ($K = 5$).

Task	FQL	QIPO-D	QIPO-OT	SSCQL	GTP (Ours)
Gym — Locomotion					
halfcheetah-m	55.6	48.2	54.2	52.3	53.9
hopper-m	60.6	89.5	94.0	102.4	90.3
walker2d-m	65.9	85.0	87.6	84.2	89.5
halfcheetah-mr	48.3	45.3	48.0	44.4	50.8
hopper-mr	50.7	101.2	101.2	101.4	101.7
walker2d-mr	38.8	90.1	78.6	85.9	94.2
halfcheetah-me	92.0	94.1	94.4	98.1	93.8
hopper-me	104.6	112.1	108.0	110.9	112.2
walker2d-me	111.7	110.1	110.9	111.1	114.2
Average	**69.8**	**86.2**	**86.3**	**87.9**	**89.0**
AntMaze — Long-Horizon Navigation					
antmaze-umaze	100	97.5	93.6	—	100
antmaze-u-diverse	—	73.9	76.1	—	81.9
antmaze-med-play	—	82.8	80.0	—	83.3
antmaze-med-diverse	85.2	86.0	86.4	—	94.2
antmaze-large-play	—	73.2	55.5	—	53.5
antmaze-large-diverse	—	40.5	32.1	—	71.0
Average	**—**	**77.3**	**72.0**	**—**	**80.6**

- Best **average** on both suites vs. recent methods; **perfect 100.0** on `antmaze-umaze`.

Ablation & Efficiency — Both Pieces Matter, by Design

Ablation — on `hopper-medium-expert` (5 seeds):

Method	Time (h)	Score
GTP (ours)	4.26	**112.2**
w/o score approx. (solver)	5.23	99.7
Shortcut (no teacher)	4.58	76.1
Mean Flows (identity)	6.03	Diverged
w/ linear Q ($\lambda = 0.01$)	5.08	111.4
w/ linear Q ($\lambda = 0.1$)	—	Diverged

Our teacher-based score approx. beats teacher-free objectives (Shortcut, Mean Flows) and ODE solvers; variational weighting avoids the divergence of naïve Q-term guidance.

Efficiency — matched setup: same machine (RTX 4090), 2M steps, batch 256.

Method	Train (h)	Infer (ms)	Steps
Diffusion-QL	7.1	1.16	5
Consistency-AC	3.7	0.55	2
FQL	10.0	0.36	10→1
GTP ($K = 5$)	**4.3**	0.94	5
GTP ($K = 2$)	**3.8**	0.67	2

- Far cheaper to train than Diffusion-QL / FQL; the $K = 2$ variant nearly matches $K = 5$ returns at just **0.67 ms/action**.