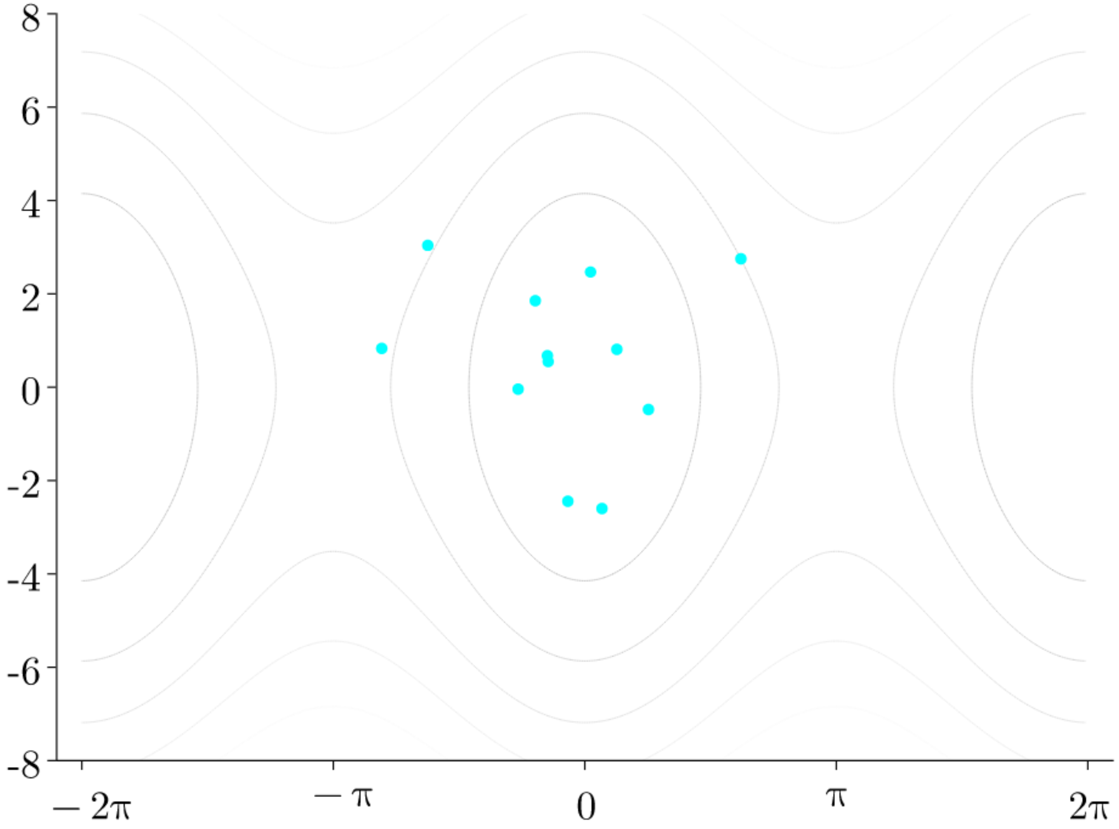
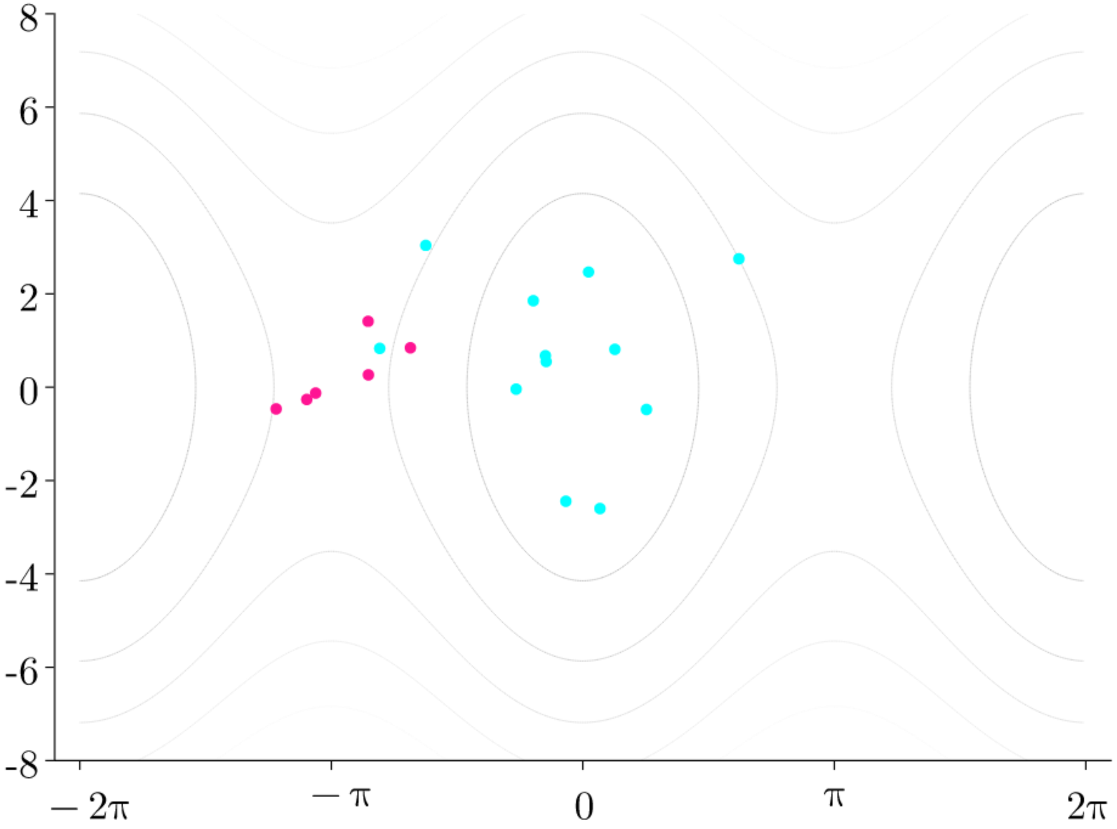
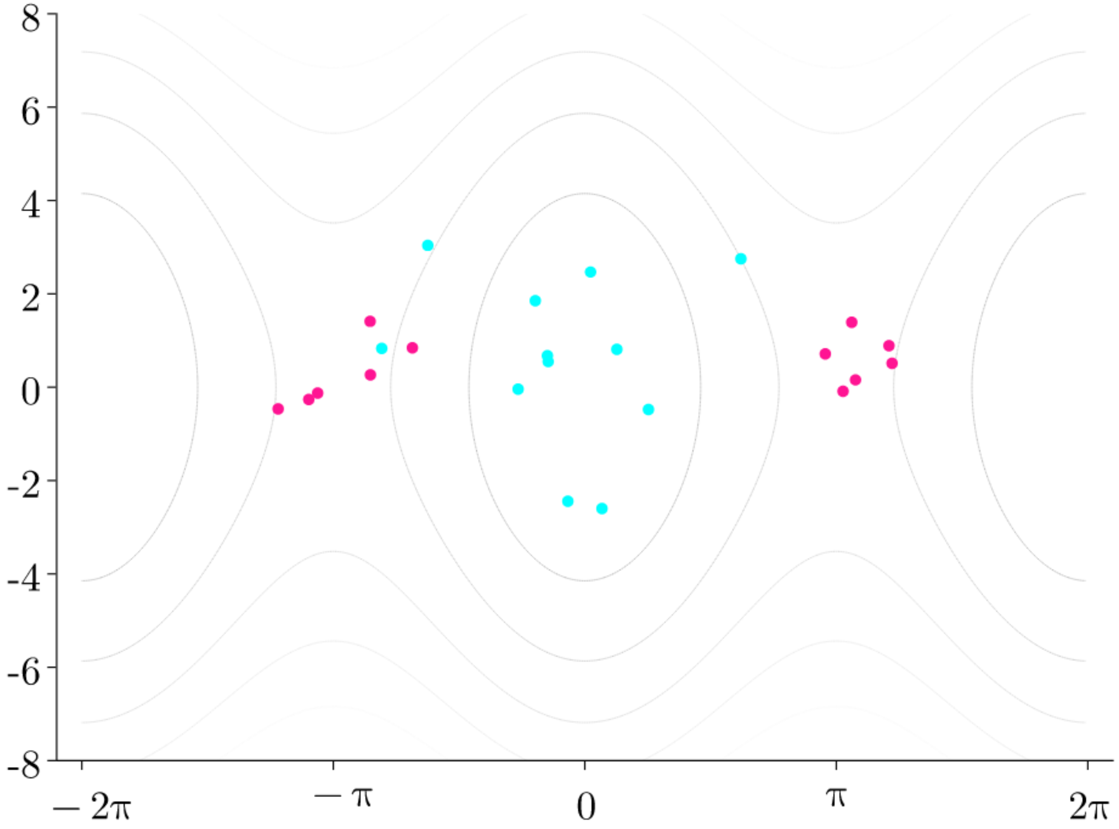
NeuralPbc Sampling State Space 📉



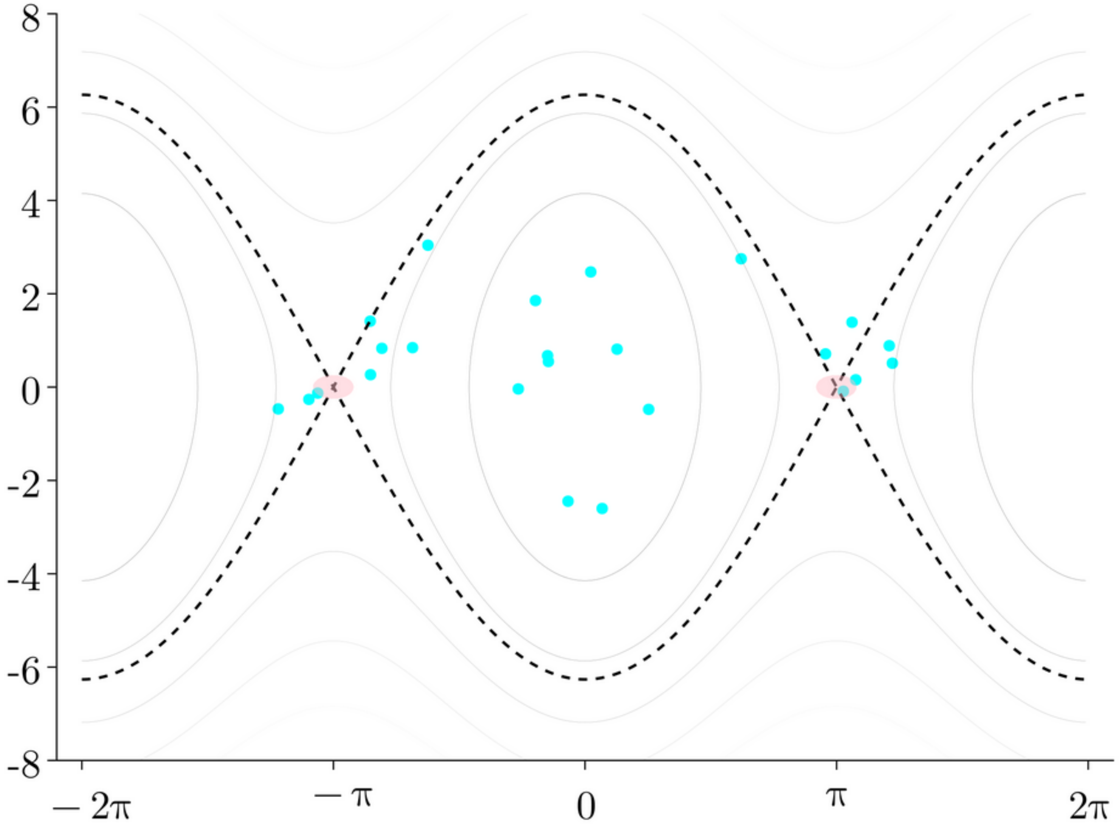
NEURALPBC Sampling State Space 📉



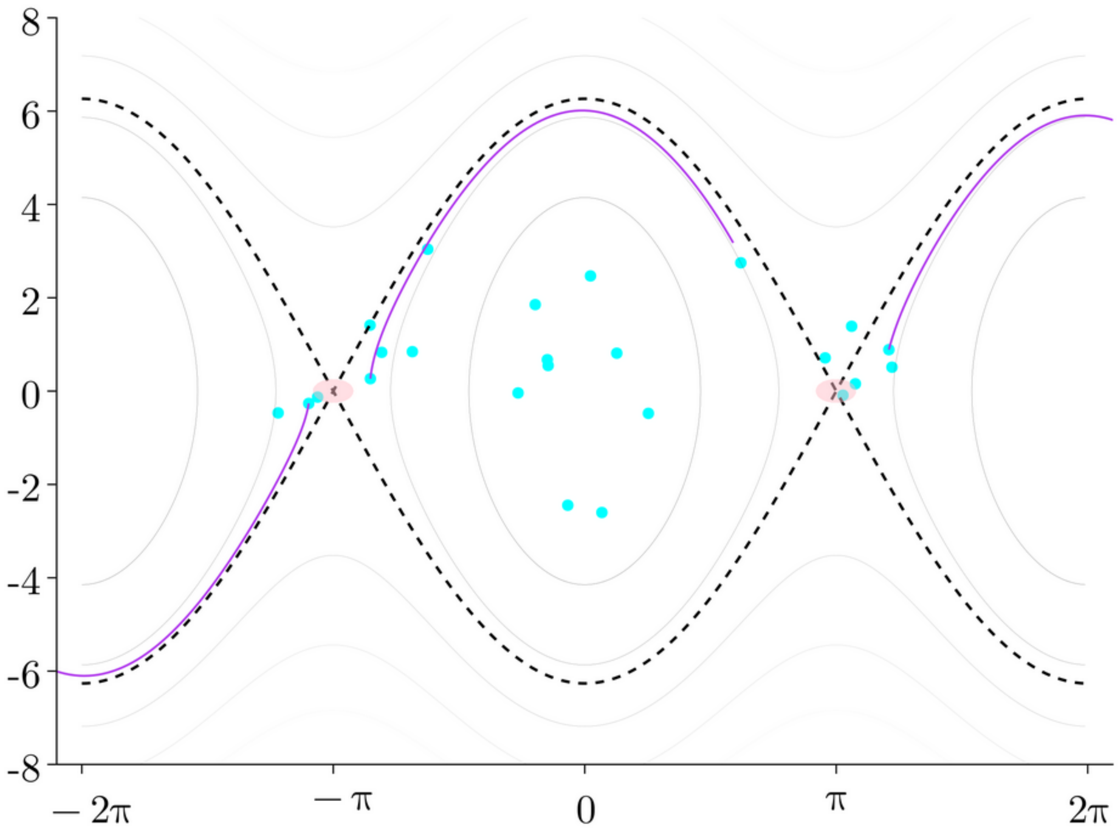
NEURALPBC Sampling State Space 📉



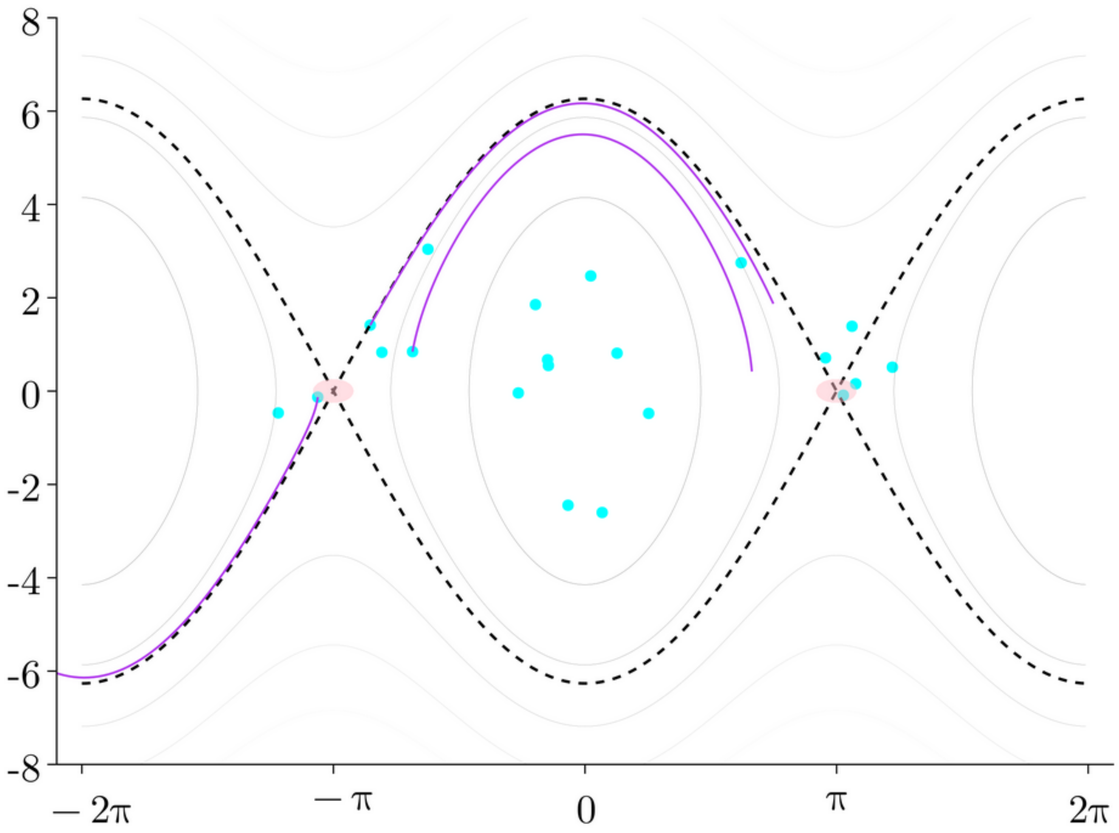
NeuralPbc Sampling State Space 📉



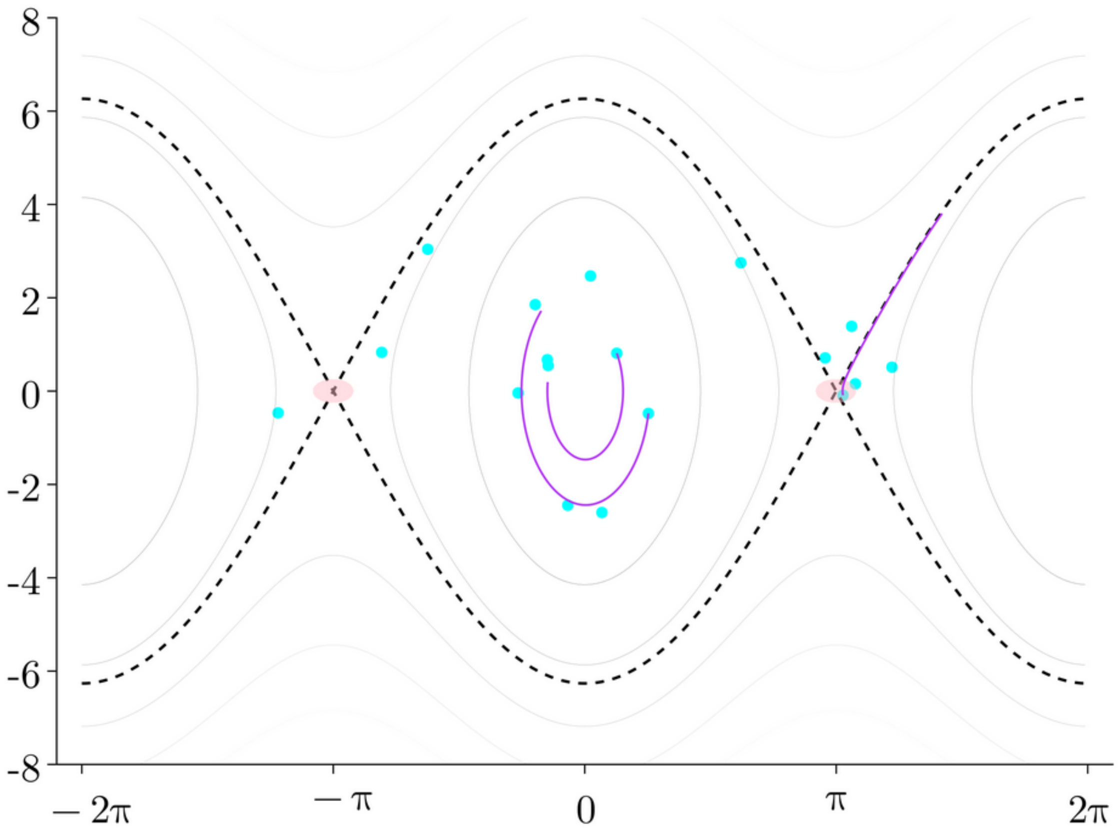
NeuralPbc Sampling State Space 📉



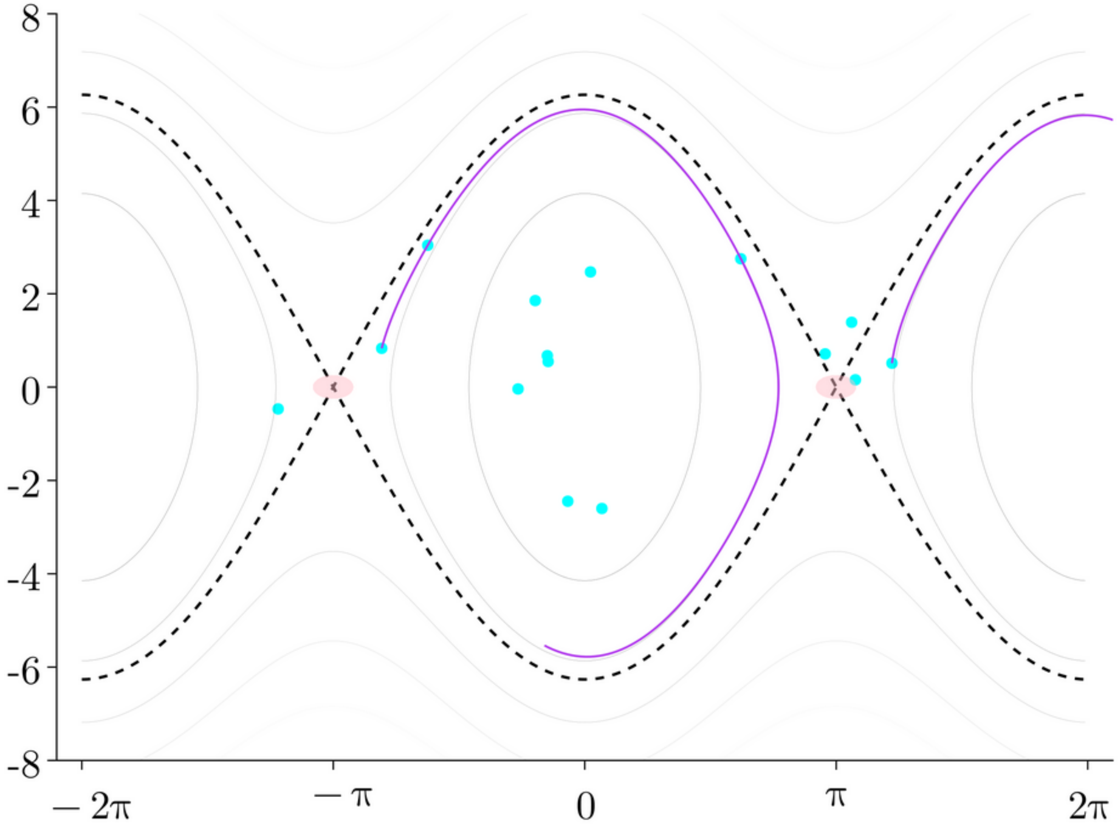
NeuralPbc Sampling State Space 📉



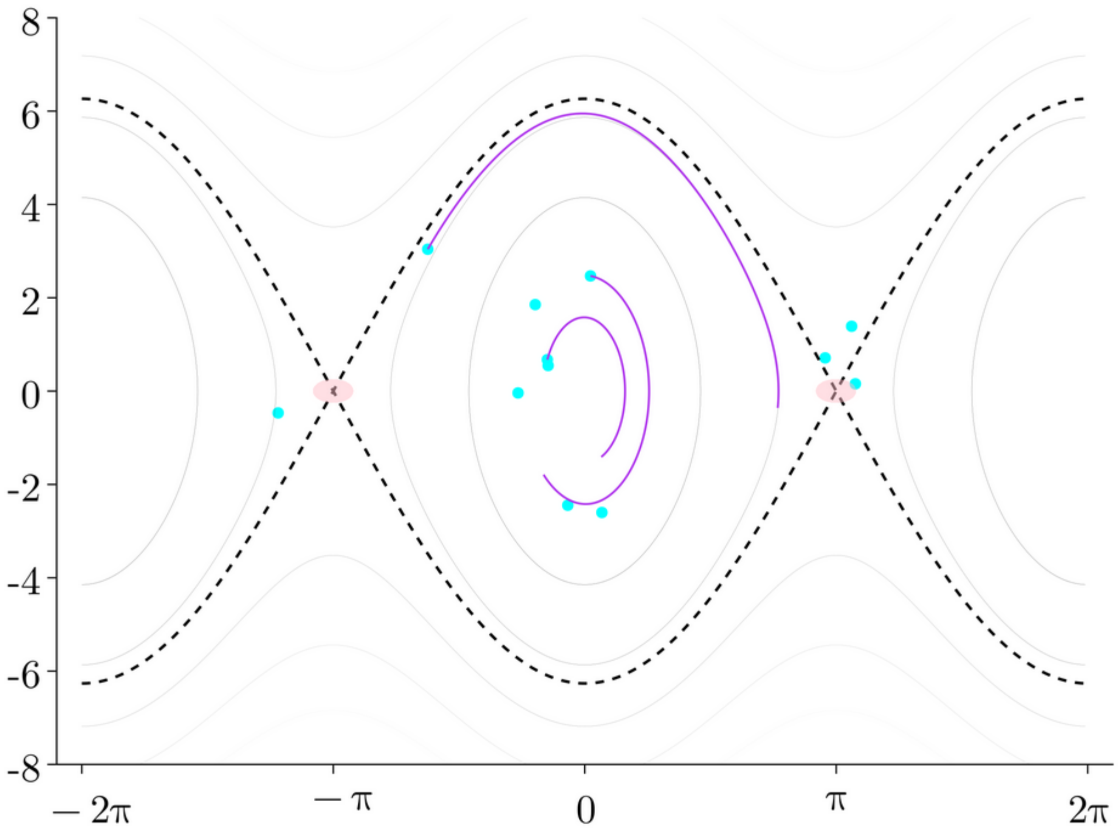
NeuralPbc Sampling State Space 📉



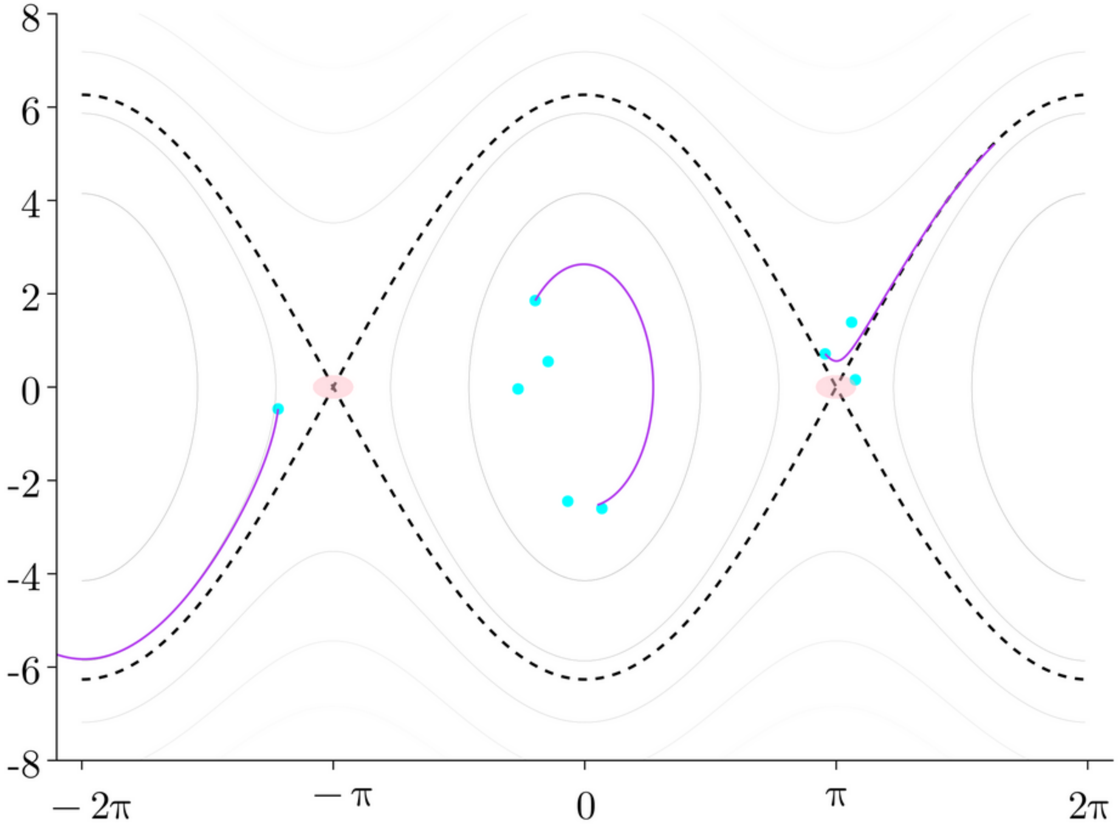
NeuralPbc Sampling State Space 📉



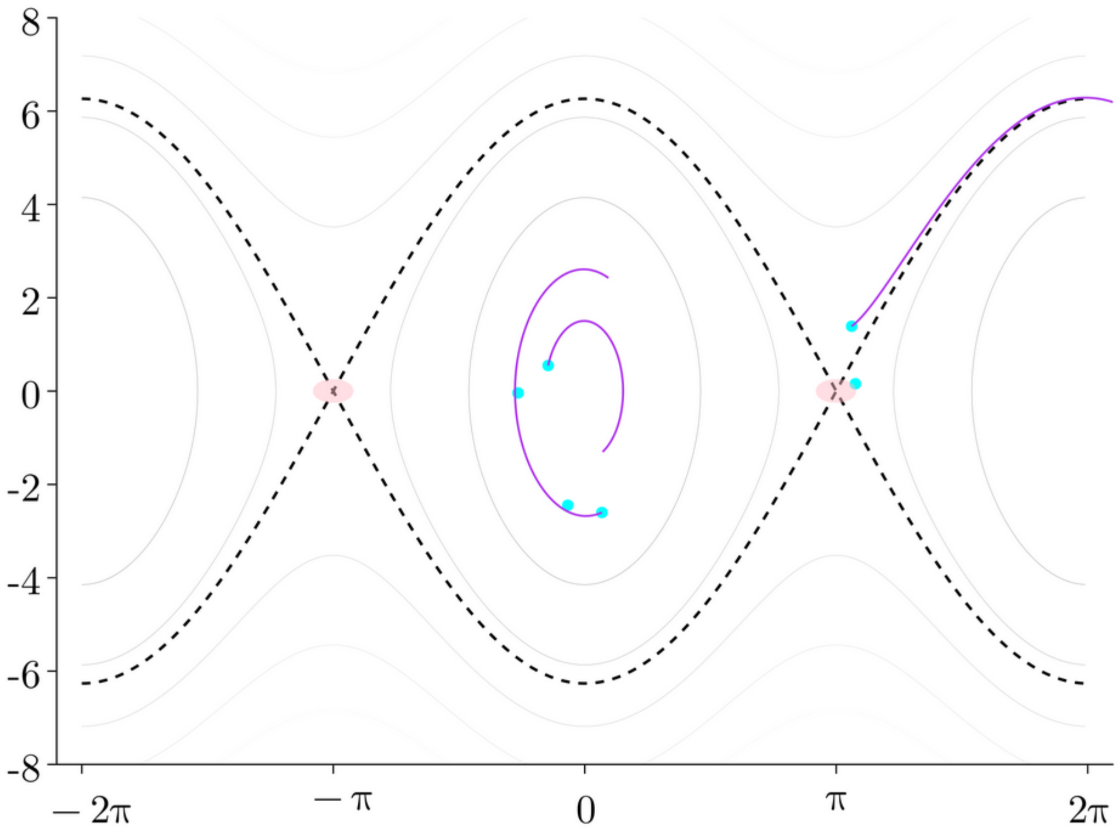
NeuralPbc Sampling State Space 📉



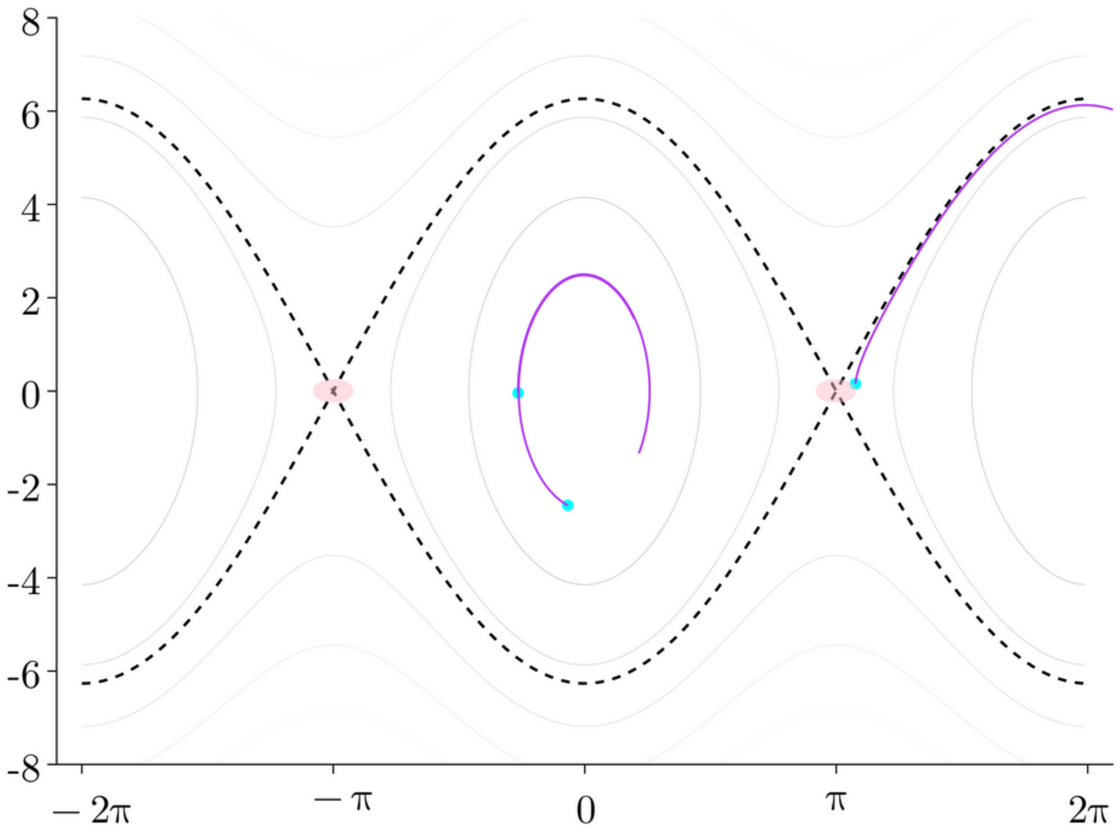
NEURALPBC Sampling State Space 📉



NeuralPbc Sampling State Space 📉



NeuralPbc Sampling State Space 📉



NeuralPbc Cost Function

$$J(\theta, x_0) = \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t$$

$\ell \triangleq \ell_{\text{set}}(\gamma) + \ell_{\perp}(\gamma, u)$ where

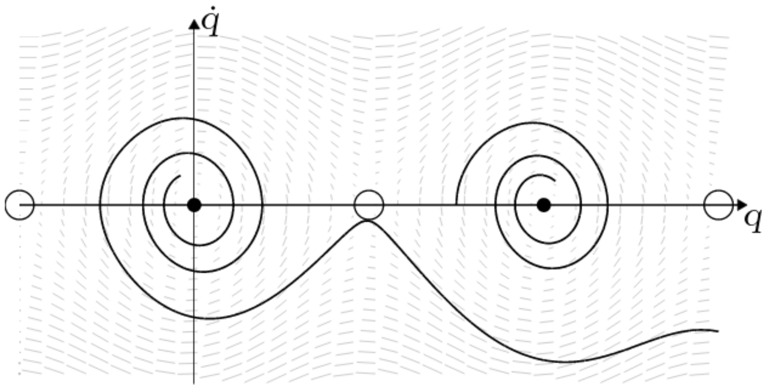
- ϕ is the flow of the equation of motion
- γ is the closed-loop trajectory starting from x_0
- T is the time horizon (hyperparameter)

NeuralPbc Cost Function

$$\ell \triangleq \ell_{\text{set}}(\gamma) + \ell_{\perp}(\gamma, u)$$

Set Distance Loss ℓ_{set}

Penalizes when closed-loop trajectory γ under the current control law is far away from a neighborhood $\mathcal{S}$ of $x^\star$



$$\ell_{\text{set}}(x) = \inf_t \{\|a - b\| : a \in \gamma(t), b \in \mathcal{S}\}$$

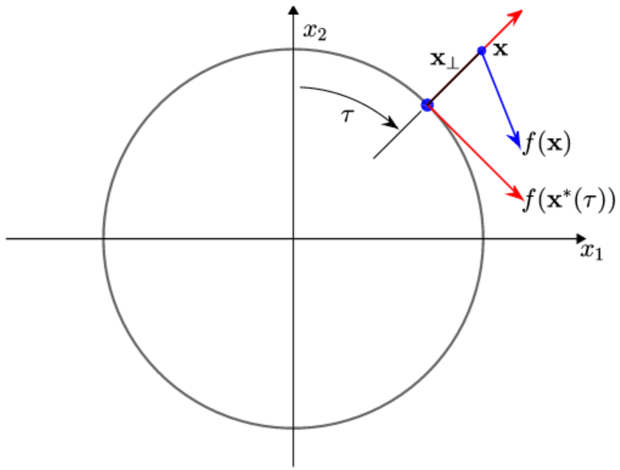
- The set $\mathcal{S}$ may be chosen as
  - A ball around $x^\star$
  - Estimated region of attraction
- No additional loss if any point in γ is in $\mathcal{S}$

NeuralPbc Cost Function

$$\ell \triangleq \ell_{\text{set}}(\gamma) + \ell_{\perp}(\gamma, u)$$

Transversal Distance Loss $\ell_{\perp}$

Measures how close γ is to $\gamma^{\star}$ (expert trajectory) using transverse coordinates $x_{\perp}$



- Coordinate transformation
  - $\tau \in \mathbb{R}$ a surrogate for time
  - $x_{\perp} \in \mathbb{R}^{2n-1}$ quantify how far away the current state is from $\gamma^{\star}$
- By construction $x_{\perp} \to 0 \Leftrightarrow \gamma = \gamma^{\star}$

$$\ell_{\perp} = x_{\perp}^{\top} Q x_{\perp} + u^{\top} R u,\ Q \succcurlyeq 0,\ R \succ 0$$

- No preferred orbit? $Q = 0$

Backprop through ODE Solutions

We need $\partial J/\partial \theta$, which depends ODE solutions

Backprop through ODE Solutions

We need $\partial J / \partial \theta$, which depends ODE solutions

😿 Combining `autodiff` with numerical ODE solvers

Backprop through ODE Solutions

We need $\partial J / \partial \theta$, which depends ODE solutions

😿 Combining `autodiff` with numerical ODE solvers

😿 Adjoint sensitivity method: solve the adjoint problem backward in time

$$\frac{\mathrm{d}\lambda}{\mathrm{d}t} = -\lambda \frac{\partial f}{\partial x}, \quad \frac{\partial J}{\partial \theta} = \lambda(t_0) \frac{\partial f}{\partial x}$$

Backprop through ODE Solutions

We need $\partial J / \partial \theta$, which depends ODE solutions

😿 Combining `autodiff` with numerical ODE solvers

😿 Adjoint sensitivity method: solve the adjoint problem backward in time

$$\frac{\mathrm{d}\lambda}{\mathrm{d}t} = -\lambda \frac{\partial f}{\partial x}, \quad \frac{\partial J}{\partial \theta} = \lambda(t_0) \frac{\partial f}{\partial x}$$

😺 Adjoint methods + `autodiff` implemented in `DiffEqFlux.jl`

Robust Control Under Uncertainties

Optimal Control under System Parameter Uncertainties

$$
\begin{aligned}
\underset{\theta}{\text{minimize}} \quad & J(\theta, x_0) && = \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t \\
\text{subject to} \quad & \dot{x} && = f\left(x, u^\theta; p\right) \\
& p && \sim \mathcal{N}\left(\widehat{p}, \sigma_p\right)
\end{aligned}
$$

Robust Control Under Uncertainties

Optimal Control under System Parameter and Measurement Uncertainties

$$
\begin{aligned}
\underset{\theta}{\text{minimize}} \qquad J(\theta, x_0) &= \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t \\
\text{subject to} \qquad dx &= f\left(x, u^\theta\right) dt + \nabla_x u(x) dW_t \\
p &\sim \mathcal{N}\left(\widehat{p}, \sigma_p\right)
\end{aligned}
$$