Training Algorithm

Part 3/7: Reverse Process

1

Sample Data

$x_0 \sim p\_data$

2

Sample Timestep

$t \sim \text{Uniform}$

$\{1, \ldots, T\}$

3

Sample Noise

$\varepsilon \sim N(0, I)$

4

Add Noise

$$x\_t = \sqrt{\bar{\alpha}\_t} \cdot x_0 + \sqrt{(1-\bar{\alpha}\_t)} \cdot \varepsilon$$

5

Predict Noise

Neural Network

$\hat{\varepsilon} = \varepsilon\_\theta(x\_t, t)$

6

Compute Loss

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

7

Backprop

Update parameters θ

8

Repeat Until Convergence

Loop through steps 1-7

💡 Core Training Loop: Sample → Add Noise → Predict → Optimize → Repeat

Concrete Calculation Example

2D Case with Actual Numbers

📝 Step-by-Step Calculation

Step 1: Sample Data Point

$x_0 = [3.0, 2.0]$

Original data point from training set

Step 2: Sample Timestep

$t = 500$ (out of $T=1000$)

Random timestep, halfway through diffusion

Step 3: Sample Noise

$\varepsilon \sim N(0, I) = [0.8, -1.2]$

Random Gaussian noise vector

Step 4: Add Noise (Forward Process)

$\bar{\alpha}_{500} = 0.5$ (example)

$$x_t = \sqrt{\bar{\alpha}_t} \cdot x_0 + \sqrt{(1-\bar{\alpha}_t)} \cdot \varepsilon$$

$$x_{500} = \sqrt{0.5} \cdot [3.0, 2.0] + \sqrt{0.5} \cdot [0.8, -1.2]$$

$$x_{500} = [2.12, 1.41] + [0.57, -0.85]$$

$$x_{500} = [2.69, 0.56]$$

Step 5: Predict Noise

$\hat{\varepsilon} = \varepsilon_\theta(x_{500}, 500) = [0.75, -1.15]$

Neural network prediction

Step 6: Compute Loss

```
L = ||ε - ε↑||²
L = ||[0.8, -1.2] - [0.75, -1.15]||²
L = ||[0.05, -0.05]||²
```

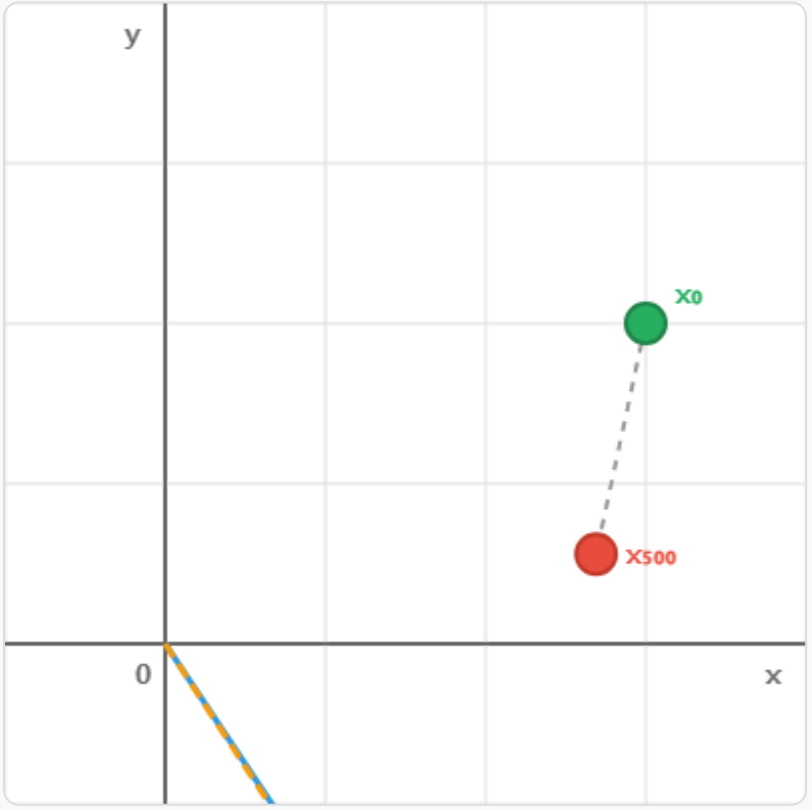
L = 0.0025 + 0.0025 = 0.005

Step 7: Update Parameters

θ ← θ - α·▽_θ L

Gradient descent to minimize loss

📊 Visual Representation



Legend

- x_0 = [3.0, 2.0] - Original data
- x_{500} = [2.69, 0.56] - Noisy data
- ε = [0.8, -1.2] - True noise
- $\hat{\varepsilon}$ = [0.75, -1.15] - Predicted noise

Goal: Make $\hat{\varepsilon}$ closer to ε by updating θ