Diffusion Policy

Thursday, August 7, 2025 11:39 AM

Diffusion Policy

Core idea:

- Diffusion Policy applies diffusion models (from img generation) to generate robot trajectories & control policies
- Model learns to refine random noise into coherent action sequence by iteratively denoising

Simple Example (1D)

Step 1 - Setup State Space

[S ... G ...]

$S = [0, 10]$

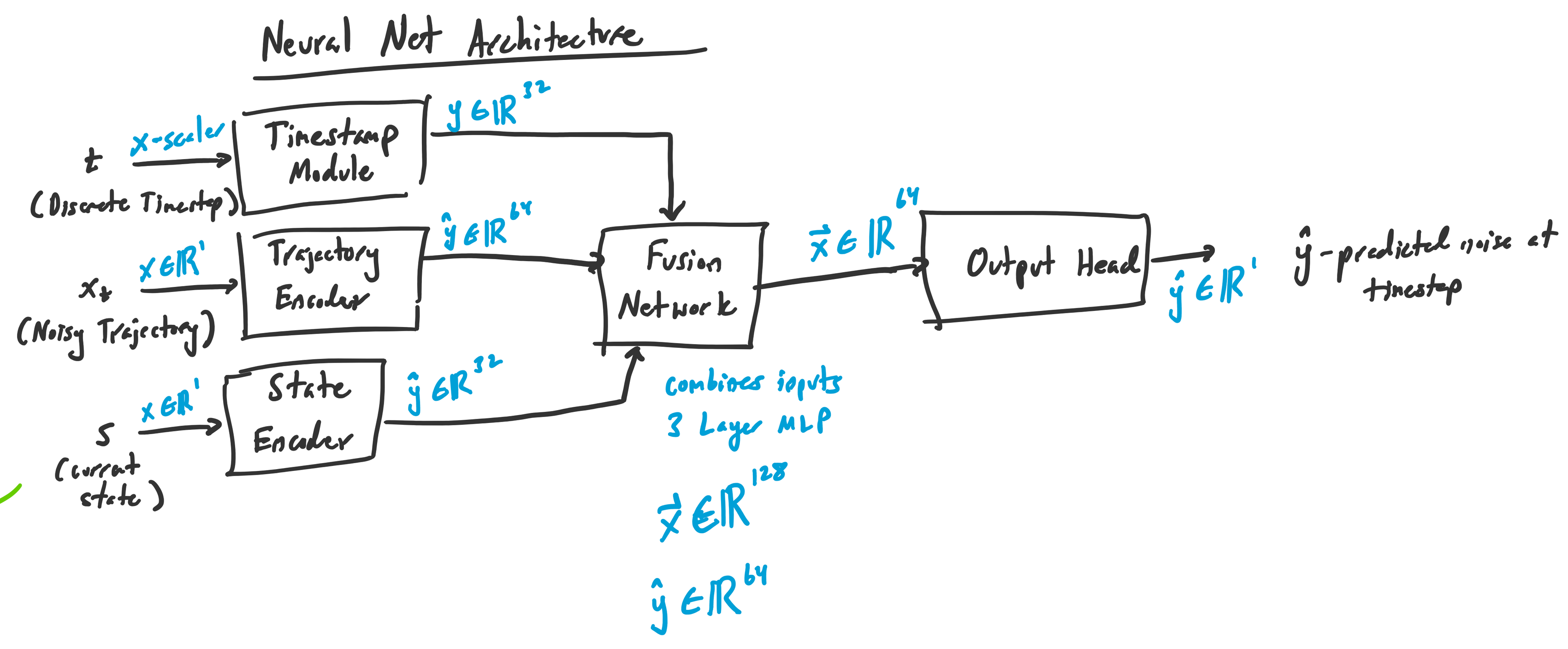
Action space = $[-5, +5]$ — Continuous Action Space

Trajectory = $[a_1]$ — Timesteps

Goal: Generate trajectory $0 \to 3$ in 1 timestamp

Perfect Trajectory = [+3.0]

Neural Net Architecture



- Timestep Module - Convert discrete timestep to continuous
- Trajectory Encoder - Process noisy action sequence
- State Encoder - Process the context robot current state & goal
- Fusion Network - combine all variable inputs
- Output Head - Predict noise to remove

Simplified to $f()$

Neural Net:

f(noisy_trajectory, timestep, current_state) → predicted_noise
(NT, T, CS)

$L = \|\hat{\varepsilon} - \varepsilon\|^2$ (MSE)

Training Phase:

Step 1: Collect good train data

sample 1

	Sample 1	Sample 2	Sample 3
Initial State:	0.0	0.0	0.0
Goal:	3.0	3.0	3.0
Good Trajectory:	[+3.0]	[+2.8]	[+3.3]
Result:	0.0+3.0=3	0+2.8=2.8	0+3.3=3.3

Step 2: Forward Diffusion (Training adding noise)

- Noise schedule
  - timestep 0: No noise
    - 1: $x_0 + N(0, \sigma = .3)$
    - 2: $x_1 + N(0, \sigma = .7)$
    - 3: $x_2 + N(0, \sigma = 1.5)$

Add noise from normal distribution. This makes it easier for the NN's to learn the noise

For [+3.0] — NN training ([NT, T, CS], true noise)
($\vec{x}$ = [NT, T, CS], y = true noise)

$t = 0$ (clean)
$x_0 = [3.0]$

$t = 1$ (light noise)
$x_1 = x_0 + N(0, .3^2)$
$x_1 = [+3.2]$
([[+3.2], 1, 0.0], [+.2])
"When I see [+3.2] at t=1 w/ CS = 0.0, the noise is [+.2]"

$t = 2$ (medium)
$x_2 = x_1 + N(0, .7^2)$
$x_2 = [+3.7]$
([[+3.7], 2, 0.0], [+.5])
"When I see [+3.7] at t=2 w/ CS = 0.0, the noise is [+.5]"

$t = 3$ (heavy)
$x_3 = x_2 + N(0, 1.5^2)$
$x_3 = [+4.8]$
([[+4.8], 3, 0.0], [+1.1])
"When I see [+4.8] at t=3 w/ CS = 0.0, the noise is [+1.1]"

Generation Phase

Step 1 - Start w/ noise

CS - 0.0 (Start position)

Goal: 3.0

Random starting trajectory: $x_3 = [+5.1]$

Would result in 0.0 + 5.1 = 5.1 >> 3.0 overshoot!

Step 2 - Reverse Diffusion (Remove Noise iteratively)

$t = 3 \to 2$ (Remove heavy noise)

Noisy_trajectory: [+5.1]
timestep: 3
current state: 0.0

f([+5.1], 3, 0.0) = [+1.1] (predicted noise)

$x_2 = x_3 -$ noise = [+5.1] = [+4.0]

$t = 2 \to 1$ (Remove medium noise)

f([+4.0], 2, 0.0) = [+.5]

$x_1 = x_2 -$ noise = [+3.5]

$t = 1 \to 0$

f([+3.5], 1, 0.0) = [+.2]

$x_0 = x_1 -$ noise = [+3.3]

Goal Check: $|3.3 - 3.0| = .3$ very close!

Execution Phase

generated trajectory = [+3.3]

execute [+3.3]

Assess:

State: 3.3

Goal: 3.0

Goal Achieved! (otherwise replan!)