Fast and Efficient Creation of an Universal Quantum Gate Set Using Reinforcement Learning Methods

Pablo Dario Conte

Democritus University of Thrace (DUTh)

March 2025

DEMOCRITUS UNIVERSITY OF THRACE | DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

MSc in QUANTUM COMPUTING AND QUANTUM TECHNOLOGIES

NATIONAL CENTRE FOR SCIENTIFIC RESEARCH "DEMOKRITOS"

Agenda

Introduction

Introduction

This master thesis presentation explores the use of reinforcement learning (RL) methods to efficiently construct a set of universal quantum gates comprising:

- Hadamard (H)
- $\pi/4$ phase shift (T)
- Controlled NOT (CNOT)

These gates, recognized for their minimality and universality, serve as essential building blocks for arbitrary quantum operations.

Objectives

Specifically, this work seeks to:

- Formulate quantum gate control as a reinforcement learning problem.
- Deploy and compare the performance of various RL agents for optimizing quantum control.
- Validate the proposed methods through numerical simulations, compare them to optimal control methods, and analyze their scalability to multi-qubit systems.

Gate Set

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

The Reinforcement Learning frameworks deployed are:

- Double Dueling Deep Q-Network (DDDQN)
- Proximal Policy Optimization (PPO)
- Group Policy Relative Optimization (GRPO)
- Twin Delayed Deep Deterministic Policy Gradient (TD3)

Quantum Mechanics and Control

Quantum Control

Quantum gates operate on qubits and are represented by unitary matrices. For example:

- the Hadamard Gate (H) creates a superposition from a basis state,
- the T Gate (T) introduces a $\pi/4$ phase shift and
- the Controlled-NOT Gate (CNOT) acts on two qubits, flipping the second qubit if the first is $|1\rangle$.

Quantum control is the process of manipulating quantum systems to achieve specific objectives, such as implementing gates with high fidelity. Control is achieved by tuning the system parameters.

The evolution of a quantum system is governed by the Schrödinger equation expressed as follows:

$$i\frac{\partial}{\partial t}\left|\psi(t)\right\rangle = \hat{H}\left|\psi(t)\right\rangle$$

where $\hat{H}$ is the Hamiltonian operator that dictates the system's dynamics.

The Hamiltonian will be determined by the control parameters

$$\hat{H} = f(\Omega, \Delta, J)$$

Ω : Rabi Frequency

Δ : Detuning

J : Coupling Strength

Challenges in Quantum Control

Quantum systems are sensitive to interactions with their environment.

Precise control of the Ω, Δ, ϕ or J pulses is required to realize unitary transformations.

Longer gate times increase the likelihood of decoherence, which requires optimization to reduce execution time.

Time Evolution

$$\hat{U}(t) = e^{-i\hat{H}t}$$

Time is just a parameter in quantum mechanics, not an operator.
— Sakurai & Napolitano

Single Qubit Hamiltonian

$$\hat{H}_{\text{RWA}}(t) = \frac{1}{2}\left(\Delta\,\hat{\sigma}_z + \Omega(t)e^{i\phi}\,\hat{\sigma}_+ + \Omega(t)e^{-i\phi}\,\hat{\sigma}_-\right)$$

$$\hat{\sigma}_\pm = \frac{1}{2}\left(\hat{\sigma}_x \pm i\,\hat{\sigma}_y\right)$$

$$\hat{\sigma}_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \hat{\sigma}_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \hat{\sigma}_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

$$\hat{\sigma}_+ = \frac{1}{2}\left(\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + i\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\right) = \frac{1}{2}\left(\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\right) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$\hat{\sigma}_- = \frac{1}{2}\left(\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} - i\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\right) = \frac{1}{2}\left(\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\right) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

Two Qubits Hamiltonian

A general two-qubit Hamiltonian can be written as

$$H = \sum_{\alpha,\beta \in \{0,x,y,z\}} c_{\alpha\beta}\, \sigma_\alpha \otimes \sigma_\beta,$$

Expanded explicitly, the Hamiltonian becomes

$$\begin{aligned} H = {} & c_{00}\, I \otimes I + \Omega_{0x}\, I \otimes \sigma_x + \Omega_{0y}\, I \otimes \sigma_y + \Delta_{0z}\, I \otimes \sigma_z + \\ & \Omega_{x0}\, \sigma_x \otimes I + J_{xx}\, \sigma_x \otimes \sigma_x + J_{xy}\, \sigma_x \otimes \sigma_y + J_{xz}\, \sigma_x \otimes \sigma_z + \\ & \Omega_{y0}\, \sigma_y \otimes I + J_{yx}\, \sigma_y \otimes \sigma_x + J_{yy}\, \sigma_y \otimes \sigma_y + J_{yz}\, \sigma_y \otimes \sigma_z + \\ & \Delta_{z0}\, \sigma_z \otimes I + J_{zx}\, \sigma_z \otimes \sigma_x + J_{zy}\, \sigma_z \otimes \sigma_y + J_{zz}\, \sigma_z \otimes \sigma_z. \end{aligned}$$

$J_{zx}\,\hat{\sigma}_z^{(c)} \otimes \hat{\sigma}_x^{(t)}$: Flips the target qubit

$J_{xz}\,\hat{\sigma}_x^{(c)} \otimes \hat{\sigma}_z^{(t)}$: Phase-shifts the target qubit

$J_{xy}\,\hat{\sigma}_x^{(c)} \otimes \hat{\sigma}_y^{(t)}$: Combines bit and phase flip on the target

$J_{yx}\,\hat{\sigma}_y^{(c)} \otimes \hat{\sigma}_x^{(t)}$: Yields a phase-dependent flip on the target

$J_{xx}\,\hat{\sigma}_x^{(c)} \otimes \hat{\sigma}_x^{(t)}$: Promotes simultaneous bit flips.

$J_{yy}\,\hat{\sigma}_y^{(c)} \otimes \hat{\sigma}_y^{(t)}$: Yields correlated phase rotations.

$J_{yz}\,\hat{\sigma}_y^{(c)} \otimes \hat{\sigma}_z^{(t)}$: Modulates target's detuning via control's phase

$J_{zy}\,\hat{\sigma}_z^{(c)} \otimes \hat{\sigma}_y^{(t)}$: Phase-dependent modulation of the target's state

$J_{zz}\,\hat{\sigma}_z^{(c)} \otimes \hat{\sigma}_z^{(t)}$: Both qubits interact via their detuning

$$\sigma_x \otimes \sigma_x = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad \sigma_x \otimes \sigma_y = \begin{pmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \end{pmatrix} \quad \sigma_x \otimes \sigma_z = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$$

$$\sigma_y \otimes \sigma_x = \begin{pmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & -i & 0 \\ 0 & i & 0 & 0 \\ i & 0 & 0 & 0 \end{pmatrix} \quad \sigma_y \otimes \sigma_y = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad \sigma_y \otimes \sigma_z = \begin{pmatrix} 0 & 0 & -i & 0 \\ 0 & 0 & 0 & i \\ i & 0 & 0 & 0 \\ 0 & -i & 0 & 0 \end{pmatrix}$$

$$\sigma_z \otimes \sigma_x = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix} \quad \sigma_z \otimes \sigma_y = \begin{pmatrix} 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \\ 0 & 0 & 0 & i \\ 0 & 0 & -i & 0 \end{pmatrix} \quad \sigma_z \otimes \sigma_z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H = \frac{1}{2}\left(\Delta_c\,\hat{\sigma}_z^{(c)} + \Delta_t\,\hat{\sigma}_z^{(t)} + \Omega_c\,e^{i\phi}\,\hat{\sigma}_x^{(c)} + \Omega_t\,e^{i\phi}\,\hat{\sigma}_x^{(t)} + J_{zx}\,\hat{\sigma}_z^{(c)} \otimes \hat{\sigma}_x^{(t)}\right)$$

Hadamard Gate Evolution

A common strategy:

$$\hat{n} = \frac{1}{\sqrt{2}}(\hat{x} + \hat{z})$$

Hamiltonian of one qubit:

$$H = \frac{1}{2}\Big(\Delta\,\sigma_z + \Omega_x\,\sigma_x\Big)$$

One often chooses

$$\Omega_x = \Delta = -1$$

so that

$$H = -\frac{1}{2}(\sigma_z + \sigma_x)$$

Hadamard Gate Evolution

The corresponding time evolution operator is

$$U(t) = e^{-i(-\frac{1}{2}(\sigma_z+\sigma_x))t}$$

A calculation shows that if the pulse duration is set to

$$t = \frac{\pi}{\sqrt{2}}$$

the resulting operator is (up to a global phase) equivalent to the Hadamard gate. In other words,

$$e^{-i\left(-\frac{1}{2}(\sigma_z+\sigma_x)\right)\frac{\pi}{\sqrt{2}}} \propto H$$

T Gate Evolution

$$H = \frac{1}{2}\sigma_z$$

$$U(t) = e^{-iHt} = e^{-i\frac{t}{2}\sigma_z} = \begin{pmatrix} e^{-it/2} & 0 \\ 0 & e^{it/2} \end{pmatrix}$$

Removing $e^{-it/2}$

$$U(t) \propto \begin{pmatrix} 1 & 0 \\ 0 & e^{it} \end{pmatrix}$$

To generate the T gate we need the relative phase to be $\pi/4$, so we set

$$t = \frac{\pi}{4}$$

T Gate Evolution

Thus,

$$e^{-i\frac{\pi}{8}\sigma_z} \propto T$$

More precisely, one obtains

$$U\left(\frac{\pi}{4}\right) = e^{-i\frac{\pi}{8}\sigma_z} = e^{-i\pi/8}\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

which implements the T gate up to a global phase.

CNOT Gate Evolution

$$\text{CNOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$$

A standard approach:

$$H_{\text{int}} = |1\rangle\langle 1| \otimes u$$

so that the time evolution operator becomes

$$U(t) = e^{-iH_{\text{int}}t} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes e^{-iut}$$

To implement the CNOT gate, choose

$$u = \frac{\pi}{2}X$$

and set the evolution time $t = 1$ (with $g = 1$ in dimensionless units).

CNOT Gate Evolution

Then,

$$U = e^{-i\left(|1\rangle\langle 1| \otimes \frac{\pi}{2} X\right)} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes e^{-i\frac{\pi}{2}X}$$

Since

$$e^{-i\frac{\pi}{2}X} = -iX$$

(up to a phase), we have

$$U \propto |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$$

which is the CNOT gate (ignoring the phase factor).

Our problem is to find the control parameters that yield these quantum gates. We set the overall time

$$T = 1$$

and we discretize the operator as:

$$U(t) = (e^{-iH(\Omega,\Delta,J)\Delta t})^N U(0)$$

where

$$\Delta t = T/N = 1/N$$

$$U(0) = \mathbb{I}$$

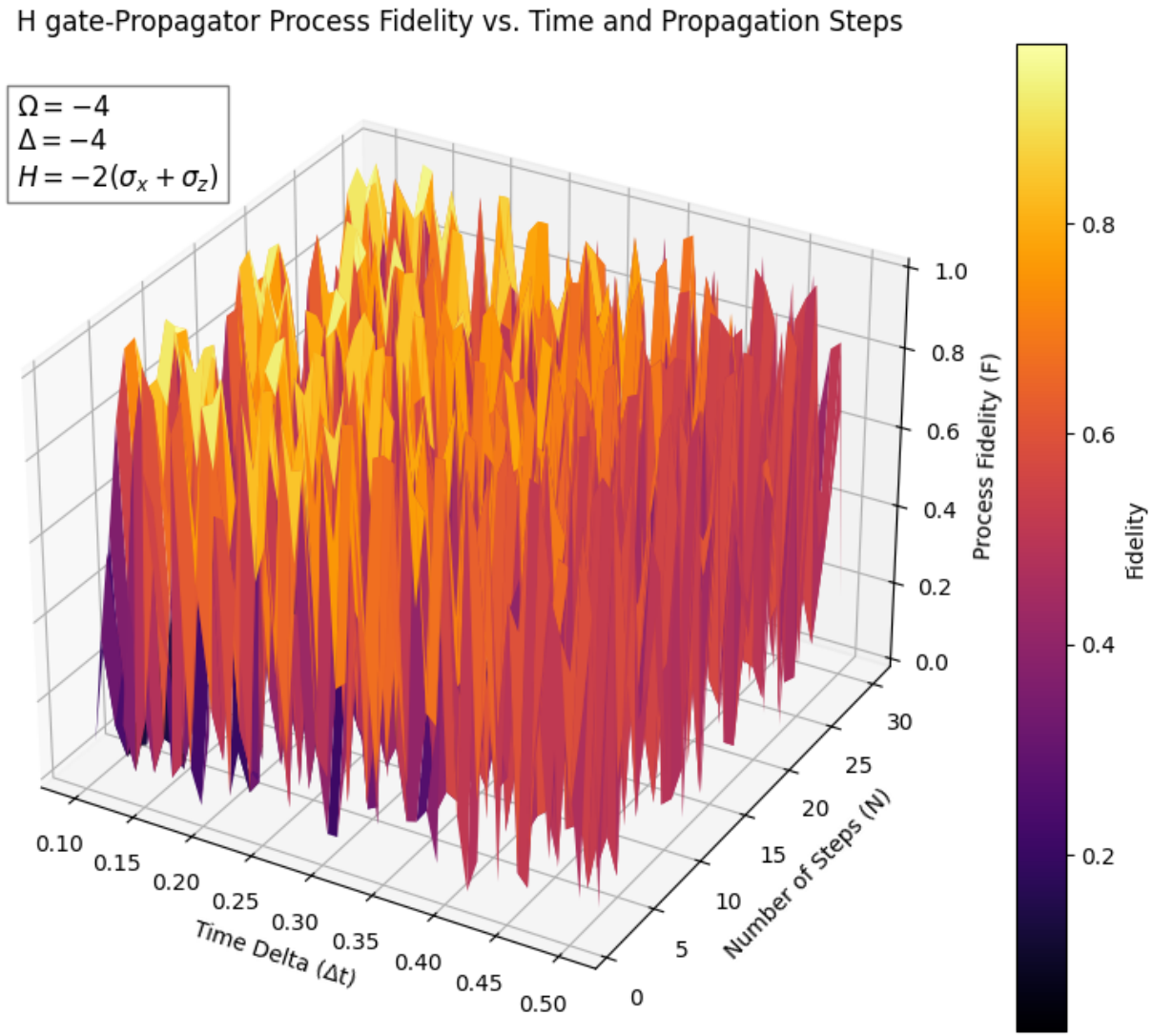
The fidelity of the implemented gate is defined as

$$F(U, T) = \frac{1}{d^2} \left| \mathrm{Tr}(T^\dagger U) \right|^2$$

where T is the target gate, U is the realized gate, d is the dimension of the Hilbert space. A fidelity close to **1** indicates a high gate accuracy.

While the gate fidelity measures how well a specific unitary transformation is achieved, a more robust measure is the average fidelity of the process, which quantifies the precision of the gate implemented across all possible input states. The average process fidelity $\bar{F}$ is given by:

$$\bar{F} = \frac{d + F(U, T)d}{d + 1}$$


H gate-Propagator Process Fidelity vs. Time and Propagation Steps
$\Omega = -4$
$\Delta = -4$
$H = -2(\sigma_x + \sigma_z)$
Time Delta (Δt)
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
Number of Steps (N)
0
5
10
15
20
25
30
Process Fidelity (F)
0.0
0.2
0.4
0.6
0.8
1.0
Fidelity
0.2
0.4
0.6
0.8

Reinforcement Learning

Markov Decision Process

The optimization of quantum gates can be formulated as a Markov Decision Process (MDP), where the goal is to maximize the fidelity of a quantum operation.

An MDP is represented as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma)$

The agent's objective is to find an optimal policy π^* that maps states to actions, maximizing the expected cumulative reward

$$J(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t\right]$$

Algorithm General

Require: Initial model parameters θ

for episode = 1, 2, ..., EPISODES **do**
  Reset the quantum environment state
  for $k = 1, 2, \ldots, N$ **do**
    Choose first action to define the pulse
    Evolve the unitary by the pulse and time
    Compute the step reward based on fidelity
    Estimate the next unitary
    Choose next action according to the agent strategy
    Store trajectory $(s_{k-1}, a_{k-1}, s_k, r_k, d_k)$
  end for
  Calculate Discounted Rewards
  Calculate Loss Function
  Update the model parameters θ
end for

Monte Carlo

Monte Carlo (MC) methods estimate action or state values by averaging sampled returns from episodes.

In quantum control, they can be used to optimize pulse sequences for gate implementation:

$$V(s) = \mathbb{E}[G_t \mid S_t = s] \tag{1}$$

where $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k}$ is the return.

Temporal Difference

Temporal Difference (TD) methods combine Monte Carlo sampling with bootstrapping to estimate value functions.

$$V(s) \leftarrow V(s) + \alpha \left[R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right]$$

Temporal Difference Methods

Q-Learning

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right]$$

SARSA

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \right]$$

Deep Q-Network

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s')} \left[\left(R_{t+1} + \gamma \max_a Q(s', a; \theta^-) - Q(s, a; \theta) \right)^2 \right]$$

Double Deep Q-Networks (DDQN) Double Deep Q-Network

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s')}\left[\left(R_{t+1} + \gamma Q(s', \arg\max_a Q(s', a; \theta); \theta^-) - Q(s, a; \theta)\right)^2\right]$$

Double Dueling Deep Q-Network (DDDQN)

$$Q(s,a) = V(s) + A(s,a) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a')$$

Policy Gradient

Policy gradient methods directly optimize the policy $\pi_\theta(a \mid s)$ by maximizing the expected return:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\pi_\theta} \left[\nabla_\theta \log \pi_\theta(a \mid s) G_t \right]$$

REINFORCE:

$$\Delta\theta \propto \nabla_\theta \log \pi_\theta(a \mid s) G_t.$$

REINFORCE with Baseline: To reduce variance, a baseline $b(s)$ is introduced:

$$\Delta\theta \propto \nabla_\theta \log \pi_\theta(a \mid s) \left[G_t - b(s)\right].$$

Policy Gradient Methods

DDPG is designed for continuous action spaces and leverages a deterministic policy:

$$a = \mu_\theta(s)$$

The critic evaluates the Q-value $Q(s, a)$, and both the actor and the critic are updated using neural networks

Policy Gradient Methods

TD3 improves DDPG by addressing its sensitivity to overestimation and instability. Key modifications include:

- Double Critics: Two critic networks are used, and the smaller Q-value is chosen during updates:

$$y = R + \gamma \min_{i=1,2} Q_i(s', \mu(s'))$$

- Delayed Policy Update The actor is updated less frequently than the critics to improve stability.
- Target Policy Smoothing: Adds noise to the target actions to prevent the use of sharp Q-value spikes.

PPO simplifies trust-region policy optimization by clipping the policy ratio:

$$L^{CLIP}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right]$$

where $r_t(\theta)$ is the probability ratio and $\hat{A}_t$ is the advantage estimate.

GRPO is built on top of PPO and is a critic-free model.

Implementation

Implementation

To optimize the gate control process, four reinforcement learning algorithms were implemented:

DDDQN: Discrete - Deterministic

TD3: Continuous - Deterministic

PPO: Both - Stocasthic

GRPO: Both - Stocasthic

Reward Shaping

$$F = \frac{1}{d^2} \left| \mathrm{Tr}(T^\dagger U) \right|^2$$

$$L = \log_{10}(1 - F)$$

$$R = -L \left(1 - \frac{t}{T_{\max}} \right)$$

The episode terminates if:

$$t = T_{\max} | F \geq F_{\text{threshold}}$$

At the final time step, an additional reward scaling factor is introduced:

$$R \leftarrow R \times 5, \quad \text{if } F \geq F_{\text{threshold}}$$

where $F_{\text{threshold}}$ is the predefined fidelity threshold.

State Representation

$$U = \begin{bmatrix} a + ib & c + id \\ e + if & g + ih \end{bmatrix}$$

$$S = \left[a, b, c, d, e, f, g, h\right]$$

Action Space

Hadamard Gate (H) and T Gate

- Control Parameters: Rabi frequency $\Omega_x(t)$, and detuning $\Delta(t)$.
- Actions:
  - Adjust Ω_x for qubit rotations.
  - Control Δ to maintain resonance.
  - Values for a discrete set: $\{-4, 0, 4\}$ for both parameters.

CNOT Gate

- Control Parameters: Coupling strength J_{zx}; target qubit Rabi frequency Ω_x^t and detuning Δ_t; and control qubit Rabi frequency Ω_x^c and detuning Δ_c.
- Actions:
  - Tune J_{zx} for entangling operations.
  - Adjust Ω_x^c, Ω_x^t for qubit rotations.
  - Control Δ_c, Δ_t to maintain resonance.
  - Values for a discrete set: $\{-4, 0, 4\}$ for single-qubit parameters and $\{-4, -2, 2, 4\}$ for the coupling strength.

Unitary Evolution

$$i\hbar \frac{d}{dt} U(t) = \hat{H}(t) U(t)$$

$$U(t + \Delta t) = e^{-i\hat{H}(t)\Delta t} U(t)$$

$$U(0) = \mathbb{I}$$

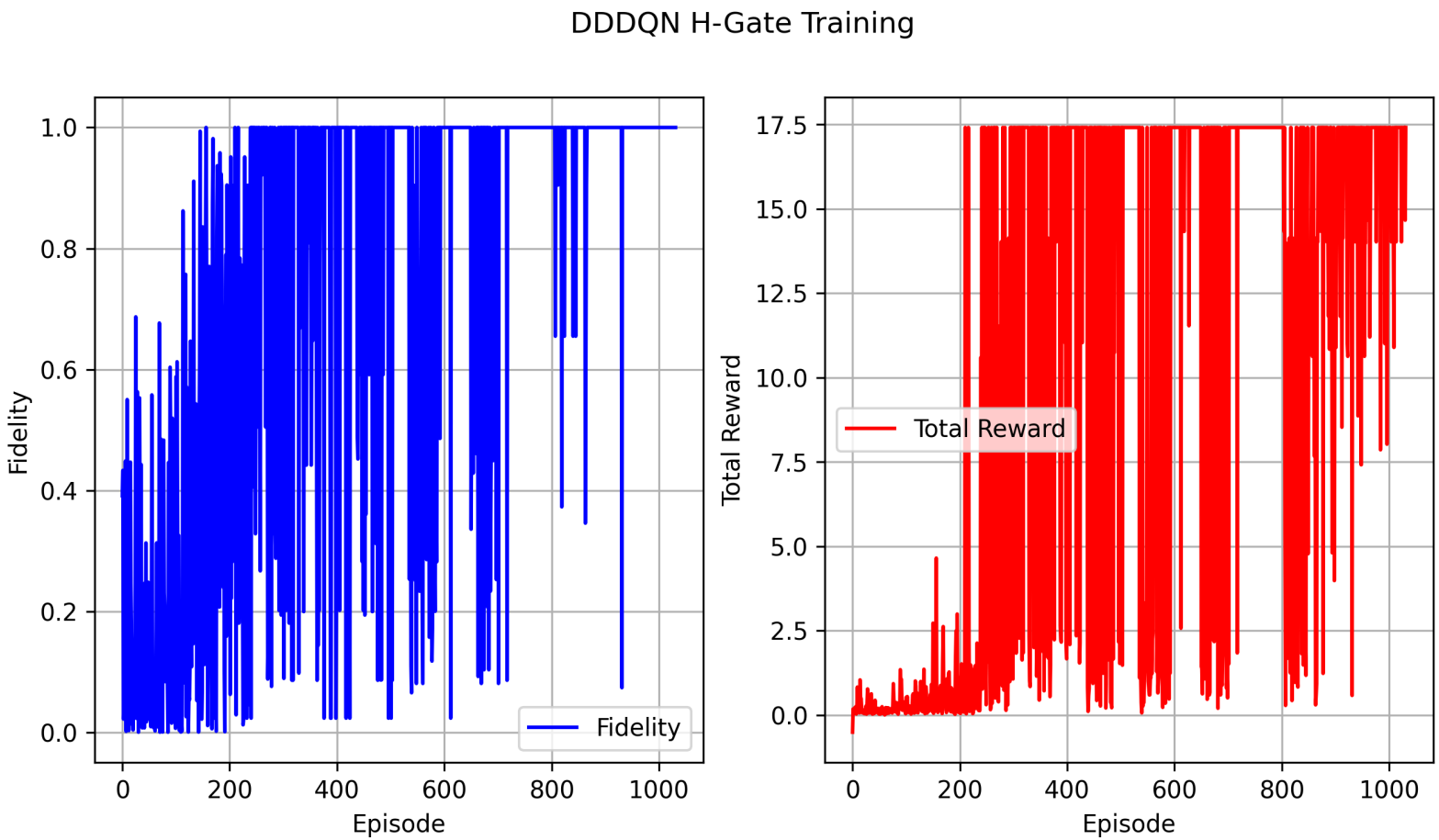
$$U_{n+1} = e^{-i\hat{H}_n \Delta t} U_n$$

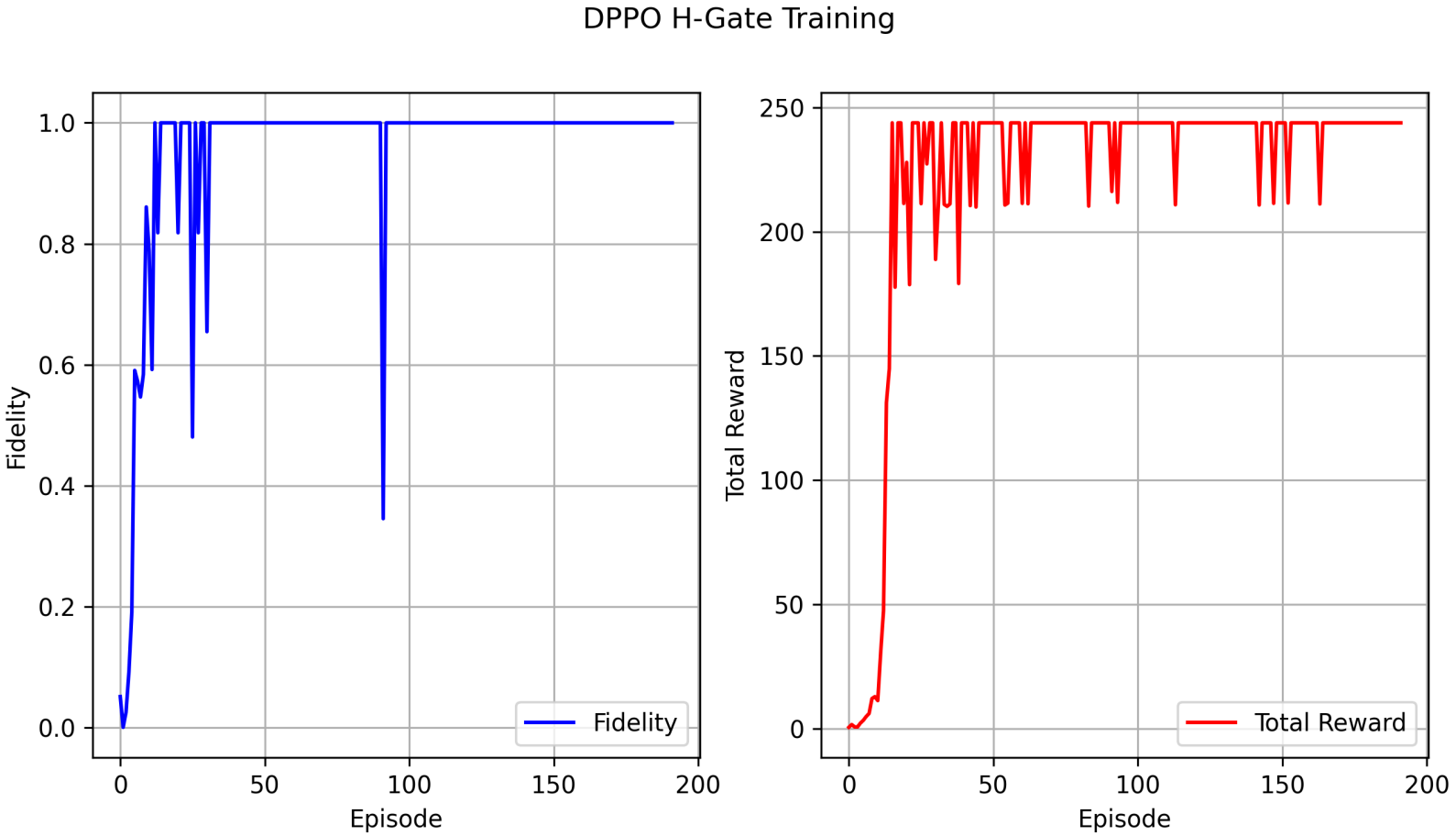
System Configuration

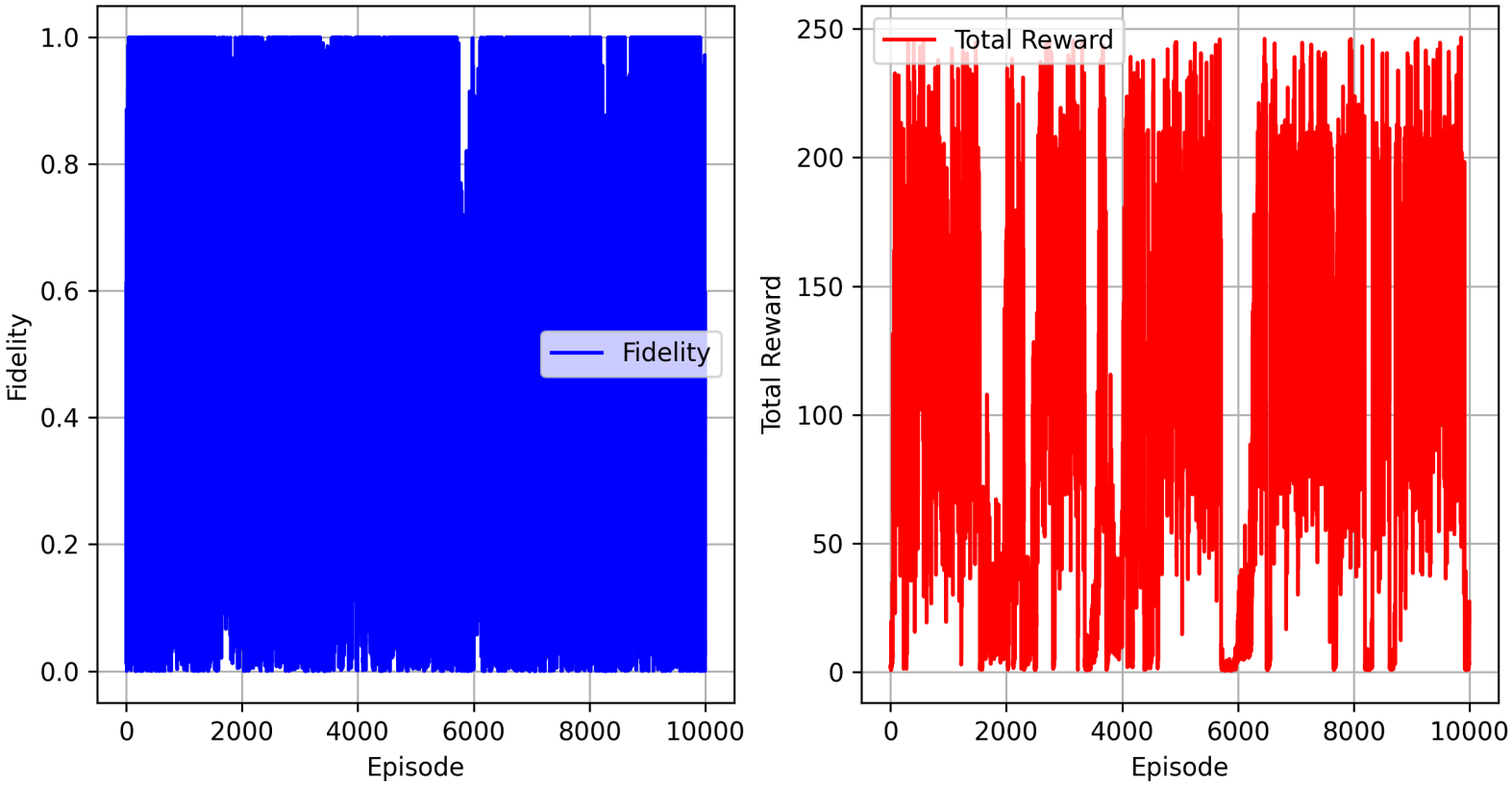
- Operating system: Linux Kernel 6.11.5-1-liquorix-amd64
- Processor: x86_64, 64-bit architecture
- CPU Cores: 6 physical cores, 12 logical threads
- CPU Frequency: 2601.00 MHz
- Total RAM: 31.00 GB
- Python Version: 3.12.7
- PyTorch Version 2.6.0+cu124
- CUDA Version: 12.6
- cuDNN Version: 9.0.1
- NVIDIA Compiler (nvcc): CUDA 12.6
- GPU Model: NVIDIA GeForce GTX 1650 Ti
- Driver Version: 560.35.03
- GPU Memory: 4 GB
- CUDA Cores: GTX 1650 Ti features 1024 CUDA cores.

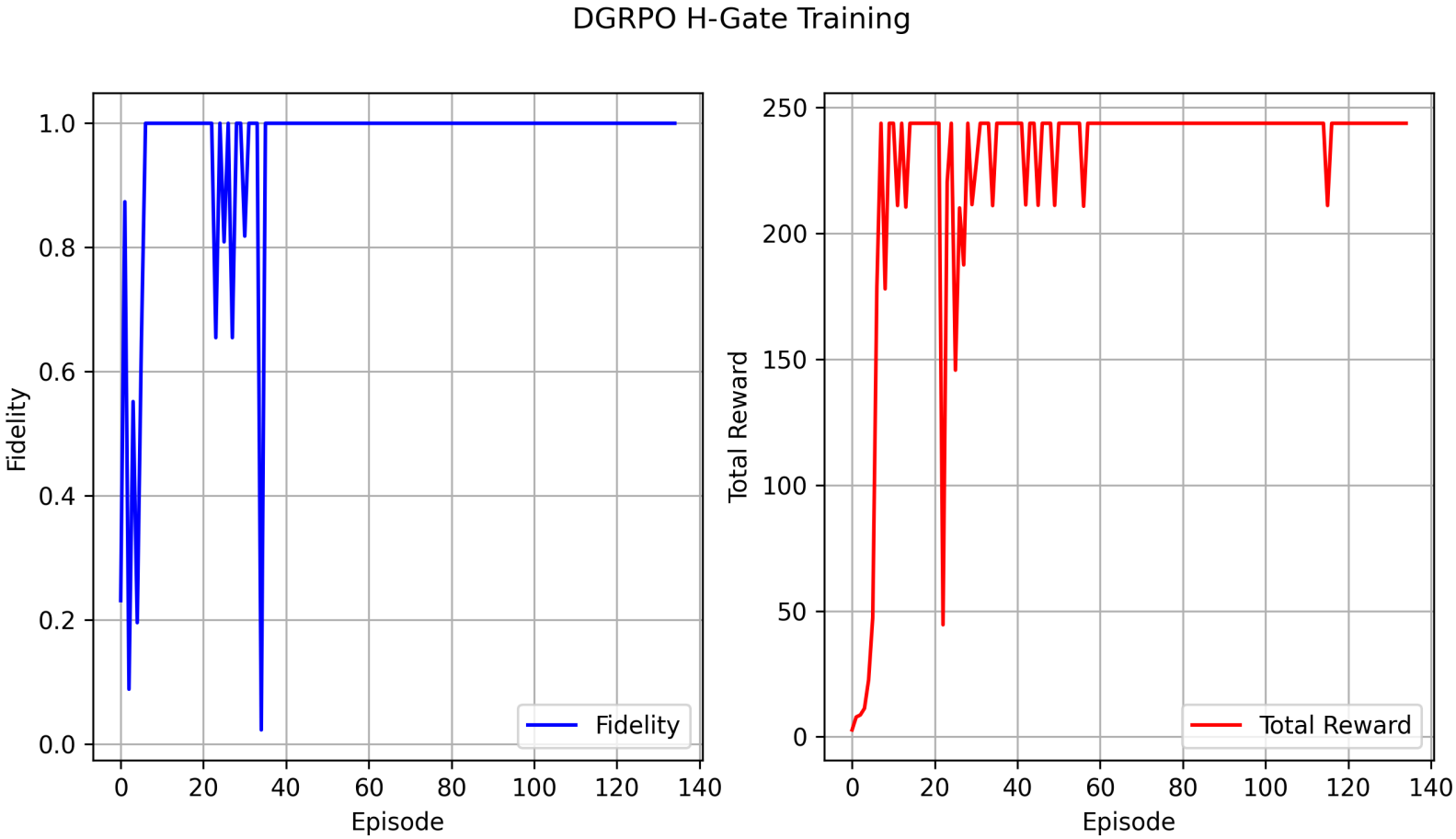
Results

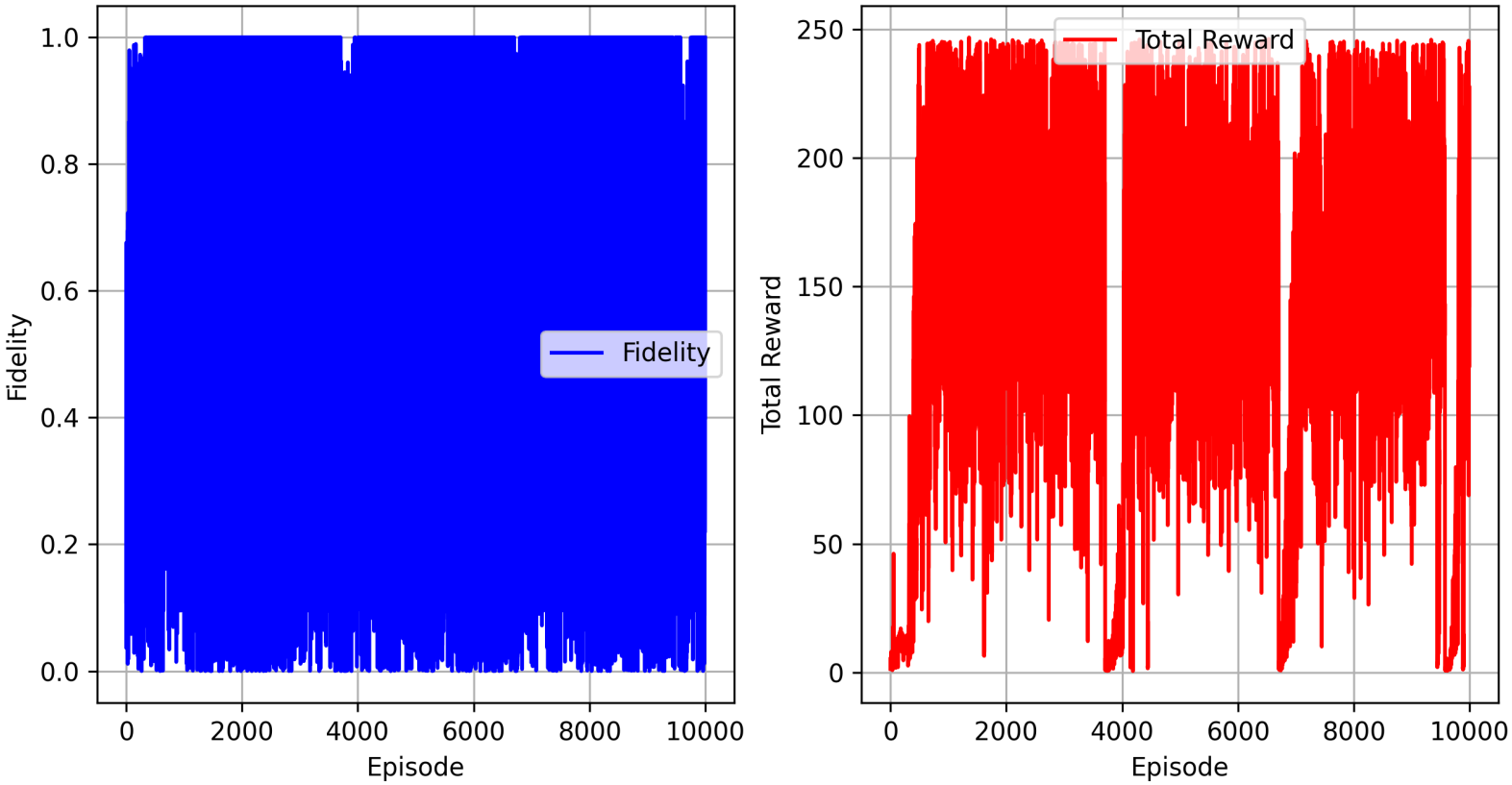
H Gate Training

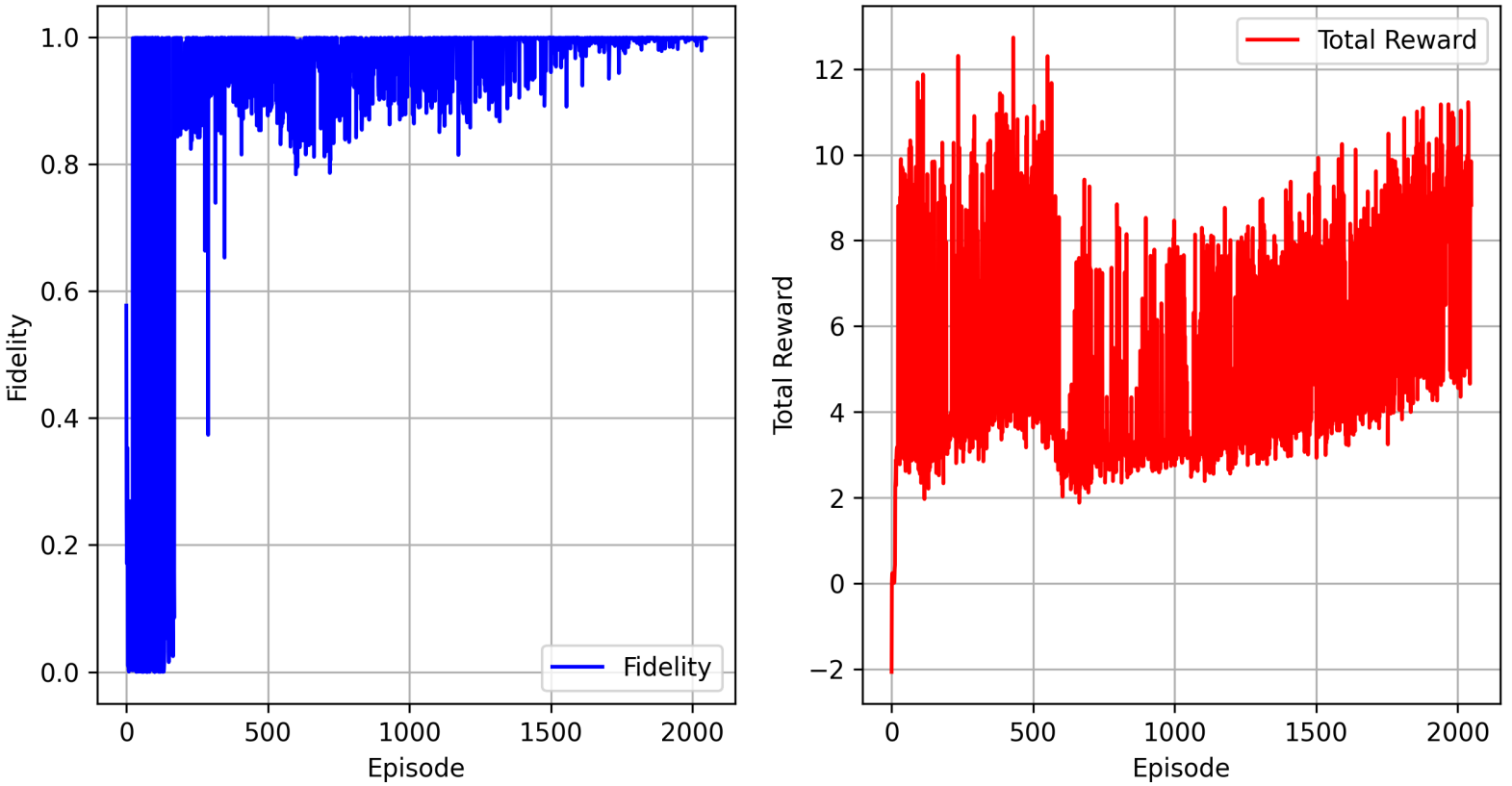



DPPO H-Gate Training
Fidelity
Episode
Fidelity
Total Reward
Episode
Total Reward


CPPO H-Gate Training
Fidelity
Episode
Fidelity
Total Reward
Episode
Total Reward

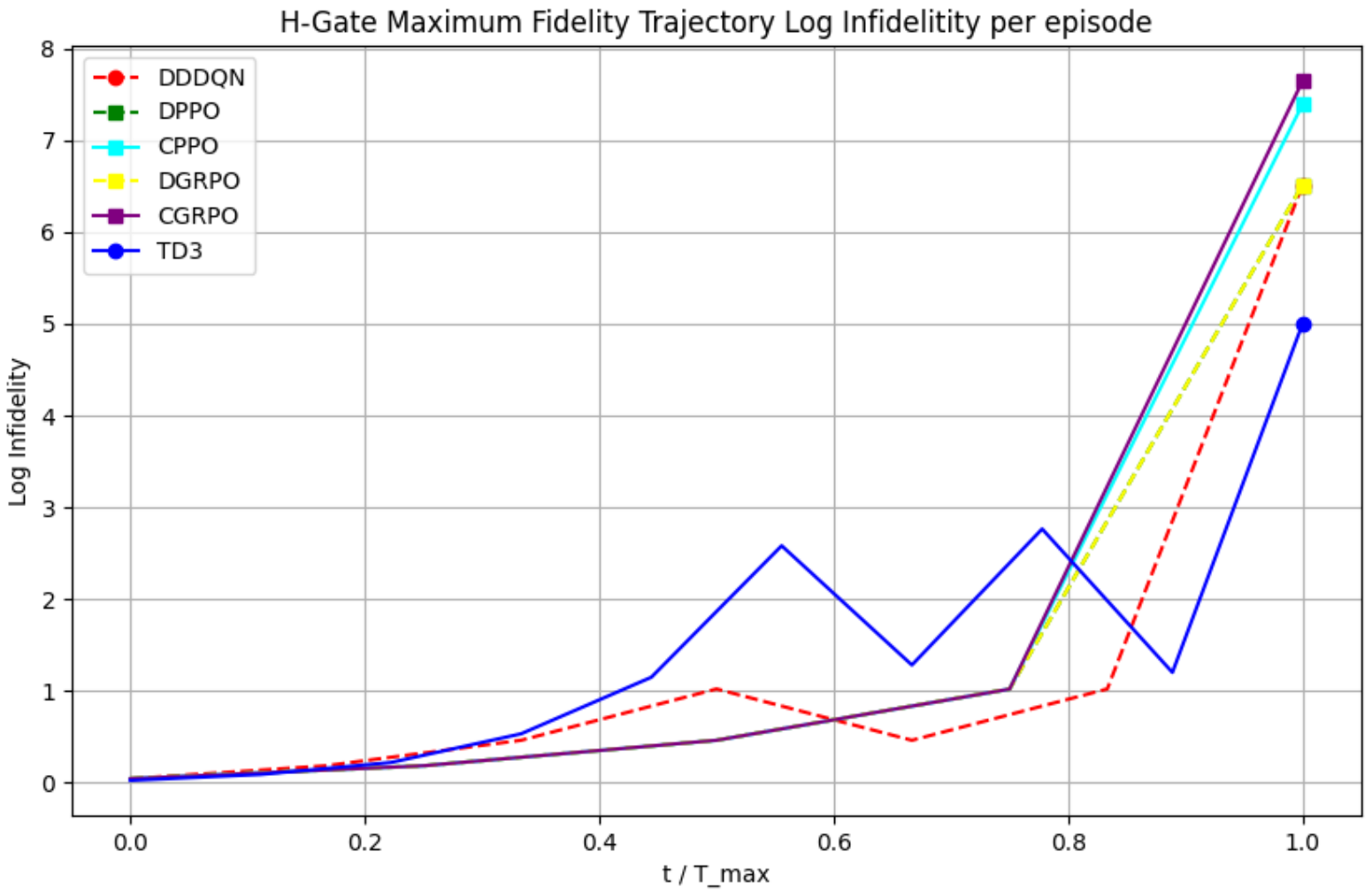

DGRPO H-Gate Training
Fidelity
Episode
Fidelity
Total Reward
Episode
Total Reward


CGRPO H-Gate Training
Fidelity
Episode
Fidelity
Total Reward
Episode
Total Reward

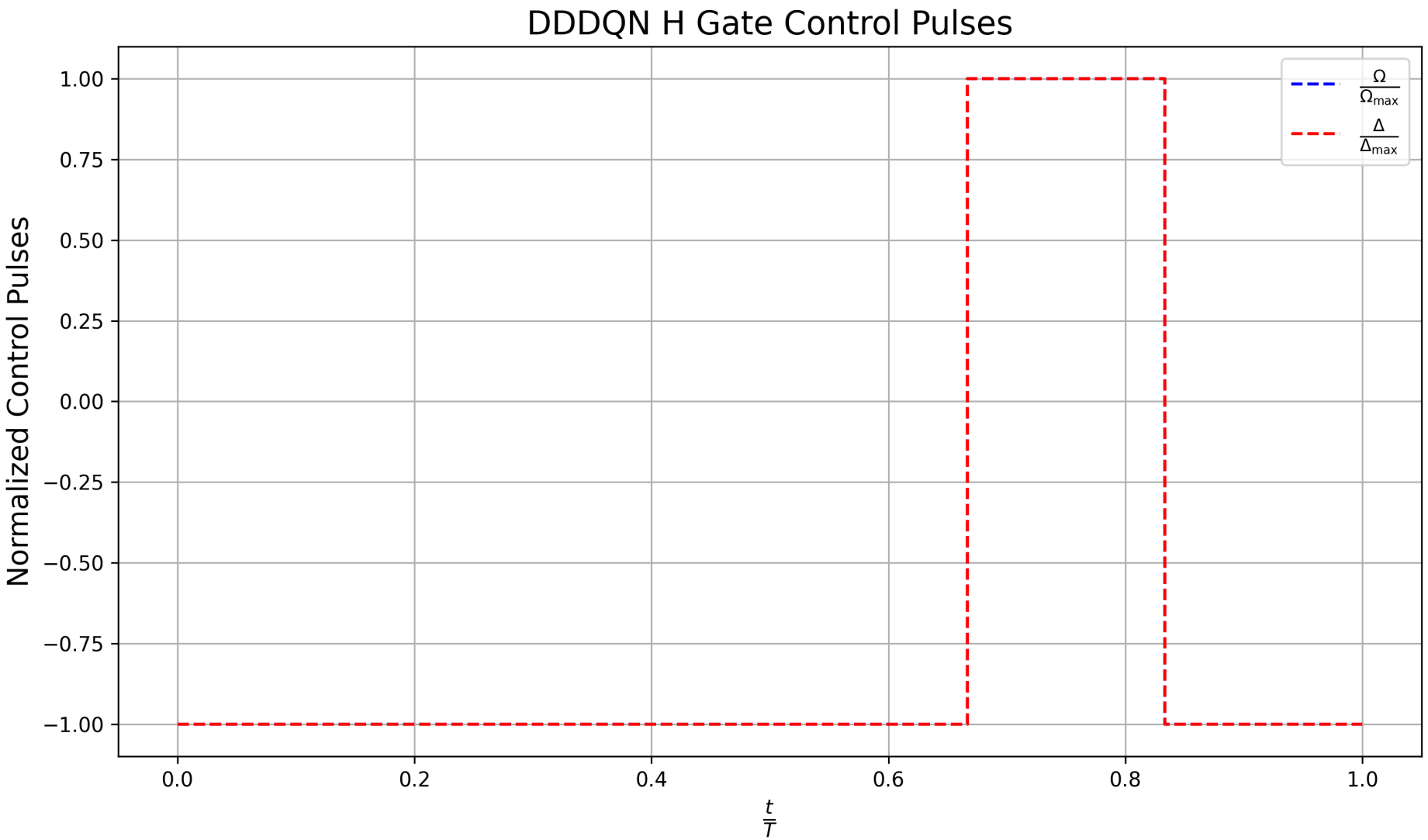

TD3 H-Gate Training
Fidelity
Episode
Fidelity
Total Reward
Episode
Total Reward

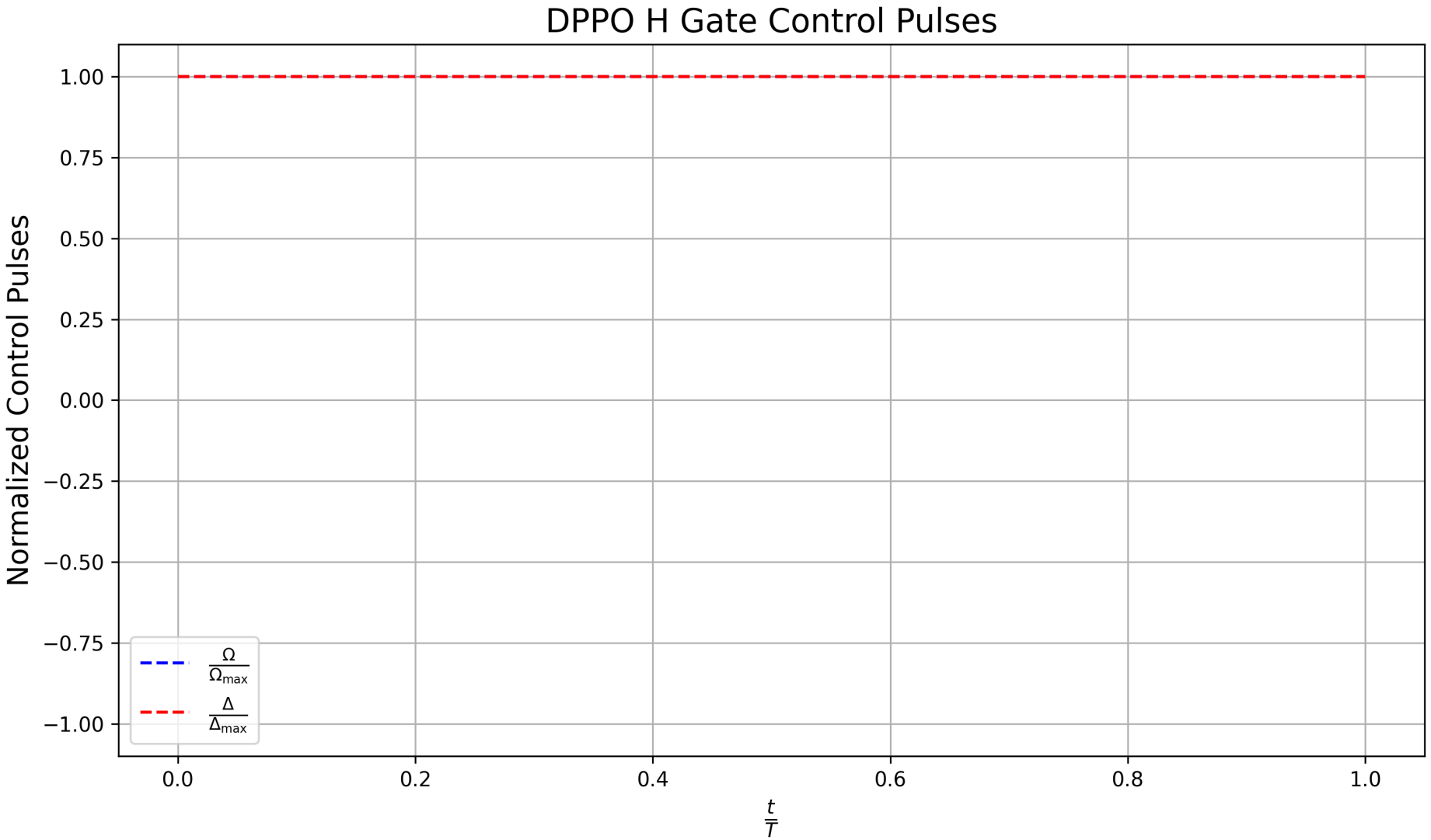
H Gate Performance Metrics

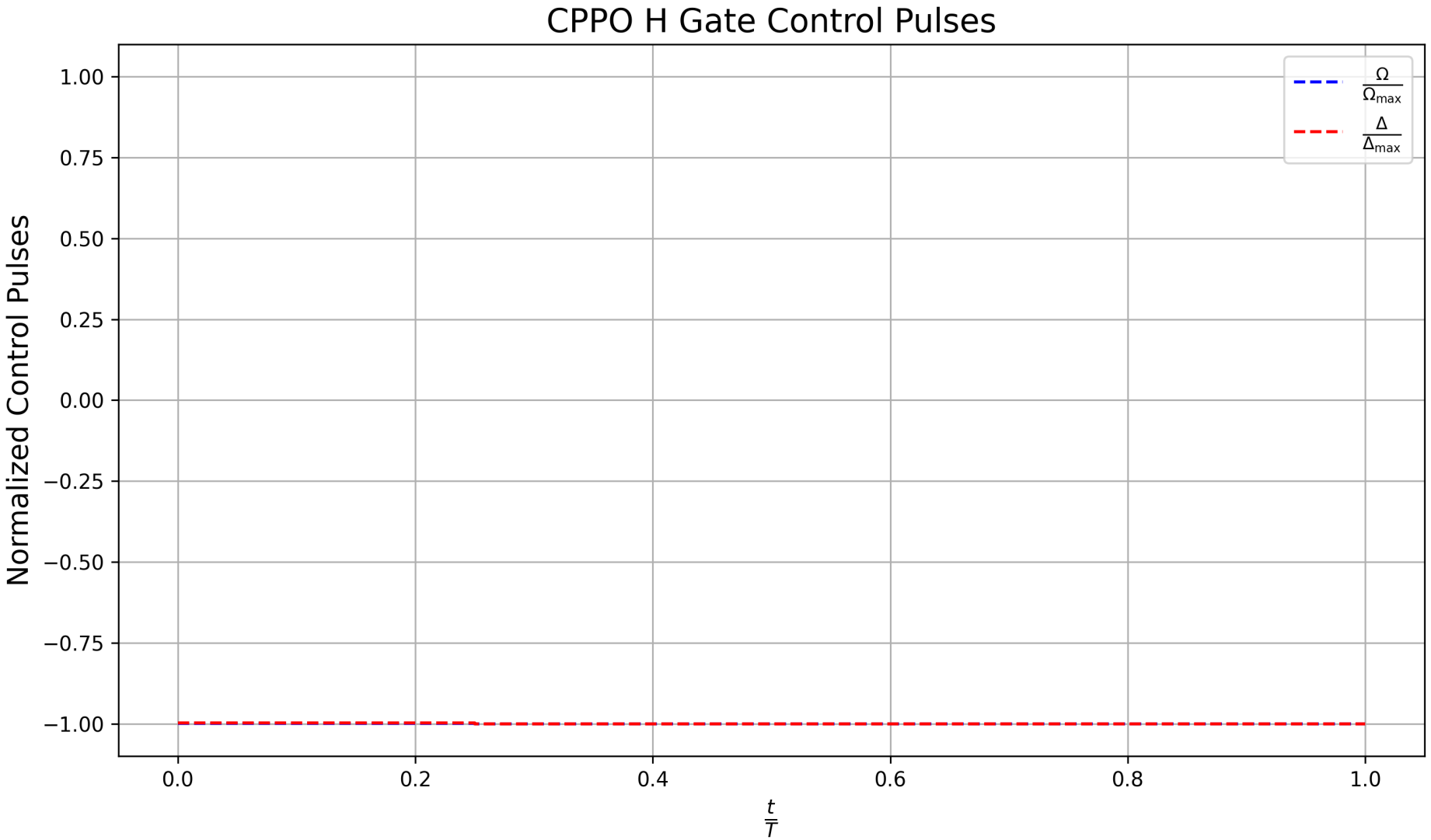
	Agent	Max Fidelity	Max Log Infidelity	Avg Fidelity	Time Steps
5	TD3	0.999990	5.005089	0.999993	10
1	DPPO	1.000000	6.516003	1.000000	5
3	DGRPO	1.000000	6.516003	1.000000	5
0	DDDQN	1.000000	6.516003	1.000000	7
2	CPPO	1.000000	7.395267	1.000000	5
4	CGRPO	1.000000	7.650909	1.000000	5

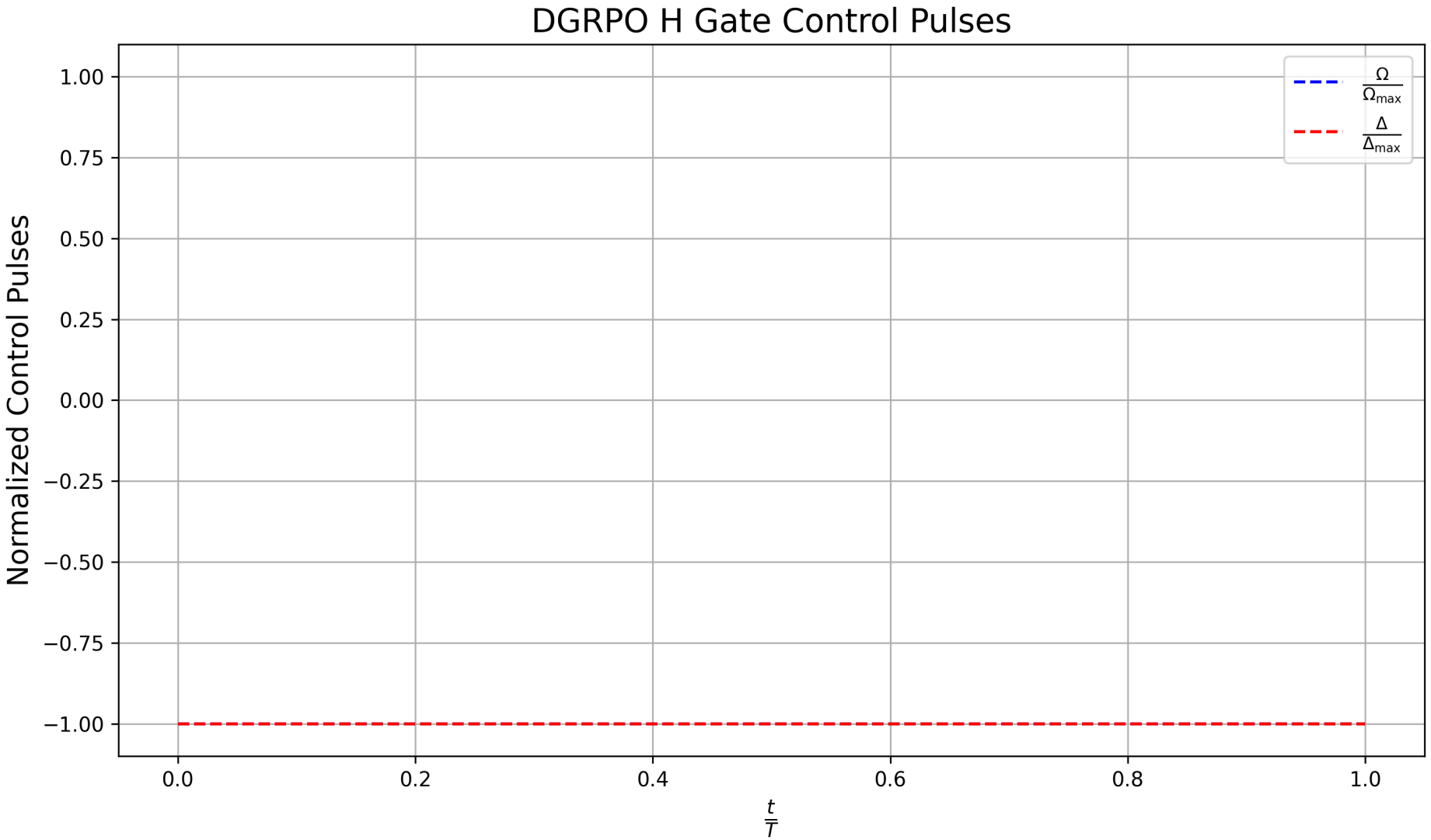

H-Gate Maximum Fidelity Trajectory Log Infidelitity per episode
DDDQN
DPPO
CPPO
DGRPO
CGRPO
TD3
Log Infidelity
t / T_max
0
1
2
3
4
5
6
7
8
0.0
0.2
0.4
0.6
0.8
1.0

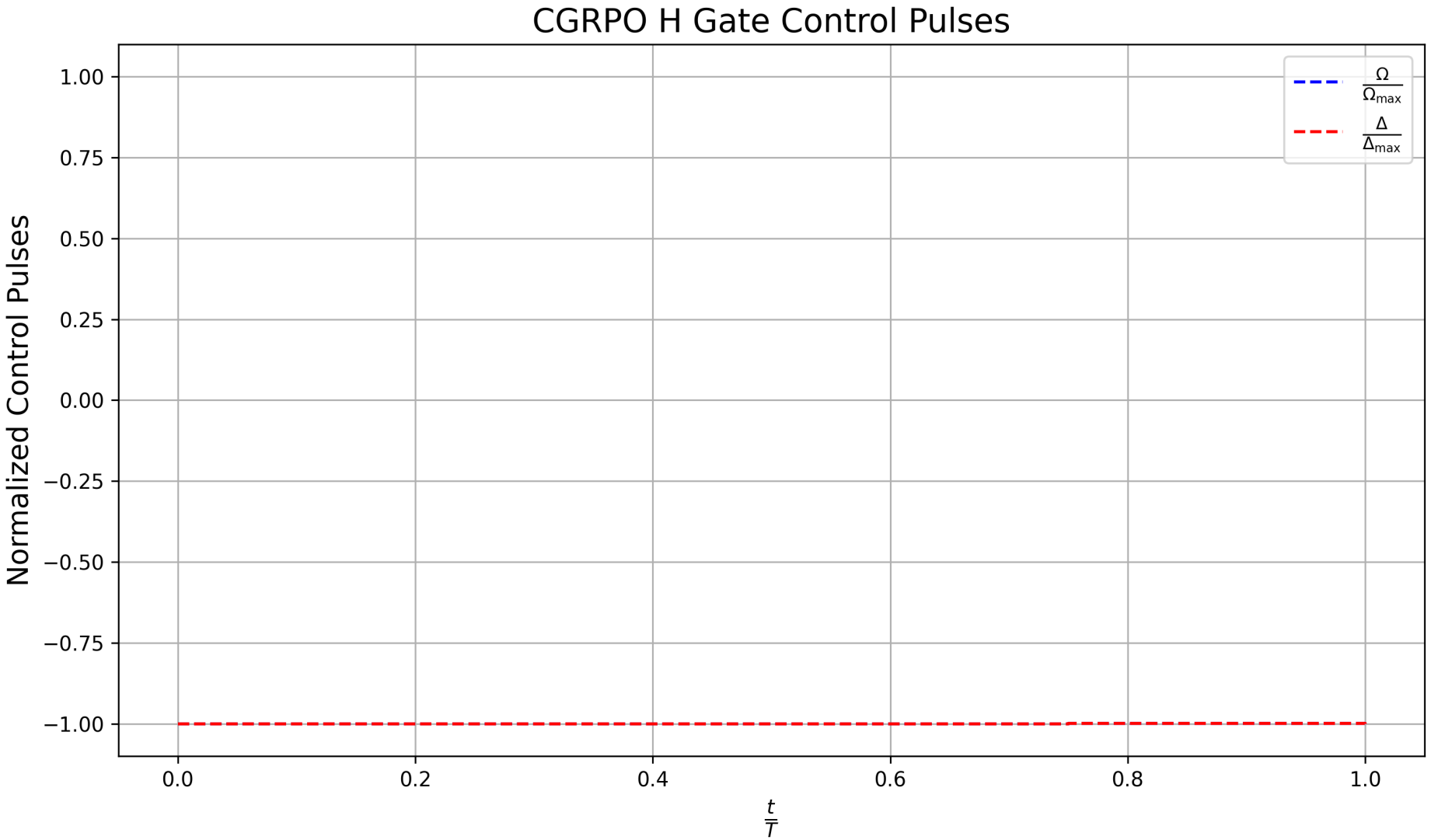
H Gate Control Pulses

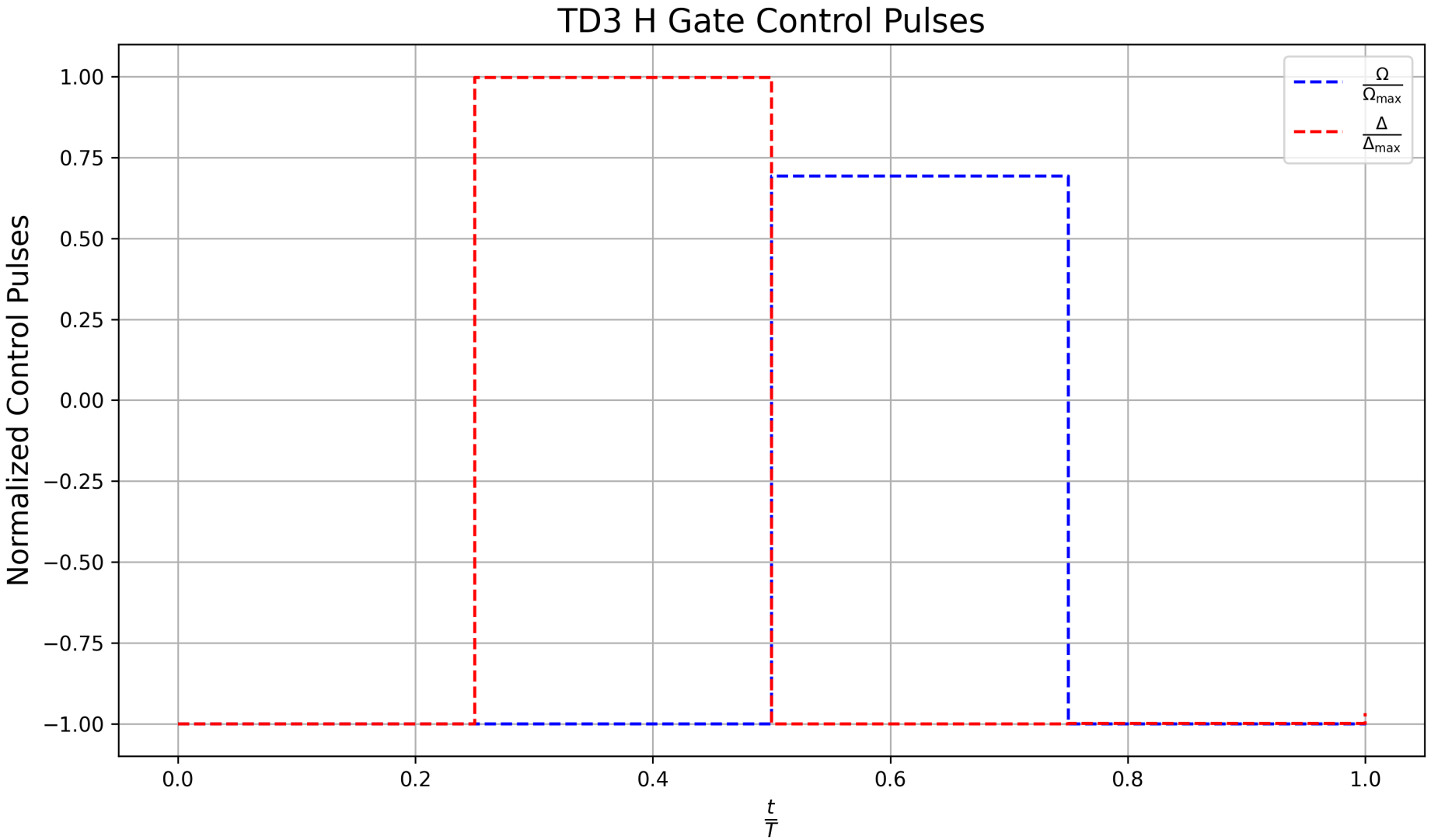



DPPO H Gate Control Pulses
Normalized Control Pulses
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0
$\frac{t}{T}$
$\frac{\Omega}{\Omega_{max}}$
$\frac{\Delta}{\Delta_{max}}$

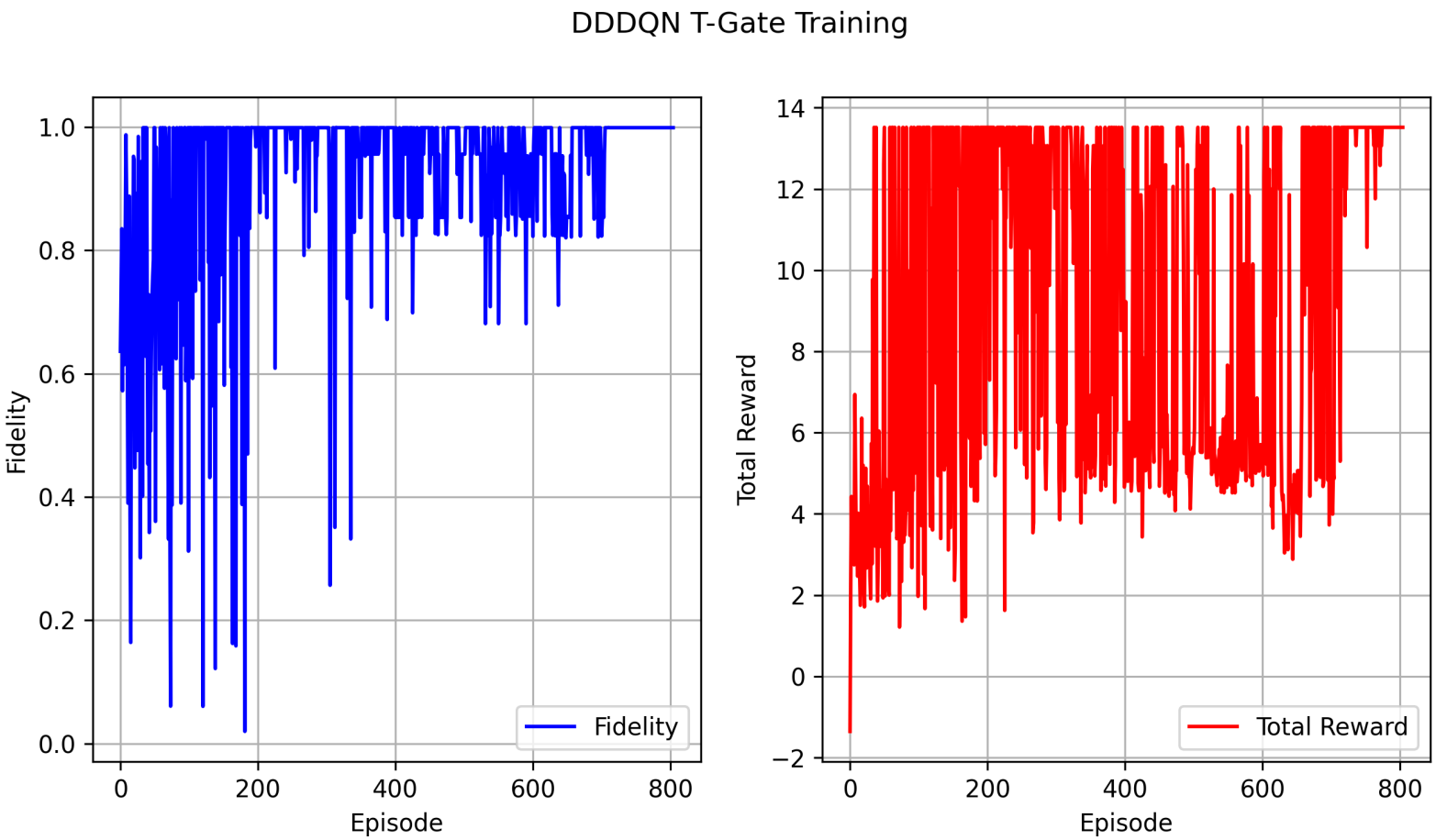

CPPO H Gate Control Pulses
Normalized Control Pulses
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0
$\frac{t}{T}$
$\frac{\Omega}{\Omega_{max}}$
$\frac{\Delta}{\Delta_{max}}$

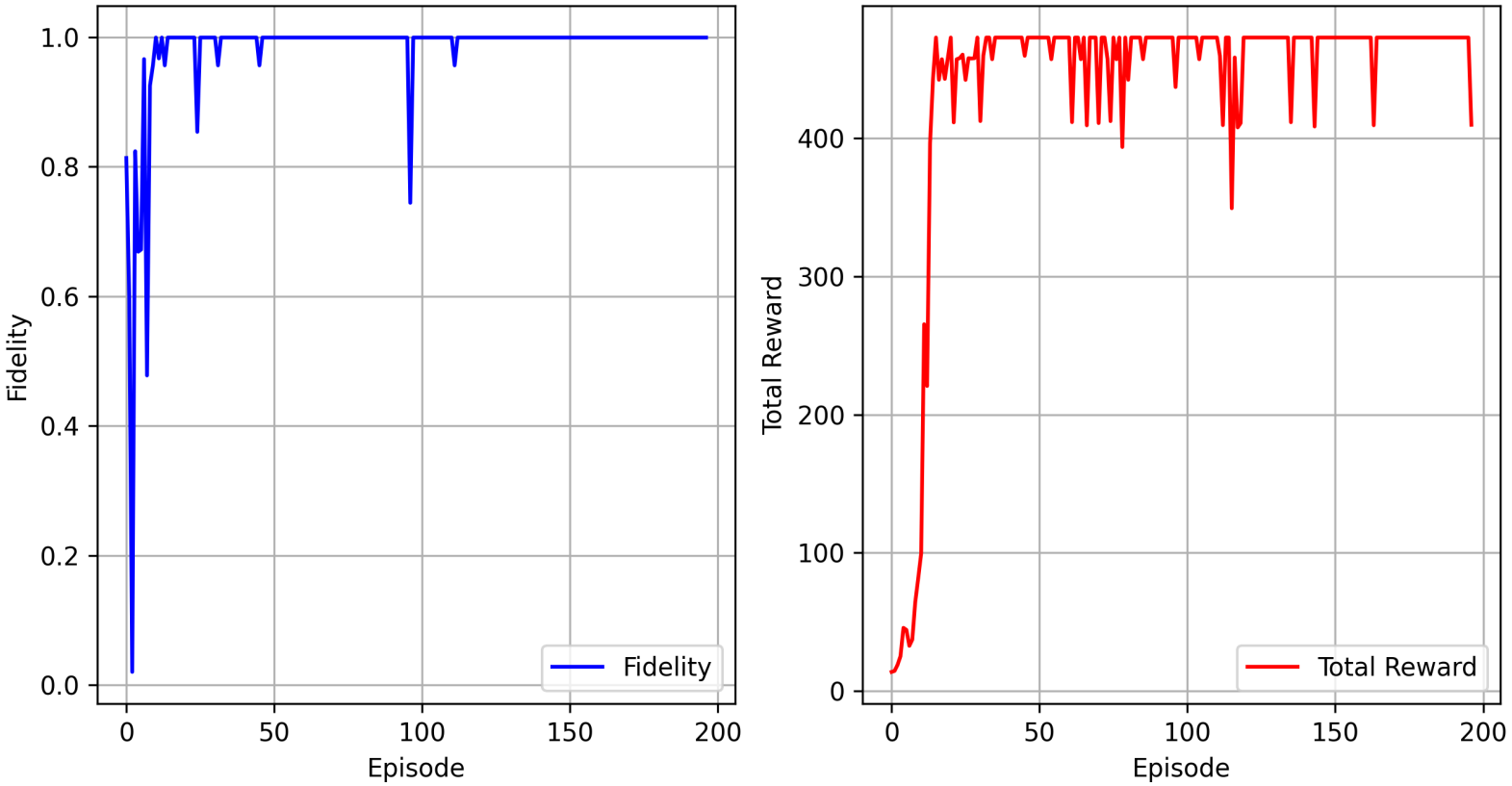

DGRPO H Gate Control Pulses
Normalized Control Pulses
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0
$\frac{t}{T}$
$\frac{\Omega}{\Omega_{max}}$
$\frac{\Delta}{\Delta_{max}}$

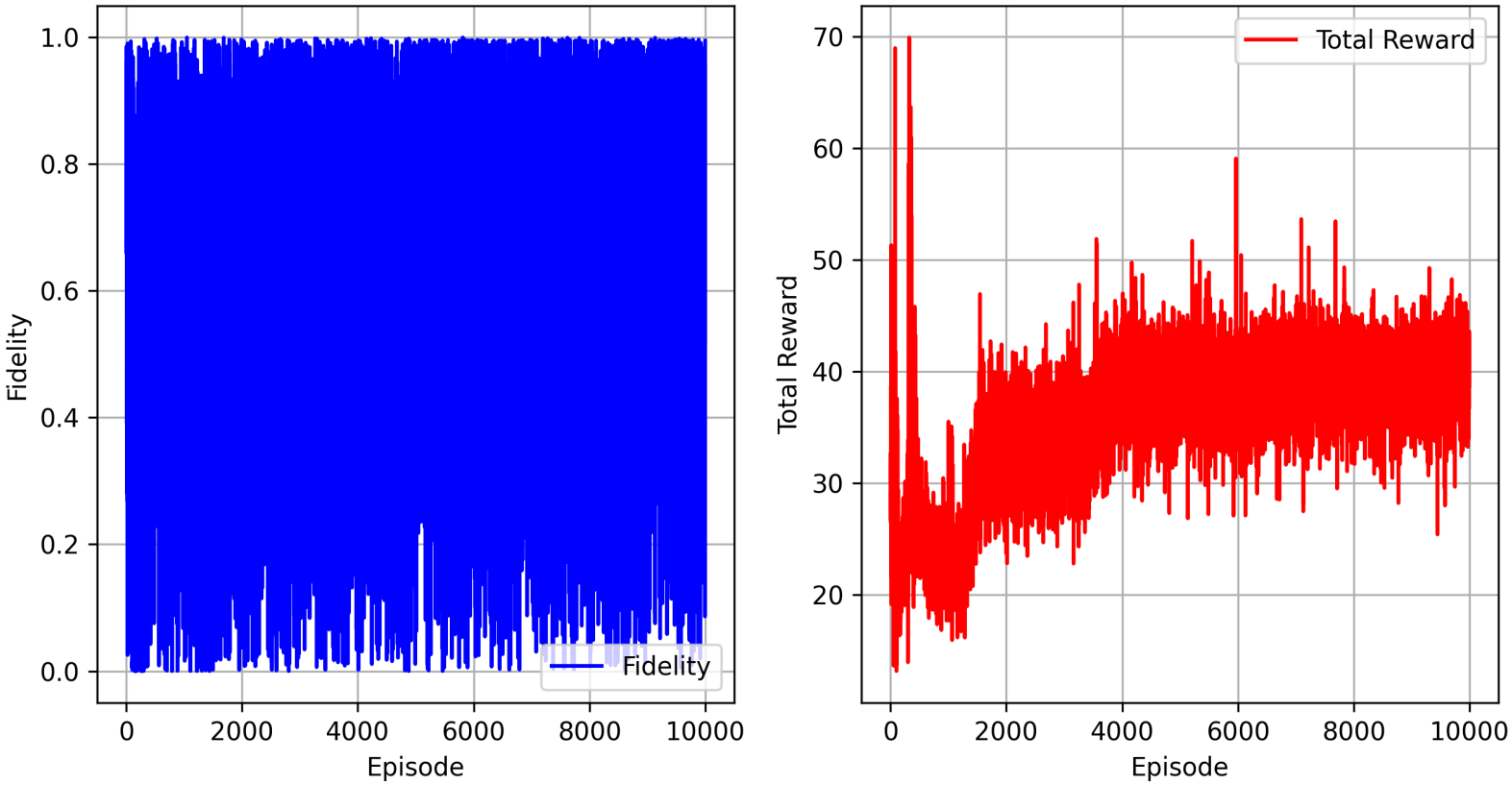

CGRPO H Gate Control Pulses
Normalized Control Pulses
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0
$\frac{t}{T}$
$\frac{\Omega}{\Omega_{max}}$
$\frac{\Delta}{\Delta_{max}}$

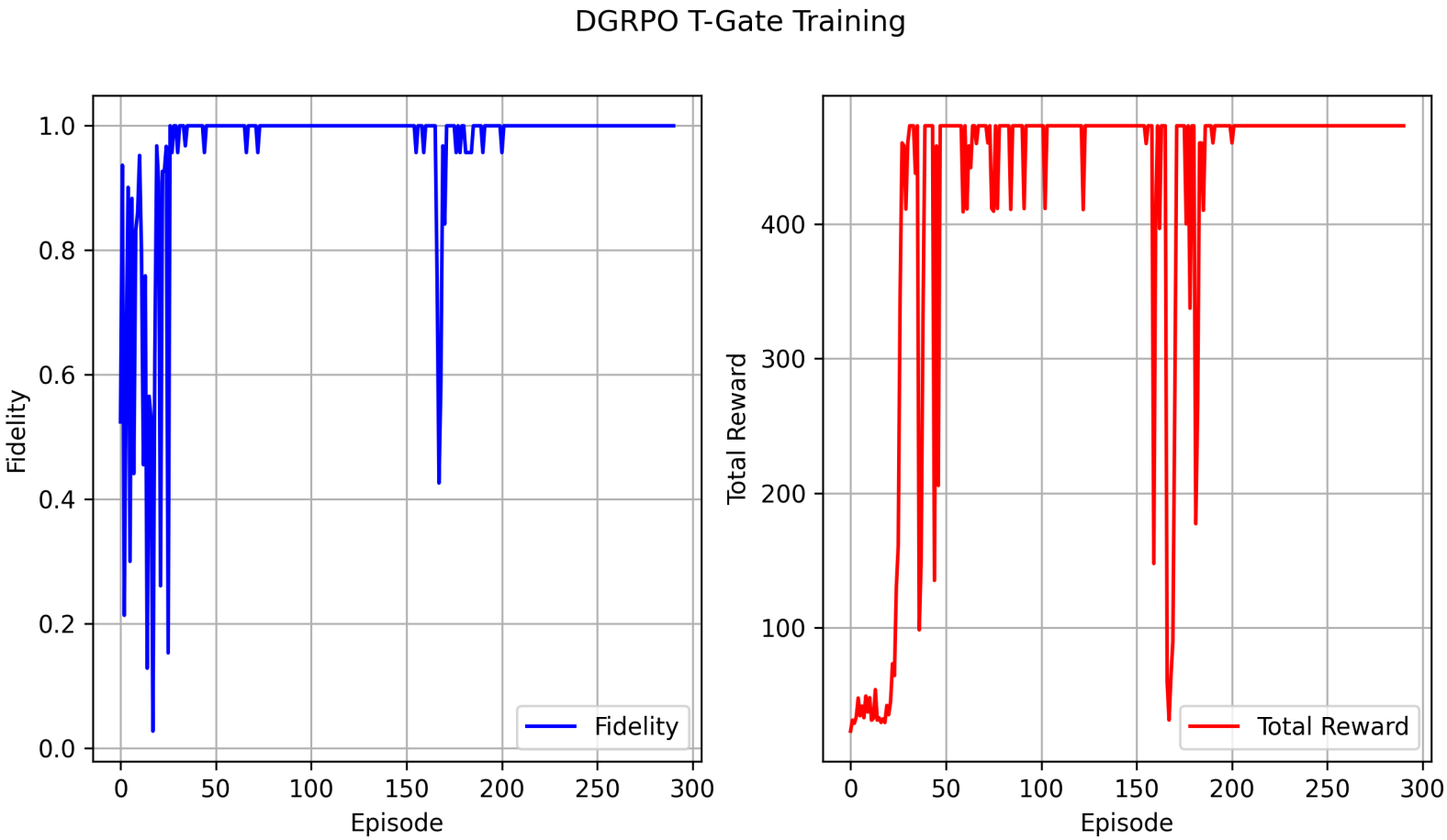

TD3 H Gate Control Pulses
Normalized Control Pulses
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0
$\frac{t}{T}$
$\frac{\Omega}{\Omega_{max}}$
$\frac{\Delta}{\Delta_{max}}$

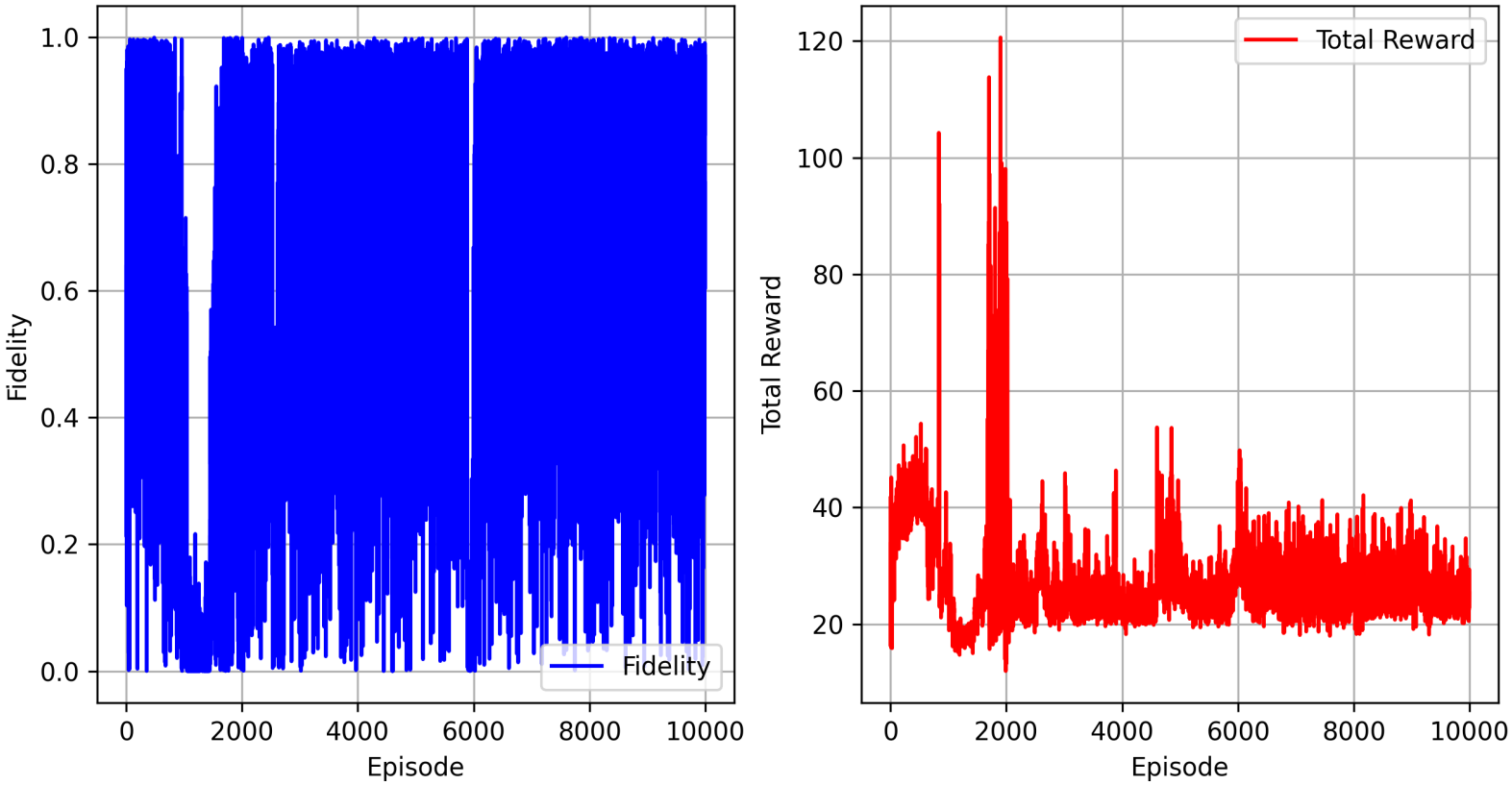
T Gate Training

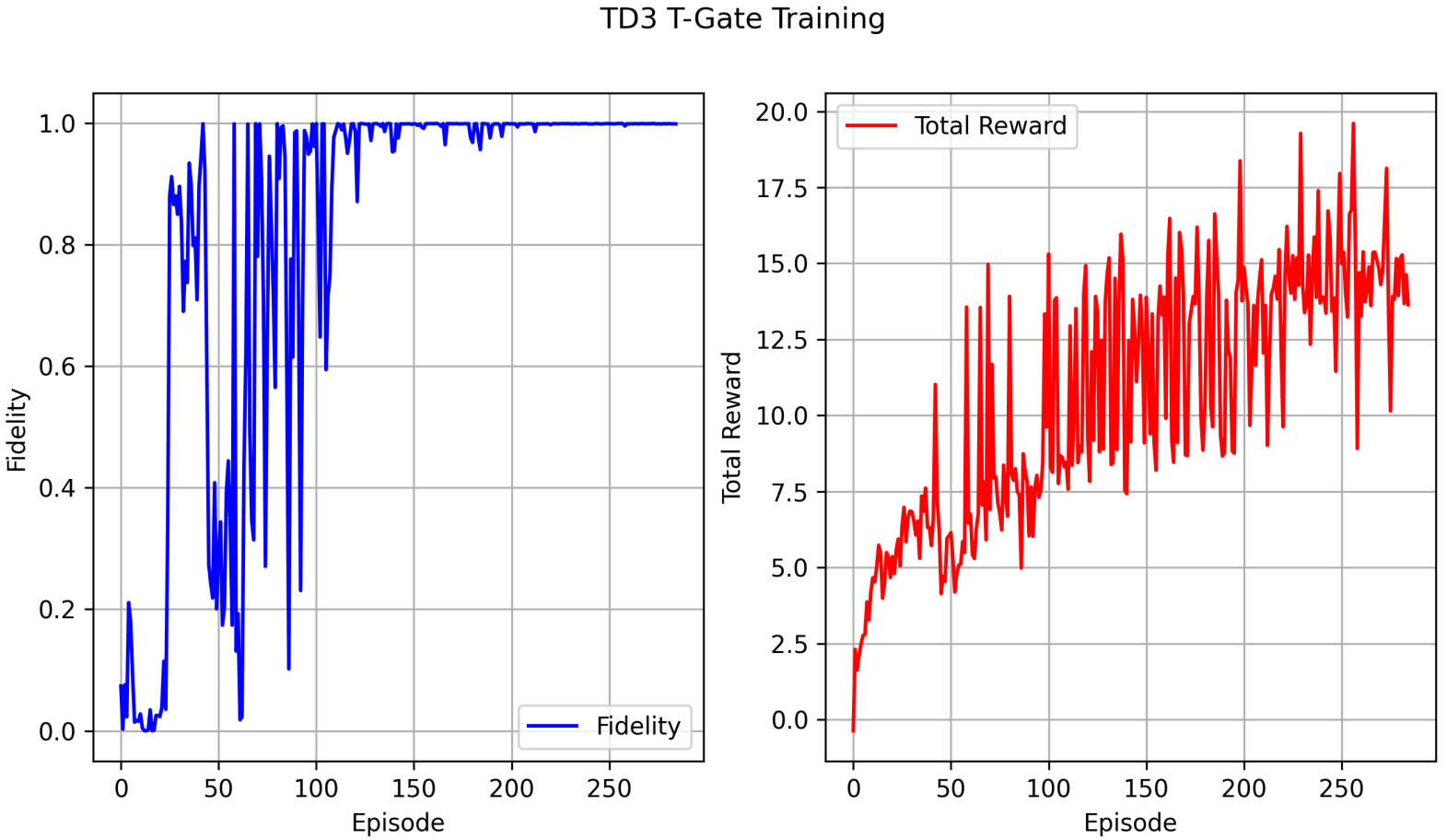



DPPO T-Gate Training
Fidelity
Episode
Fidelity
Total Reward
Episode
Total Reward


CPPO T-Gate Training
Fidelity
Episode
Fidelity
Total Reward
Episode
Total Reward

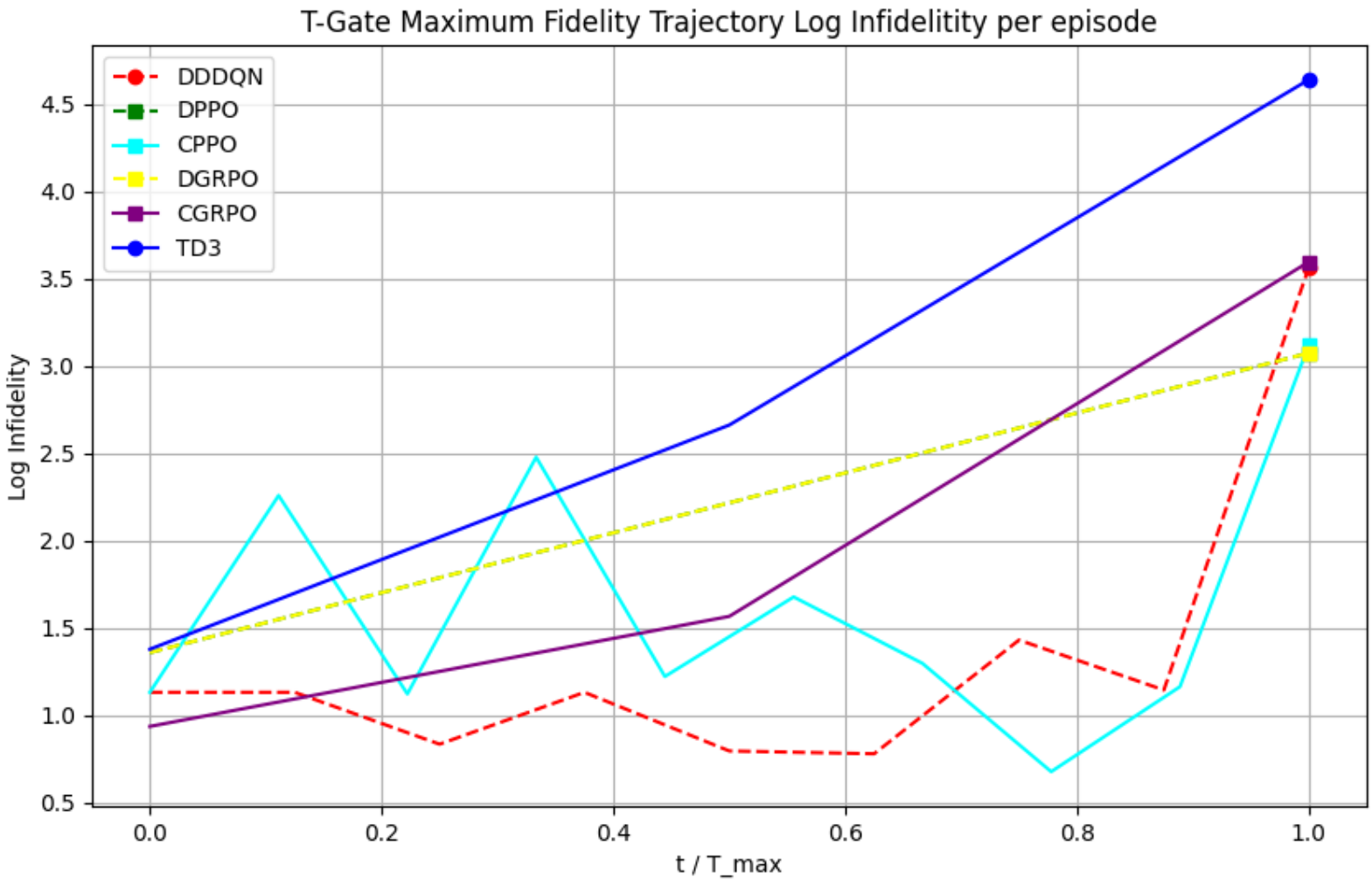

DGRPO T-Gate Training
Fidelity
Episode
Fidelity
Total Reward
Episode
Total Reward


CGRPO T-Gate Training
Fidelity
Episode
Fidelity
Total Reward
Episode
Total Reward

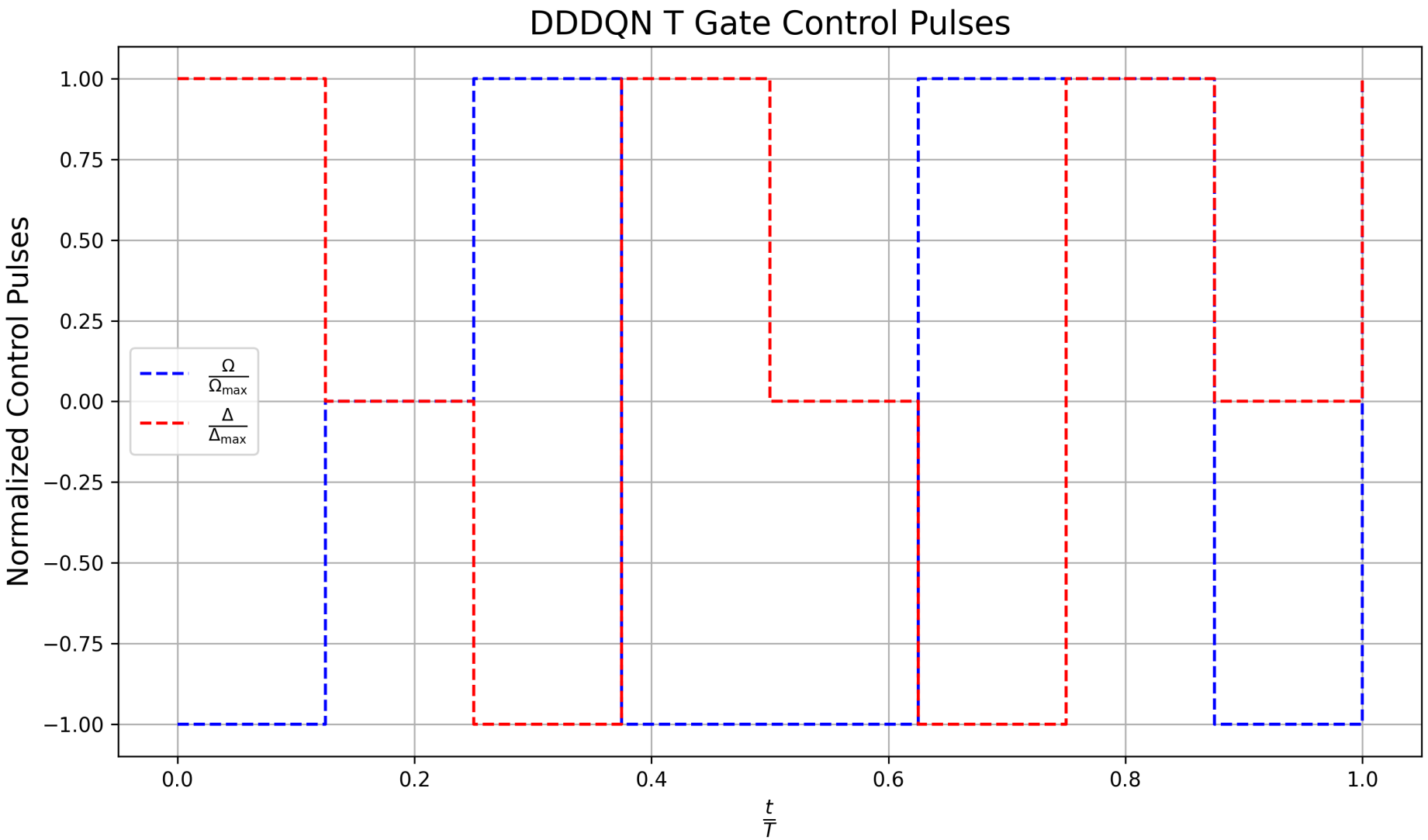

TD3 T-Gate Training
Fidelity
Episode
Fidelity
Total Reward
Episode
Total Reward

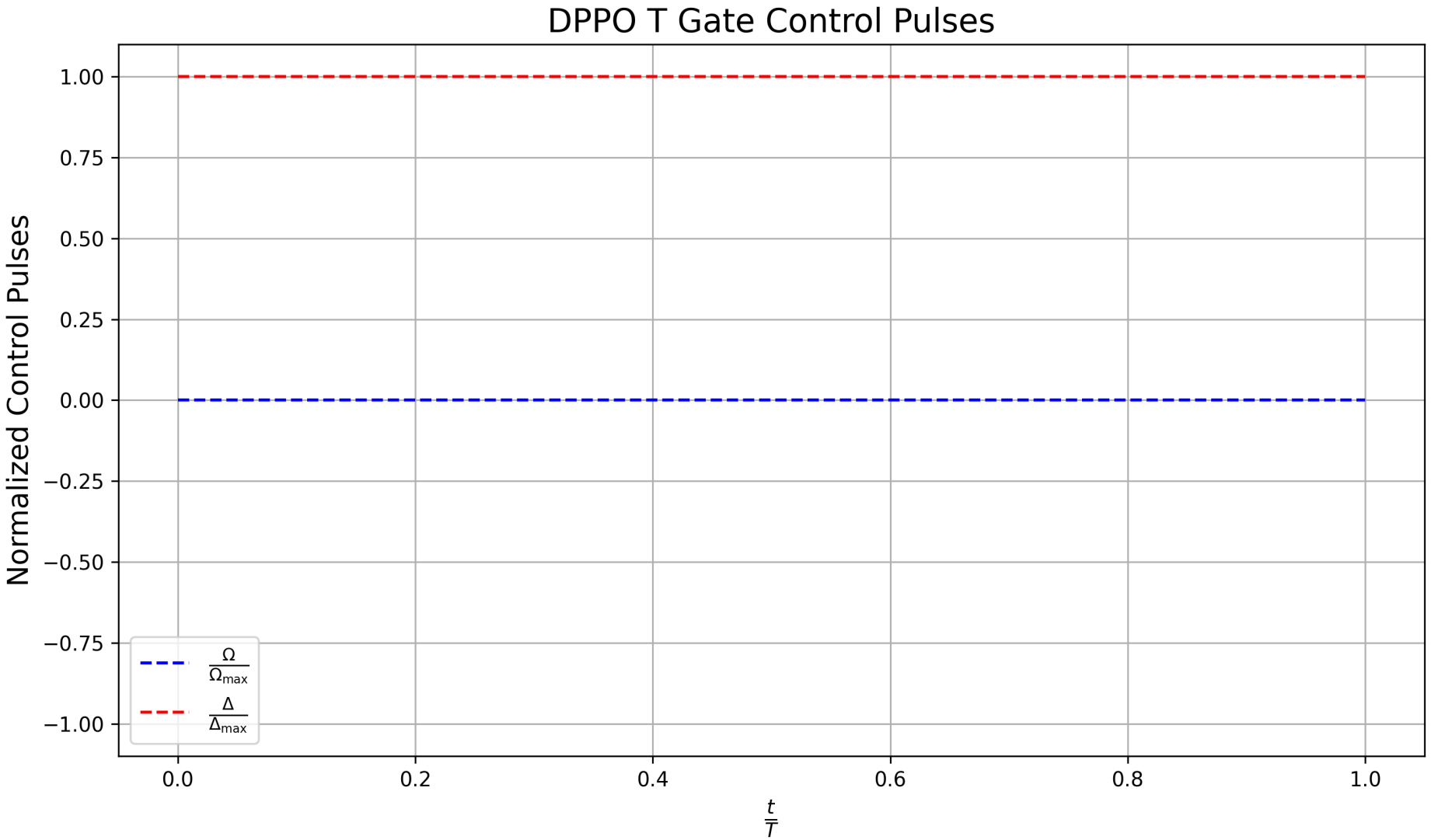
T Gate Performance Metrics

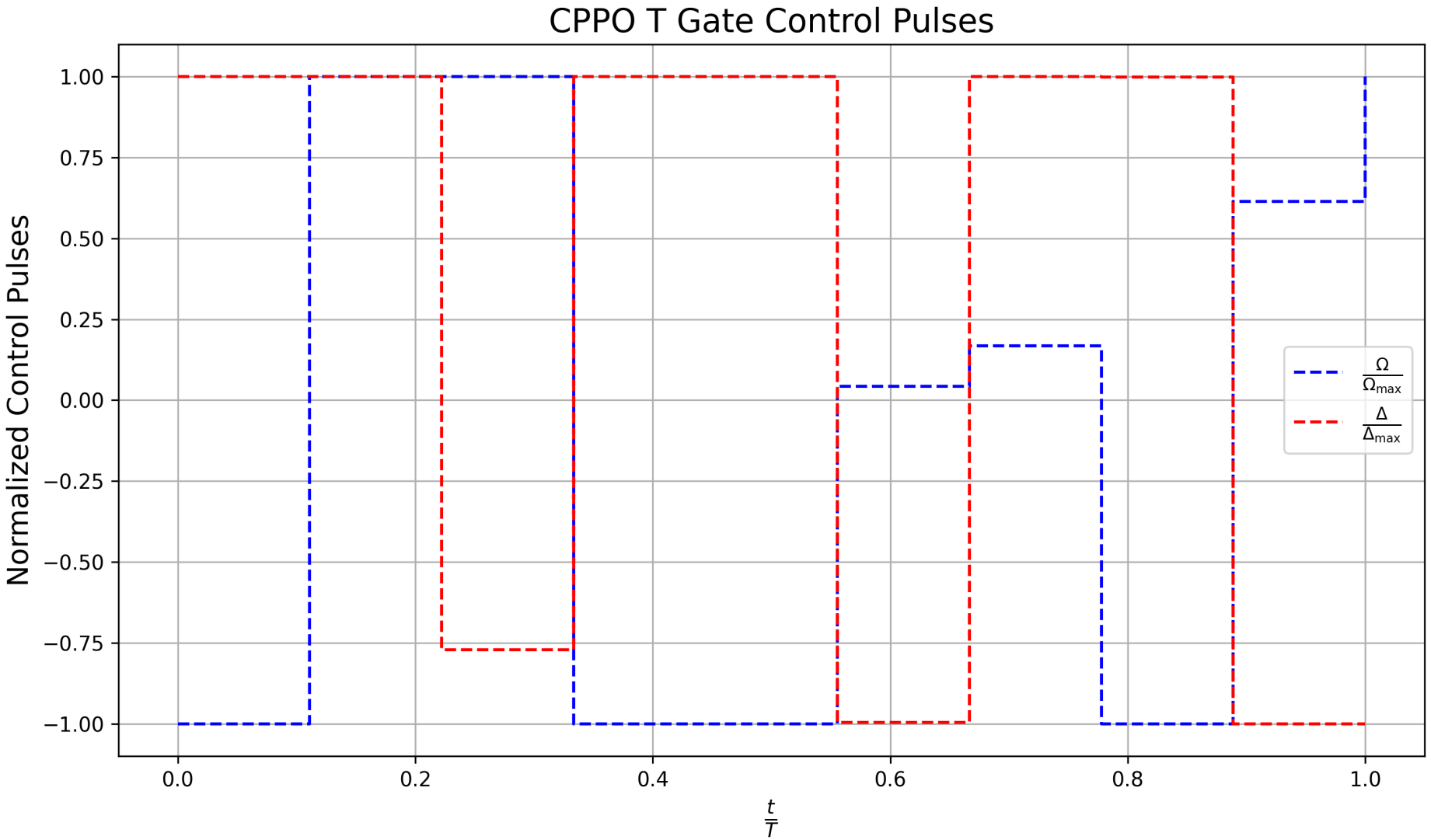
	Agent	Max Fidelity	Max Log Infidelity	Avg Fidelity	Time Steps
1	PPO-D	0.999156	3.073450	0.999437	2
3	GRPO-D	0.999156	3.073450	0.999437	2
2	PPO-C	0.999240	3.119451	0.999494	10
0	DDDQN	0.999726	3.562547	0.999817	9
4	GRPO-C	0.999746	3.594970	0.999831	3
5	TD3	0.999977	4.638075	0.999985	3

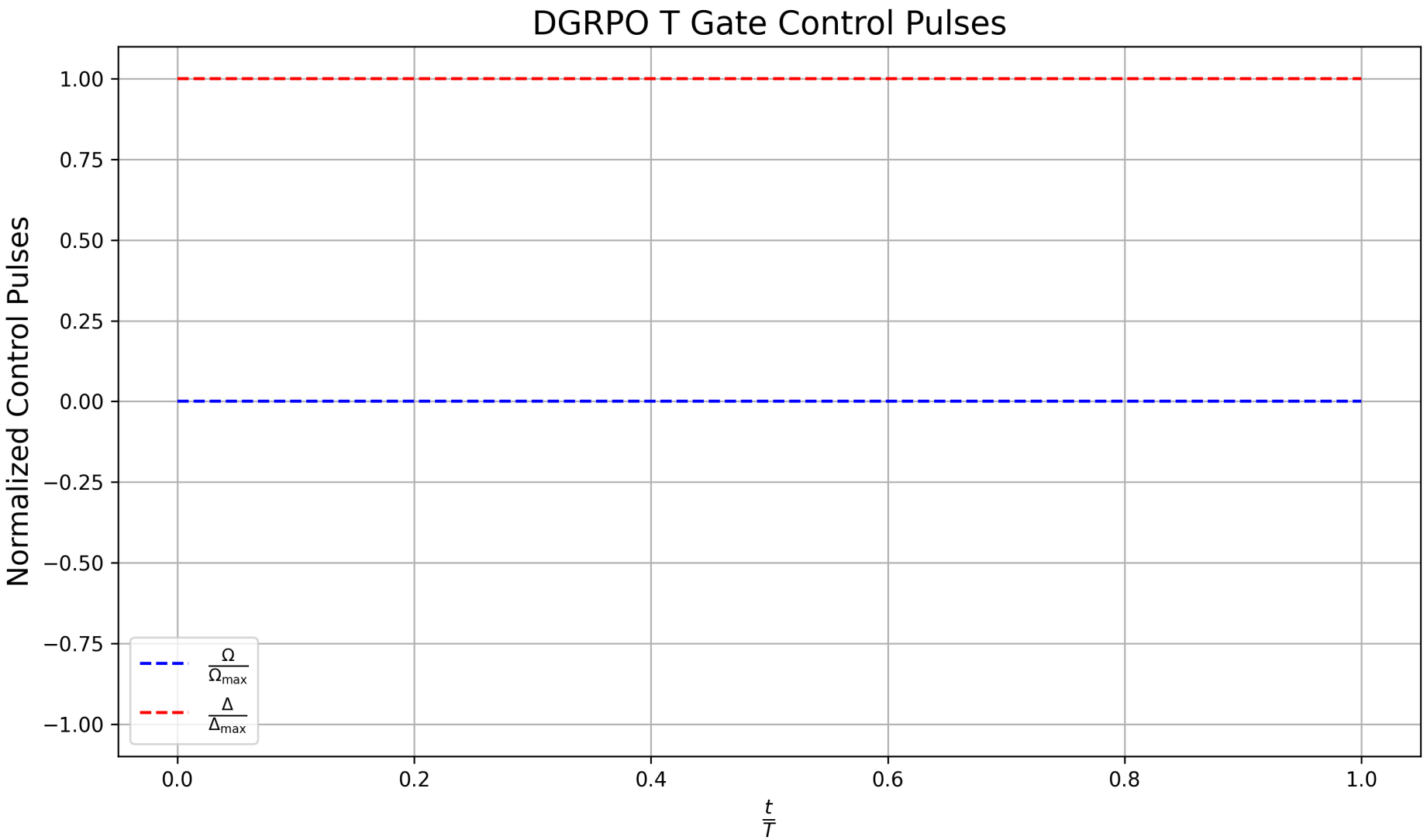

T-Gate Maximum Fidelity Trajectory Log Infidelitity per episode
DDDQN
DPPO
CPPO
DGRPO
CGRPO
TD3
Log Infidelity
t / T_max
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
0.0
0.2
0.4
0.6
0.8
1.0

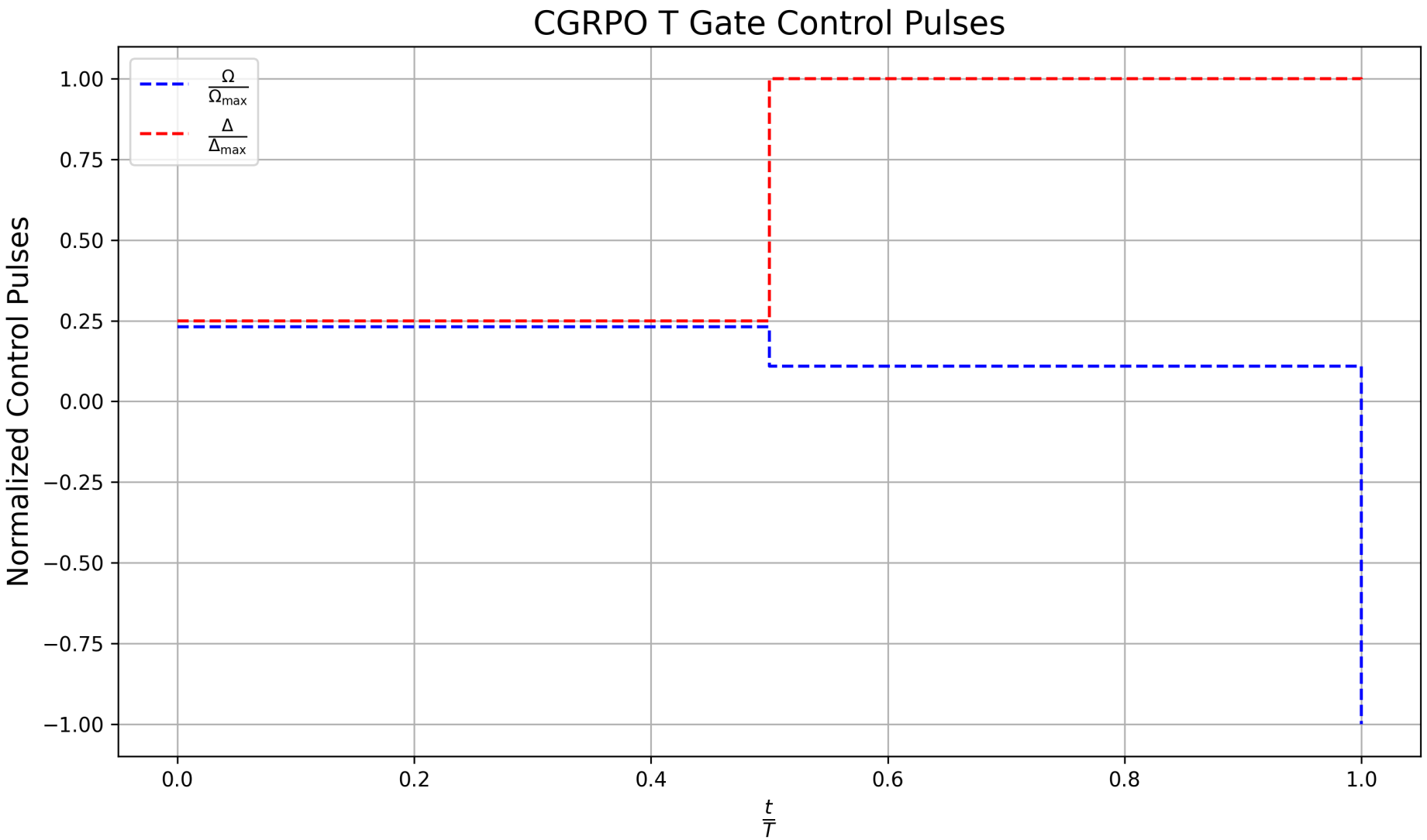
T Gate Control Pulses

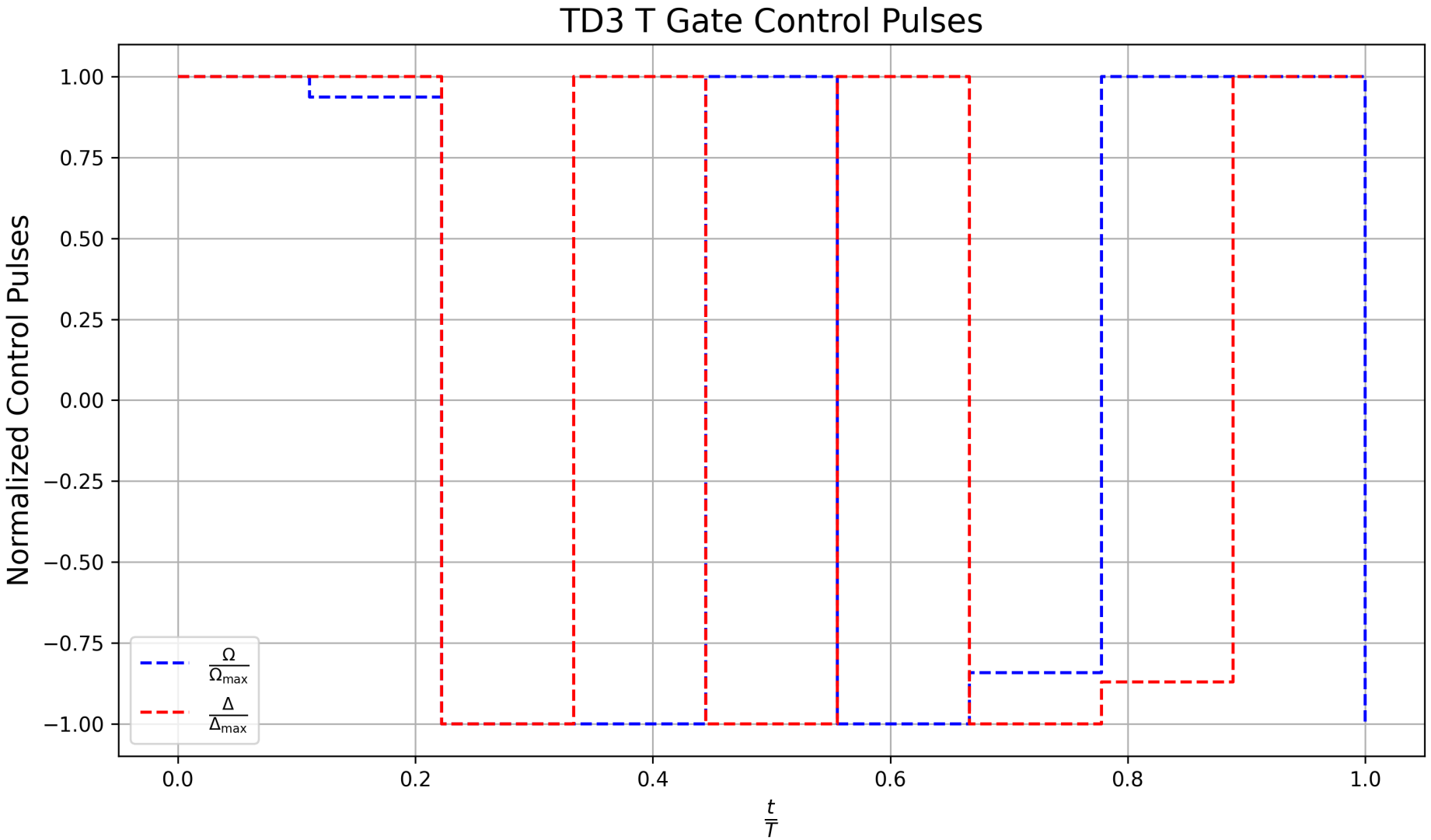



DPPO T Gate Control Pulses
Normalized Control Pulses
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0
$\frac{t}{T}$
$\frac{\Omega}{\Omega_{max}}$
$\frac{\Delta}{\Delta_{max}}$

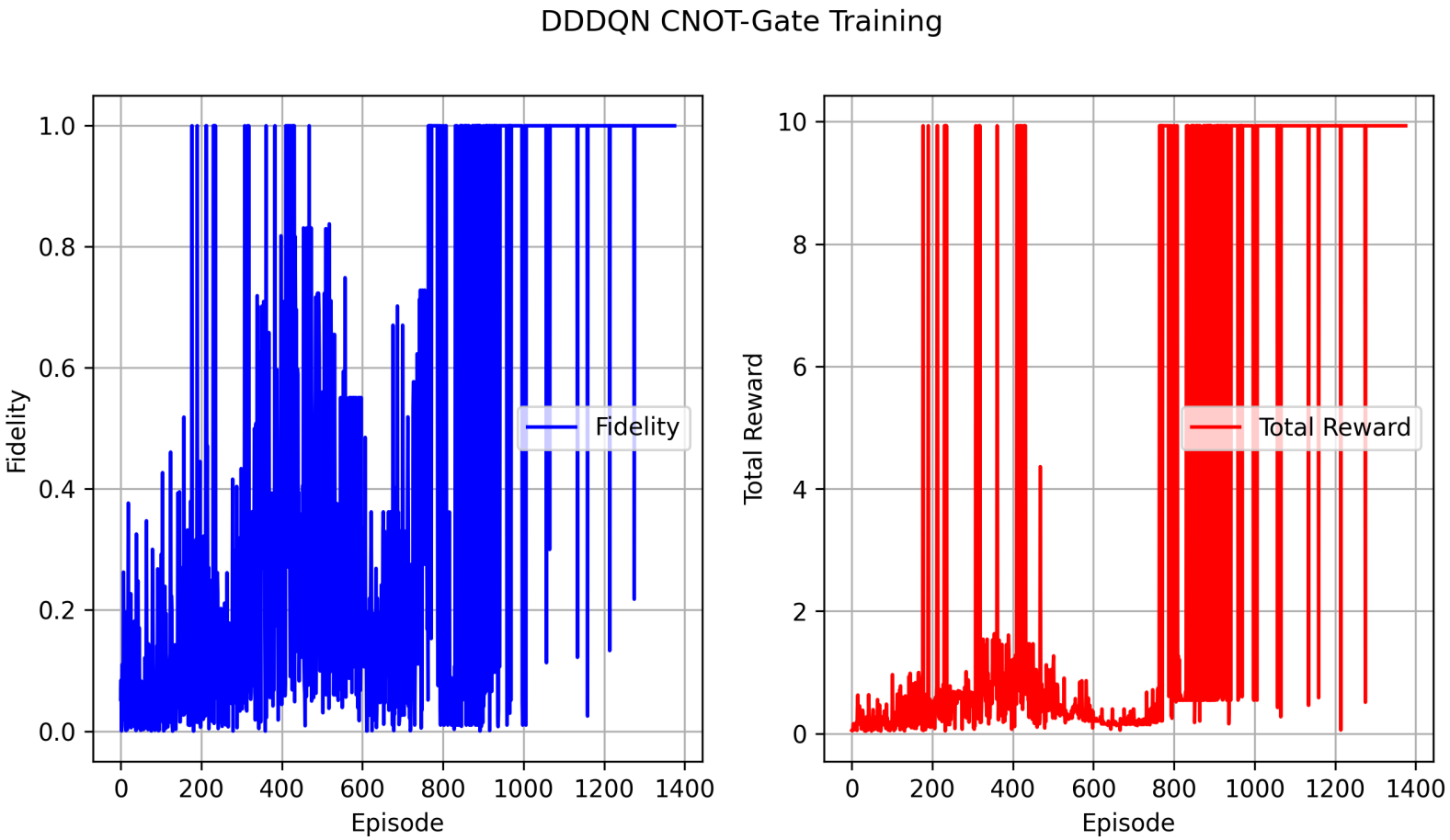

CPPO T Gate Control Pulses
Normalized Control Pulses
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0
$\frac{t}{T}$
$\frac{\Omega}{\Omega_{max}}$
$\frac{\Delta}{\Delta_{max}}$

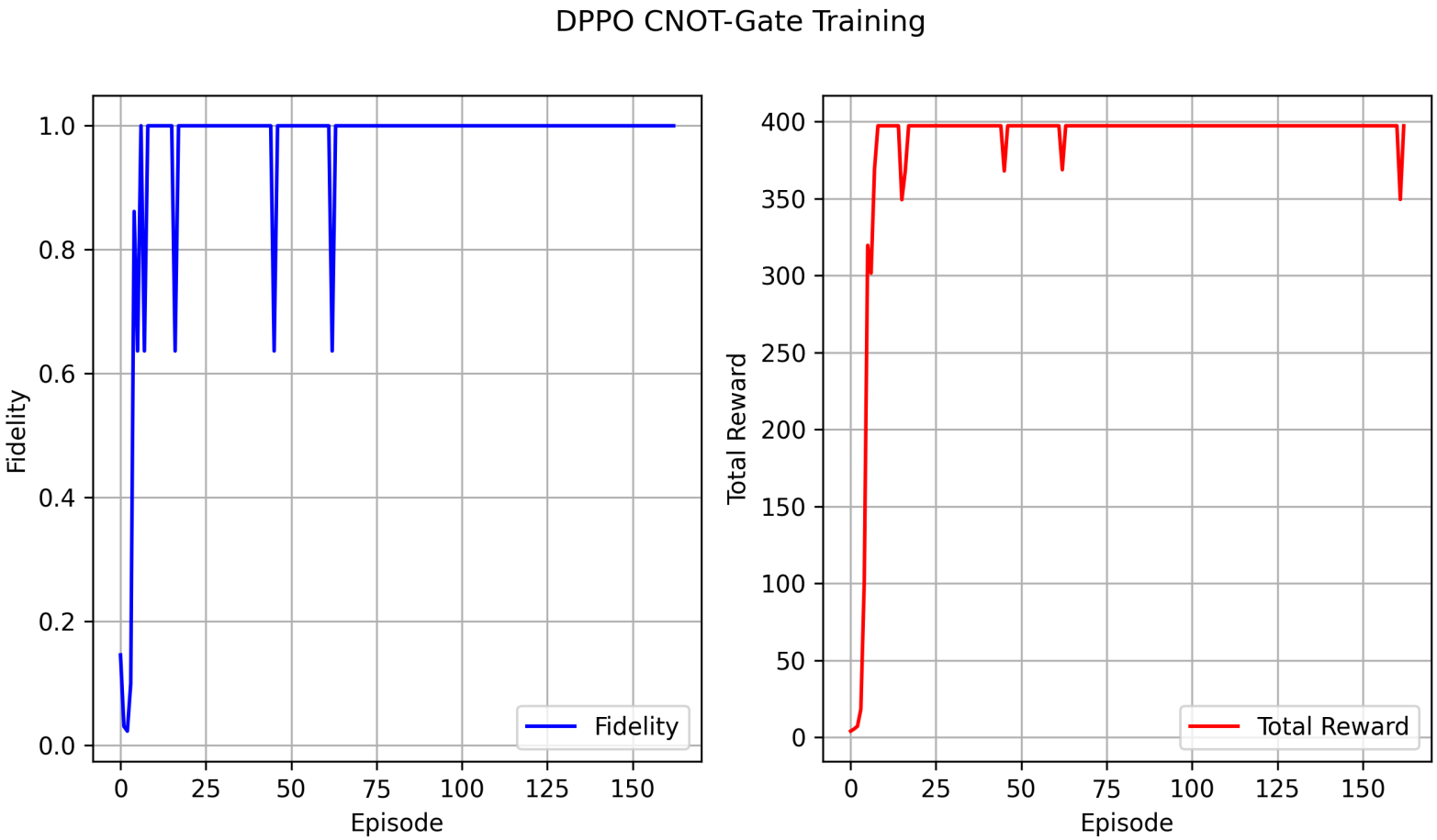

DGRPO T Gate Control Pulses
Normalized Control Pulses
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0
$\frac{t}{T}$
$\frac{\Omega}{\Omega_{max}}$
$\frac{\Delta}{\Delta_{max}}$

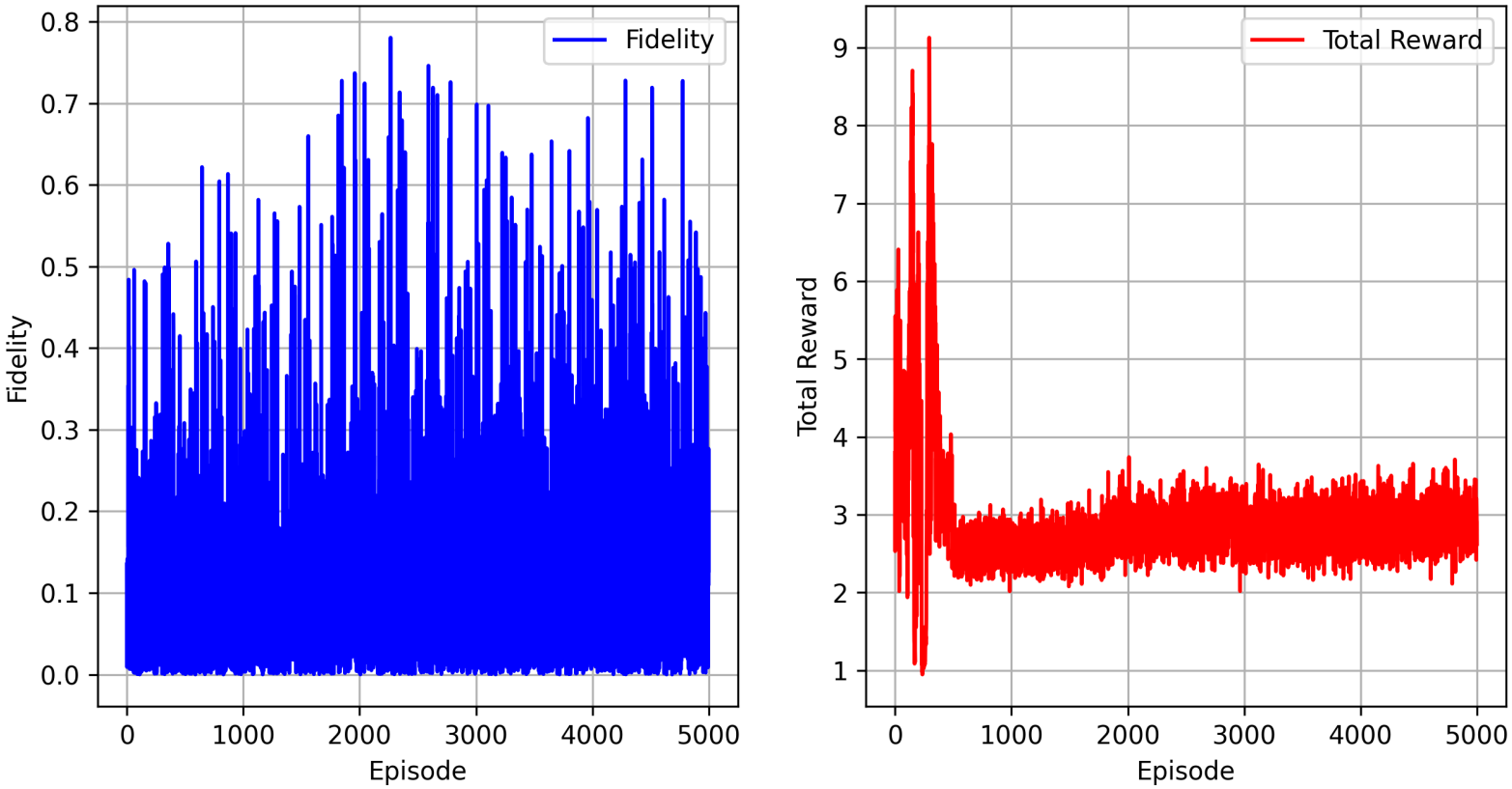

CGRPO T Gate Control Pulses
Normalized Control Pulses
$\frac{\Omega}{\Omega_{max}}$
$\frac{\Delta}{\Delta_{max}}$
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0
$\frac{t}{T}$

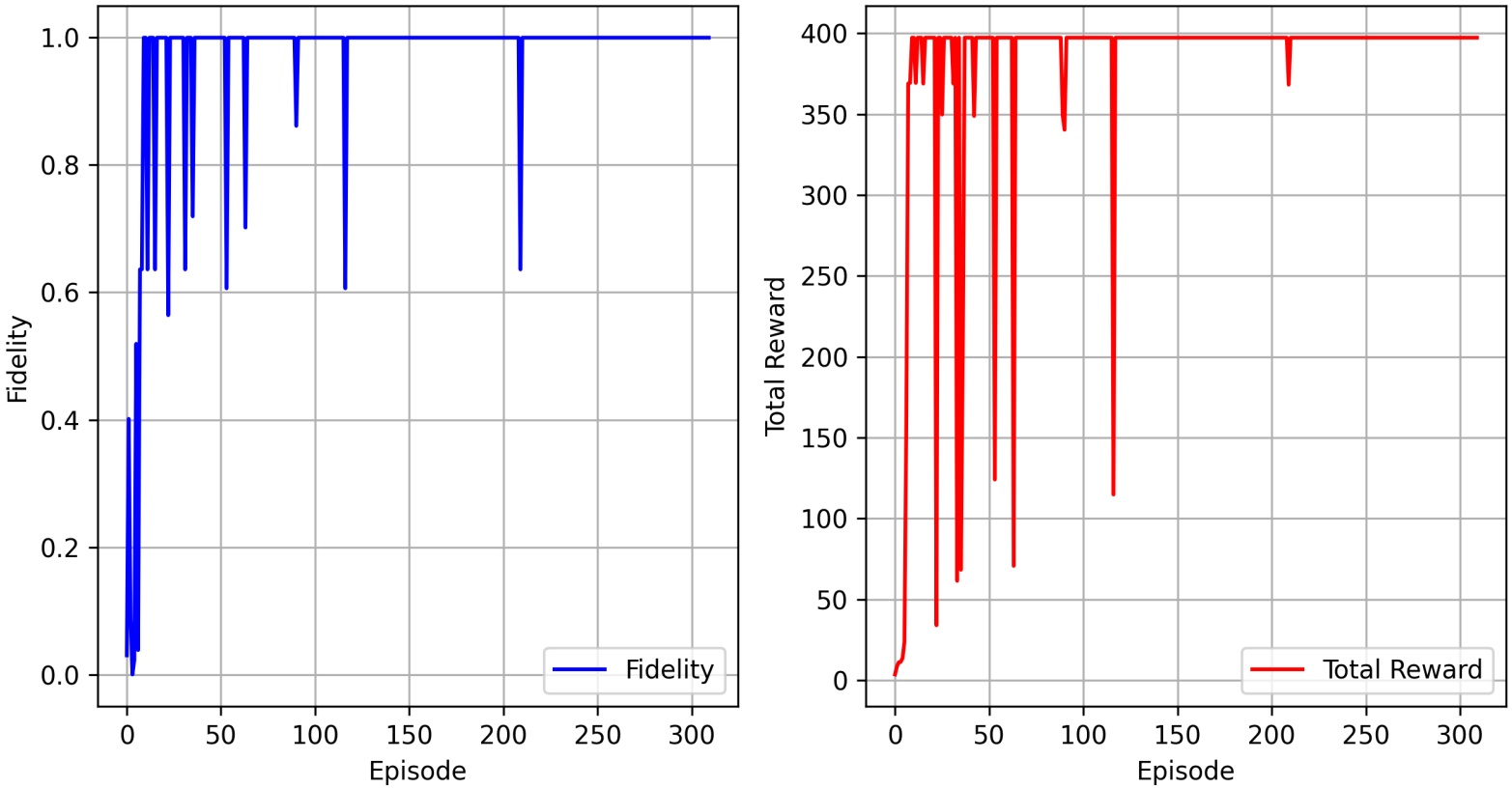

TD3 T Gate Control Pulses
Normalized Control Pulses
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0
$\frac{t}{T}$
$\frac{\Omega}{\Omega_{max}}$
$\frac{\Delta}{\Delta_{max}}$

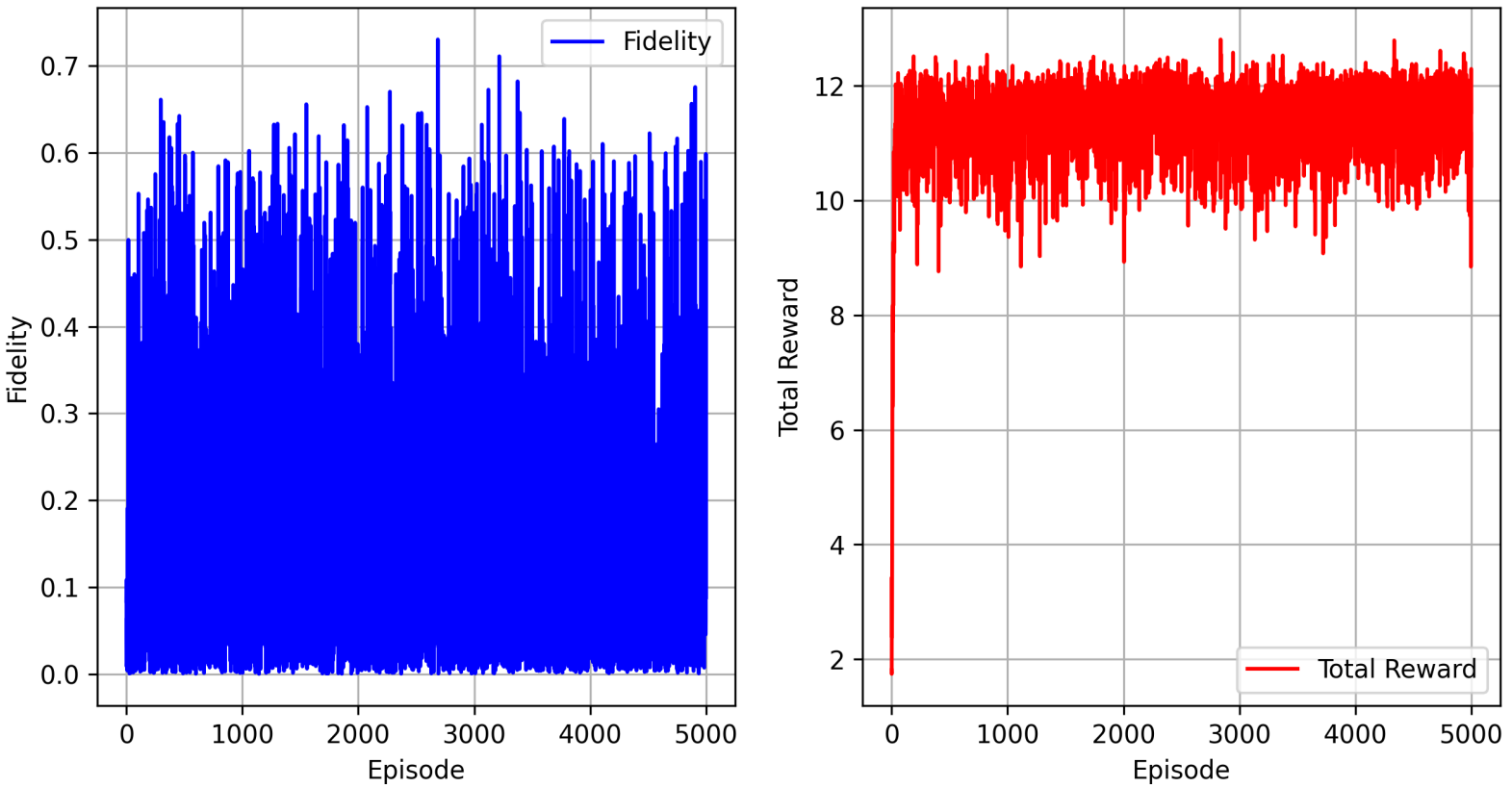
CNOT Gate Training

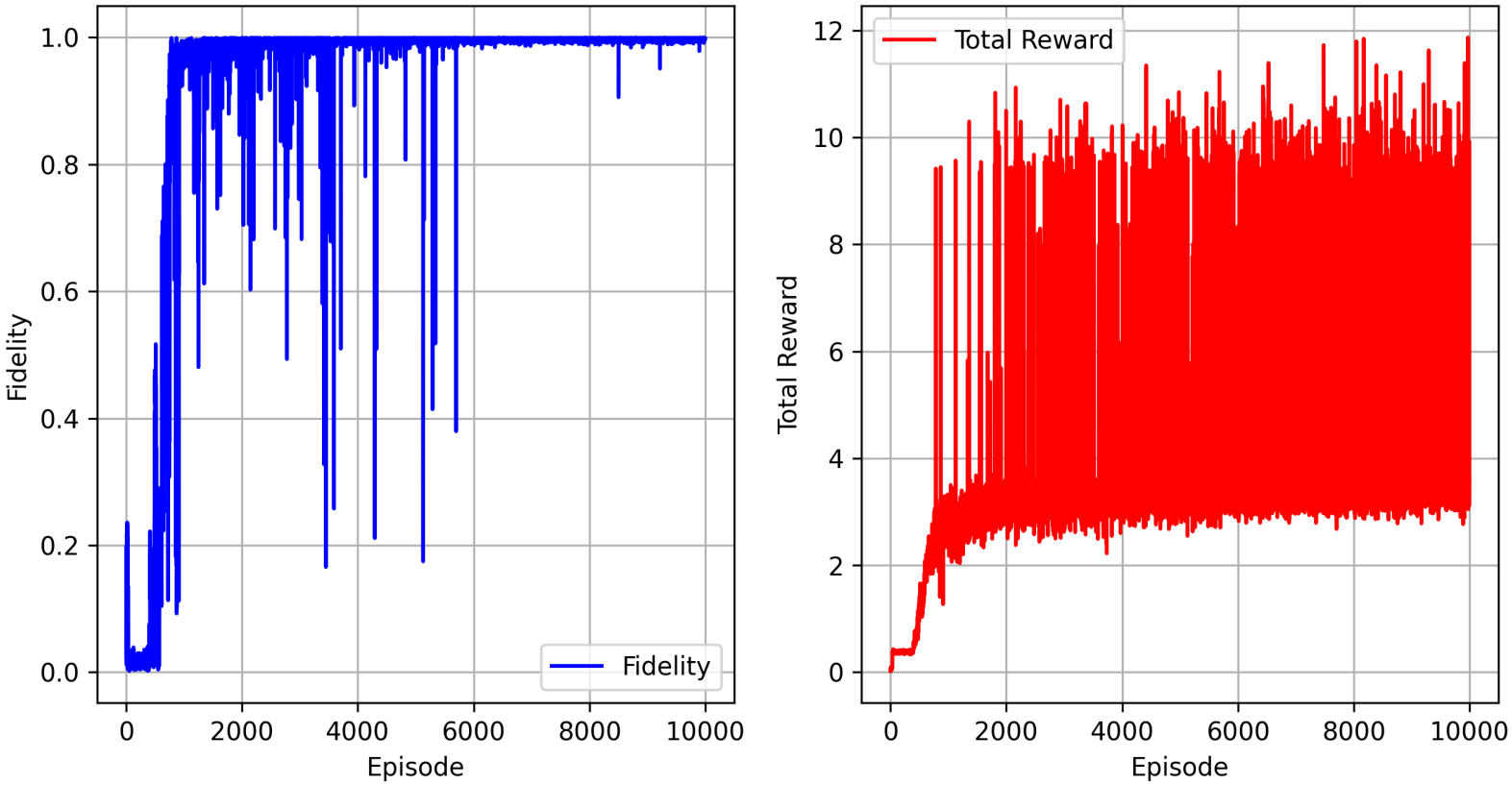



DPPO CNOT-Gate Training
Fidelity
Episode
Fidelity
Total Reward
Episode
Total Reward


CPPO CNOT-Gate Training
Fidelity
Episode
Total Reward
Episode

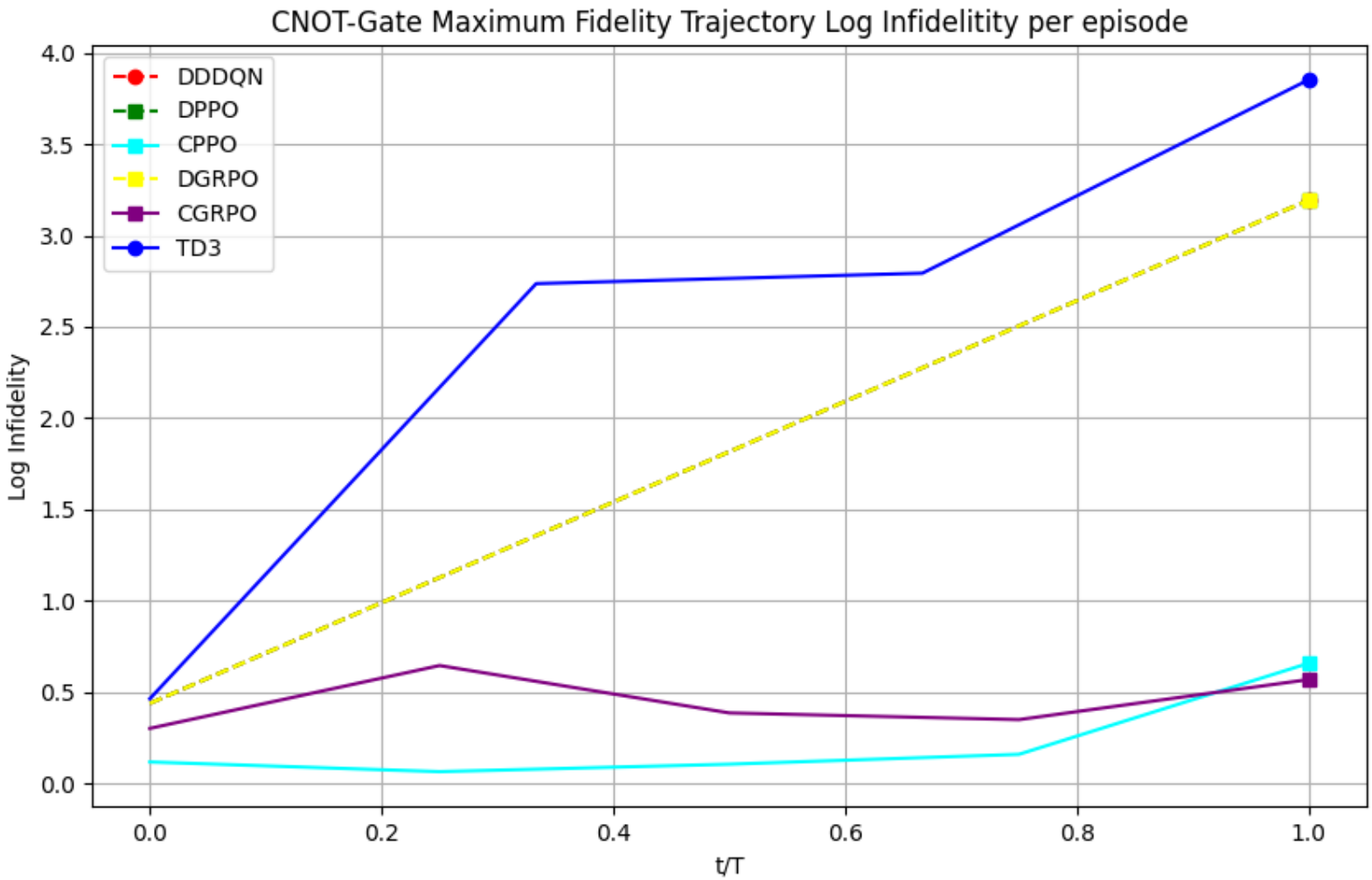

DGRPO CNOT-Gate Training
Fidelity
Episode
Fidelity
Total Reward
Episode
Total Reward


CGRPO CNOT-Gate Training
Fidelity
Episode
Total Reward
Episode
Total Reward

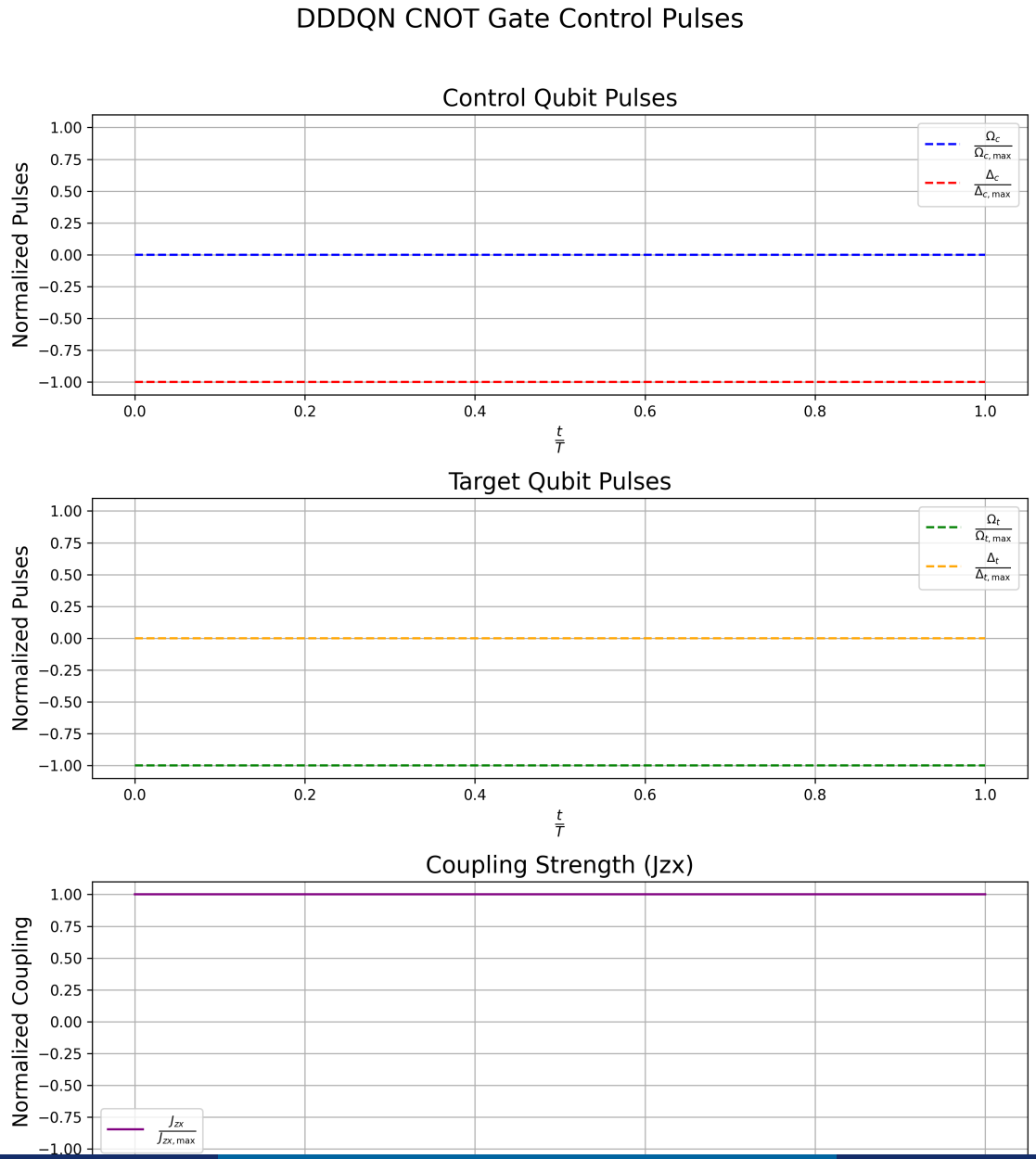

TD3 CNOT-Gate Training
Fidelity
Episode
Fidelity
Total Reward
Episode
Total Reward

CNOT Gate Performance Metrics

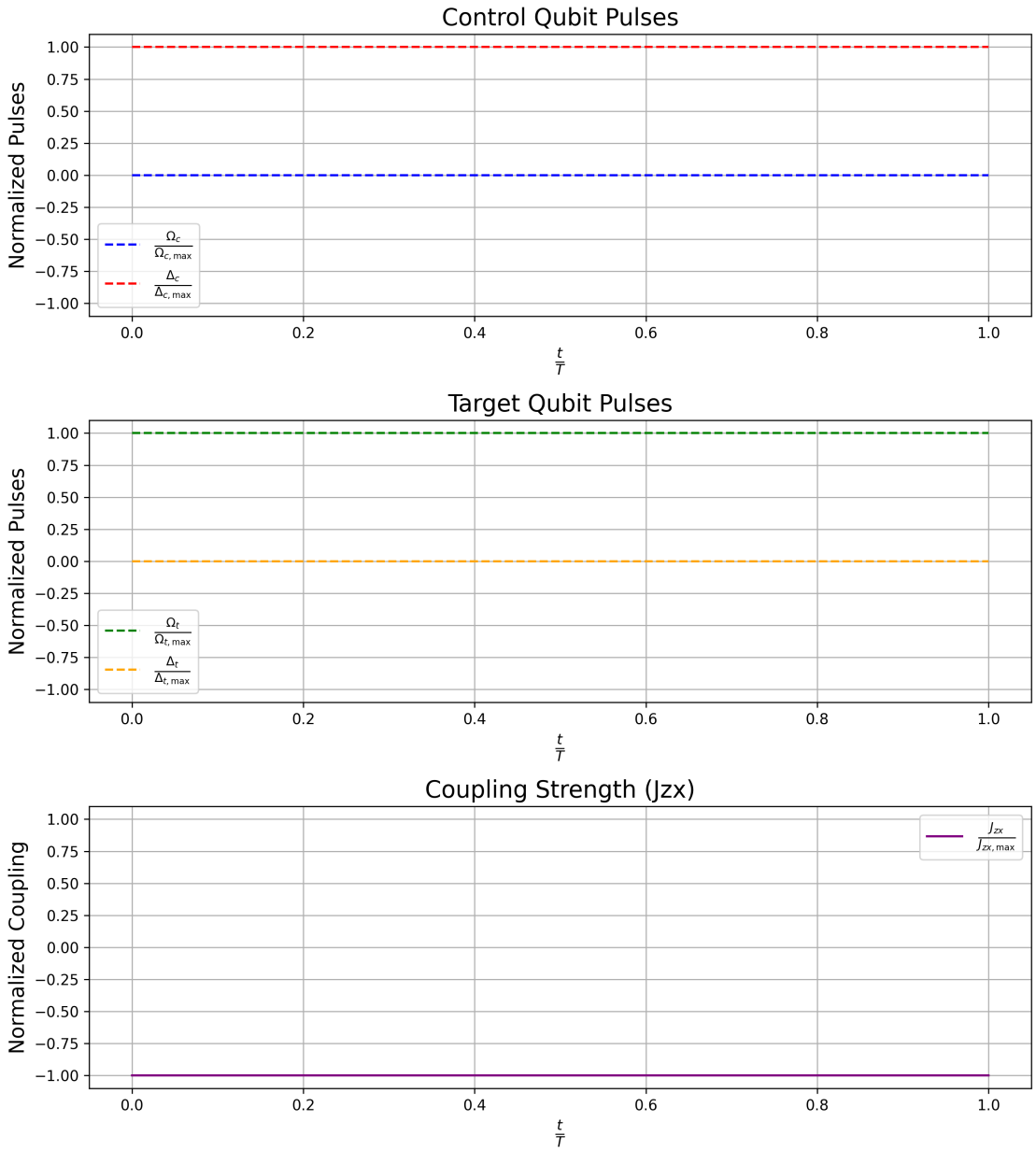
	Agent	Max Fidelity	Max Log Infidelity	Avg Fidelity	Time Steps
4	GRPO-C	0.730429	0.569327	0.784343	5
2	PPO-C	0.780537	0.658638	0.824429	5
0	DDDQN	0.999361	3.194187	0.999488	2
1	PPO-D	0.999361	3.194187	0.999488	2
3	GRPO-D	0.999361	3.194187	0.999488	2
5	TD3	0.999860	3.852707	0.999888	4


CNOT-Gate Maximum Fidelity Trajectory Log Infidelitity per episode
DDDQN
DPPO
CPPO
DGRPO
CGRPO
TD3
Log Infidelity
t/T
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.2
0.4
0.6
0.8
1.0

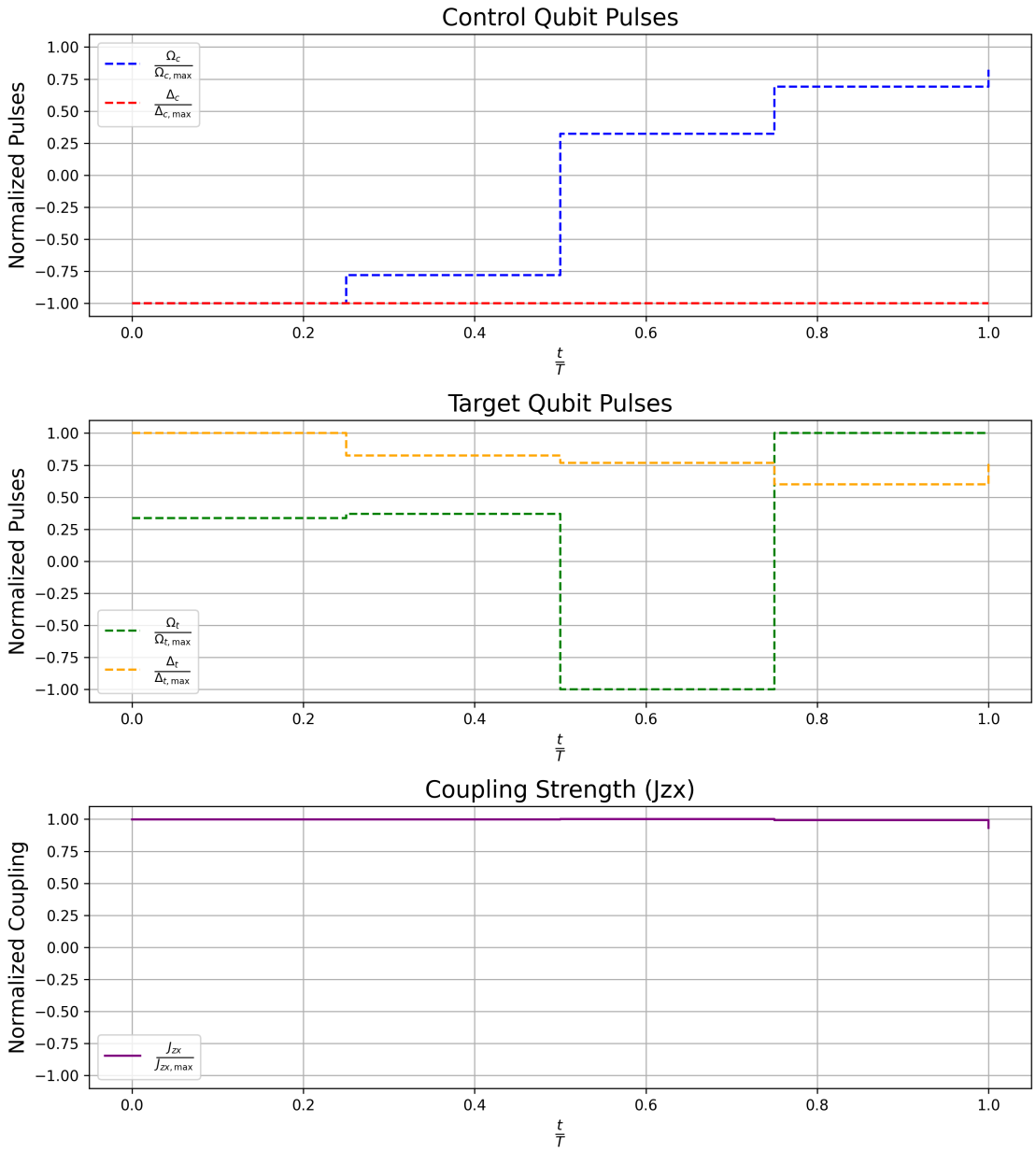
H Gate Control Pulses



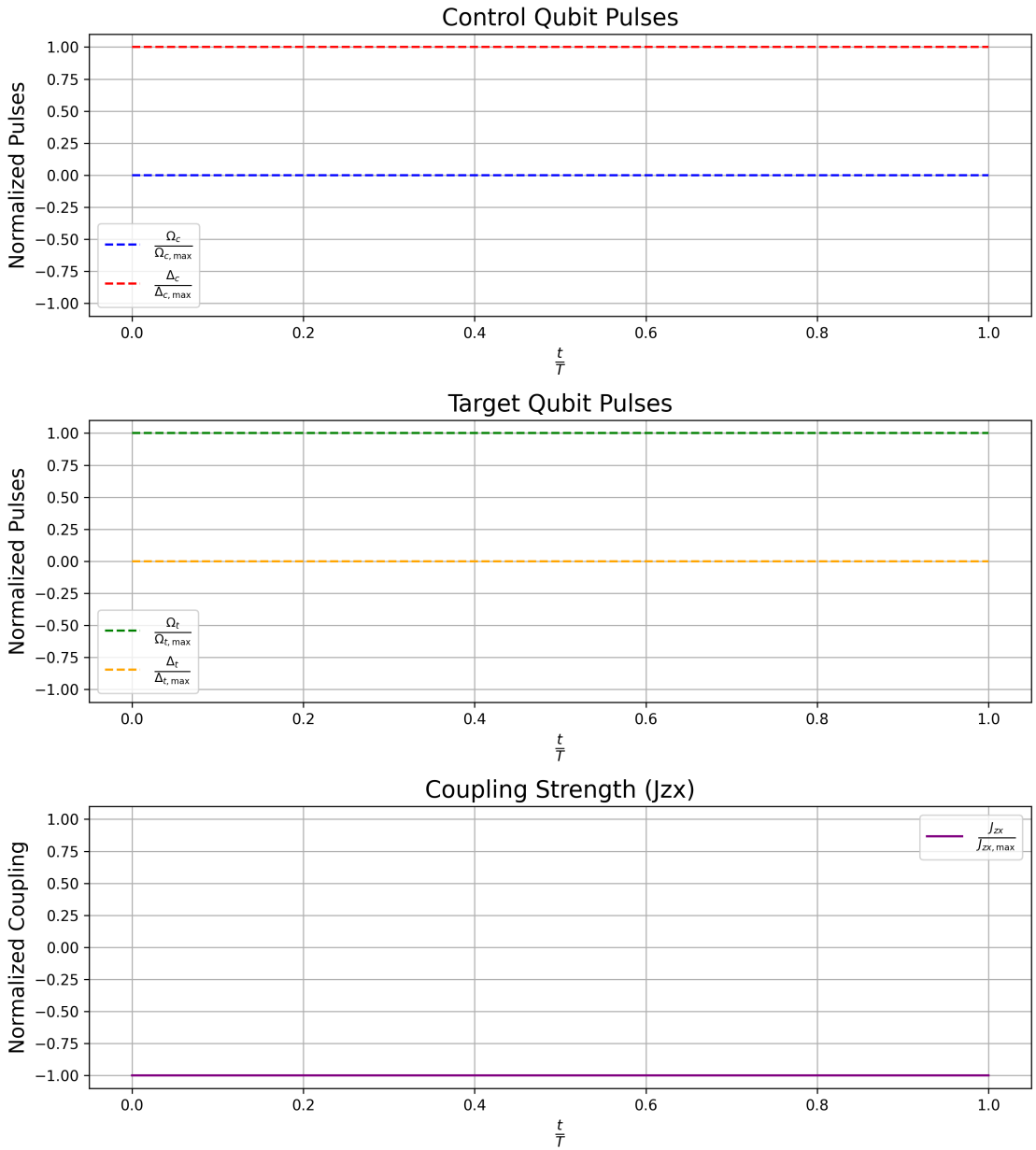
DPPO CNOT Gate Control Pulses



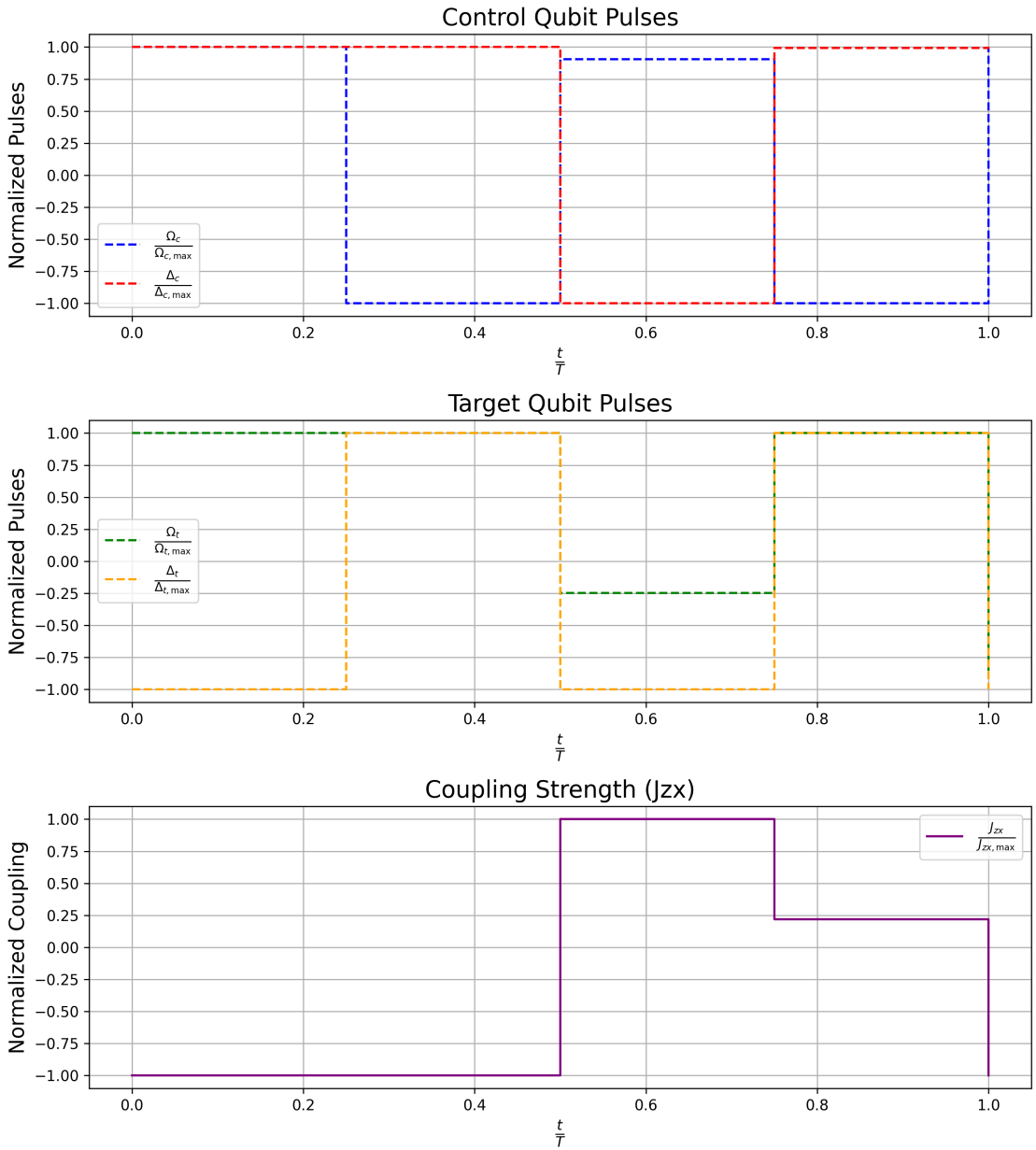
CPPO CNOT Gate Control Pulses



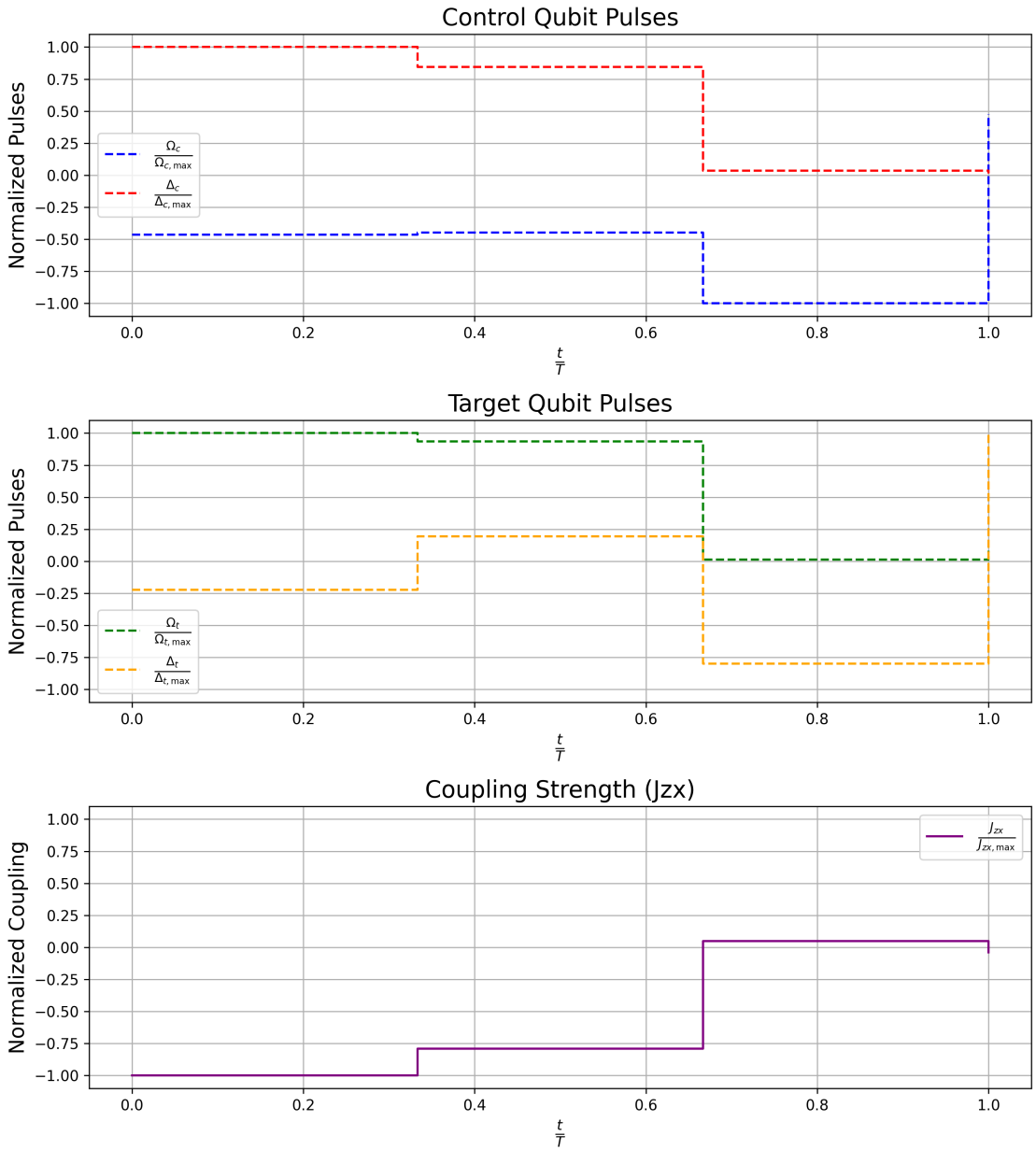
DGRPO CNOT Gate Control Pulses



CGRPO CNOT Gate Control Pulses



TD3 CNOT Gate Control Pulses



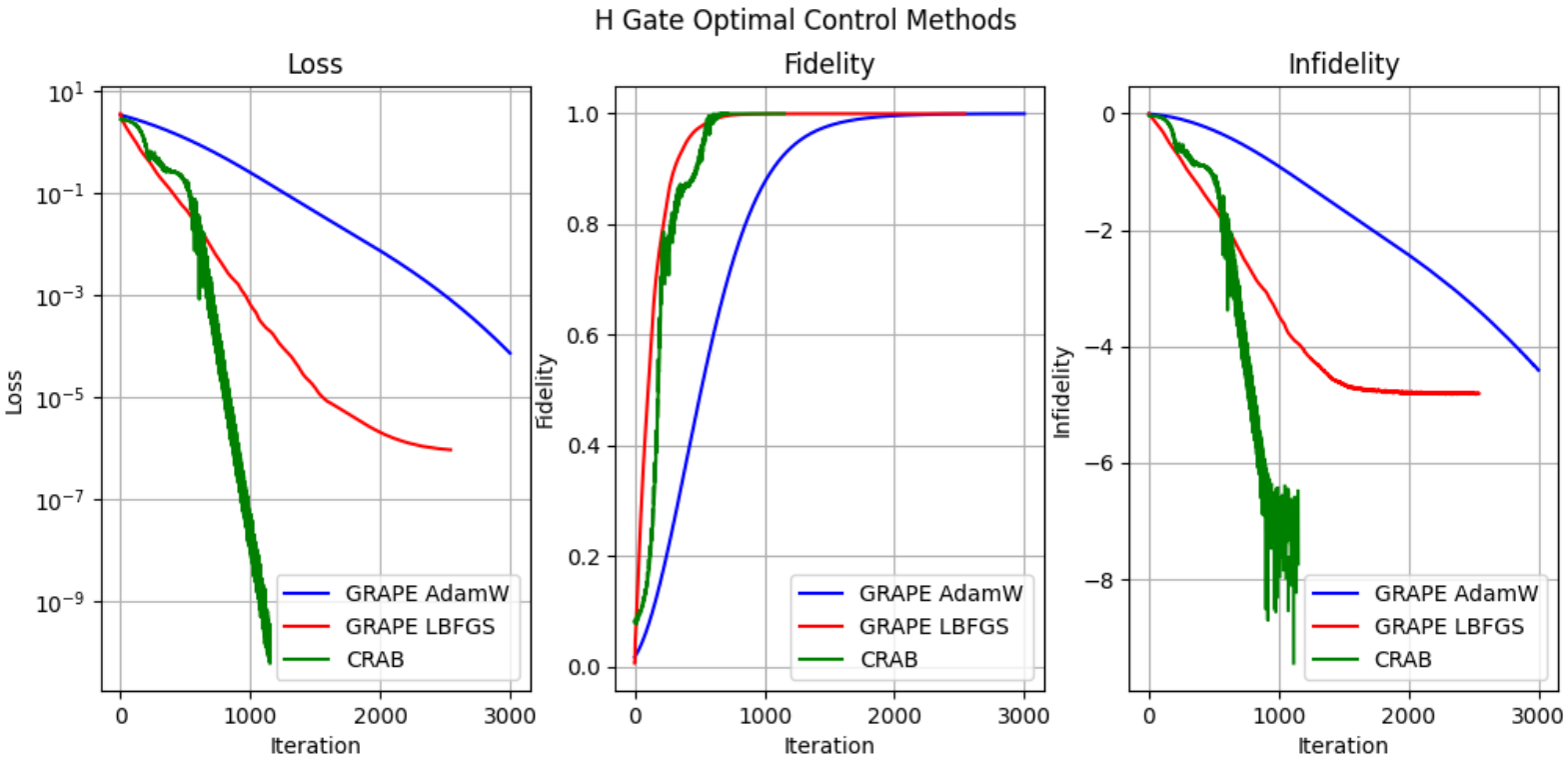
Summary

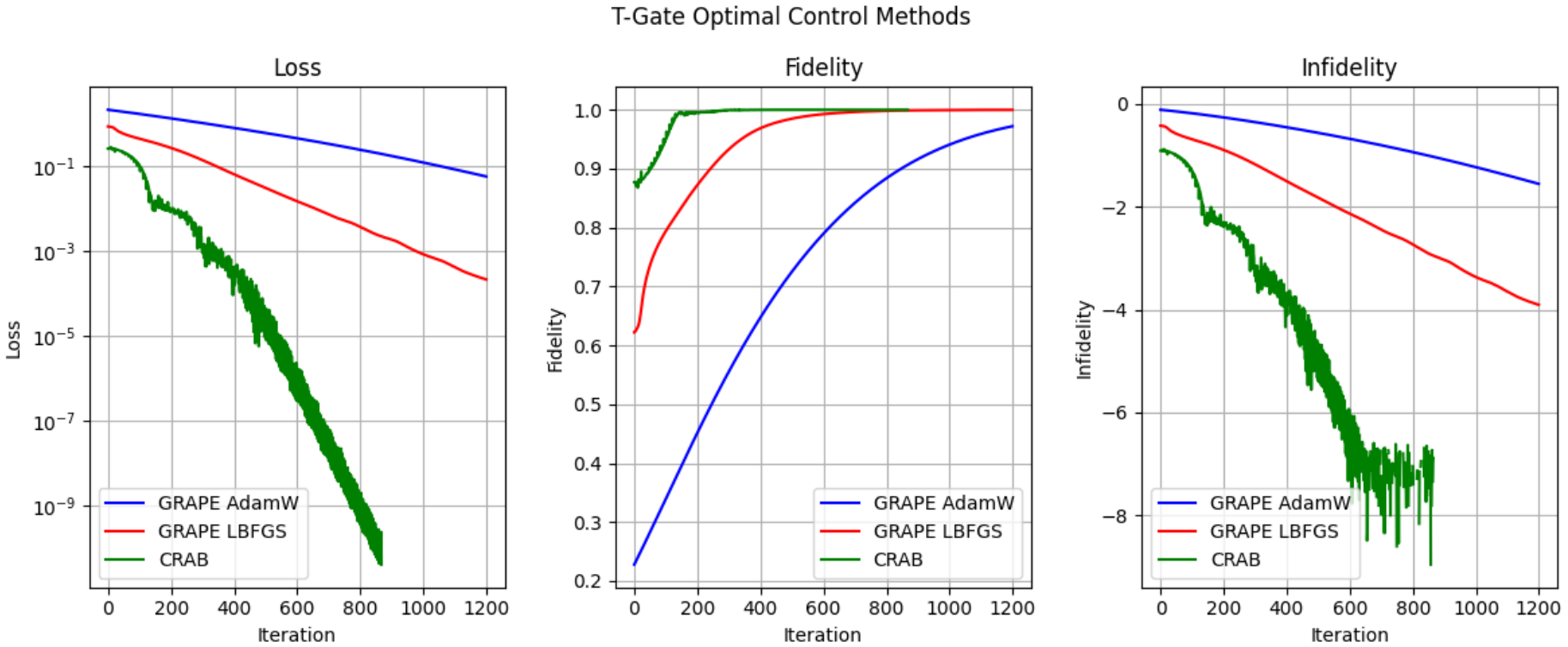
DDDQN, DPPO and DGRPO required the least number of episodes to achieve high fidelities. TD3 converged in the second place. CPPO and CGRPO as they struggle to achieve stable learning curves, they required more time than deterministic or discrete approaches.

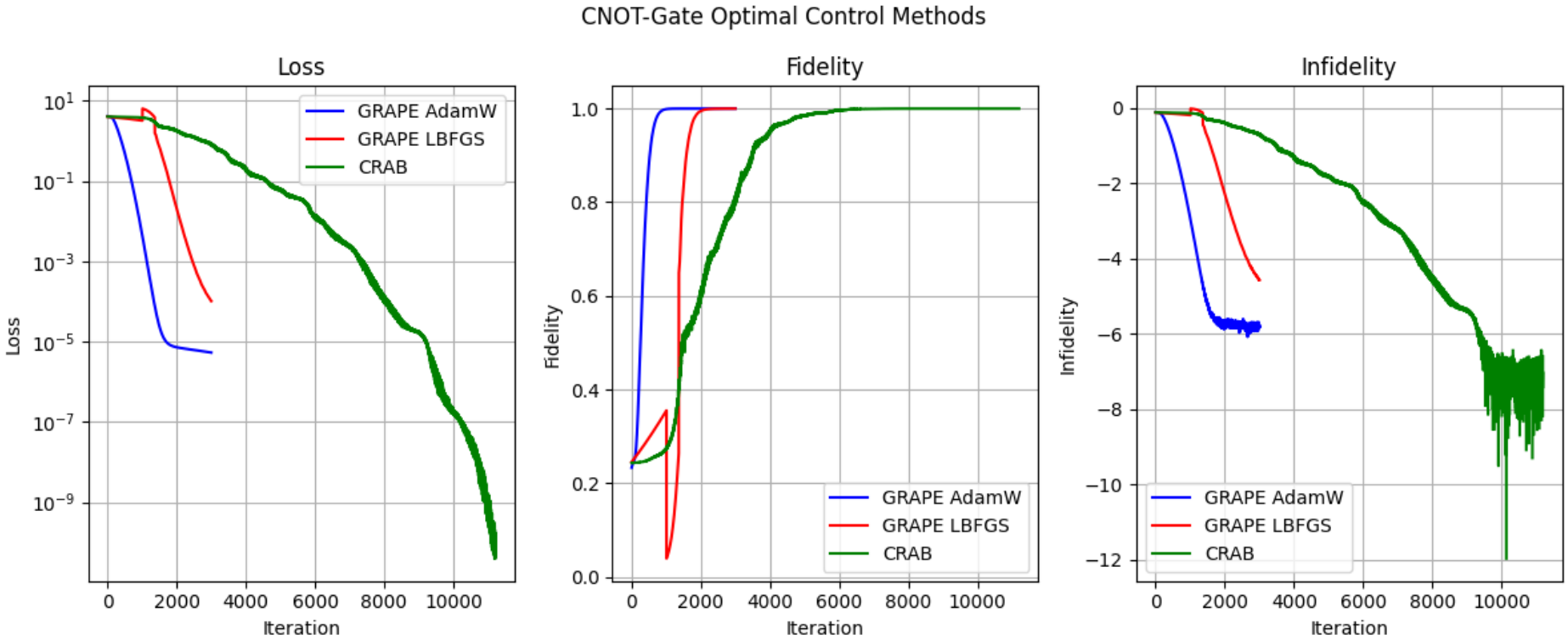
TD3 exhibited minor fluctuations due to the approximation errors of the function. DDDQN, DPPO and DGRPO showed the most stable learning due to their deterministic way of solving the problem. CPPO and CGRPO struggle due to General Advance Estimation (GAE) and stochasticity. DGRPO and GRPO were in line with the deterministic albeit stochastic due to categorical sampling in a finite and small discrete set.

To ensure the accuracy of the RL-based quantum control, the results were compared with GRAPE and CRAB.

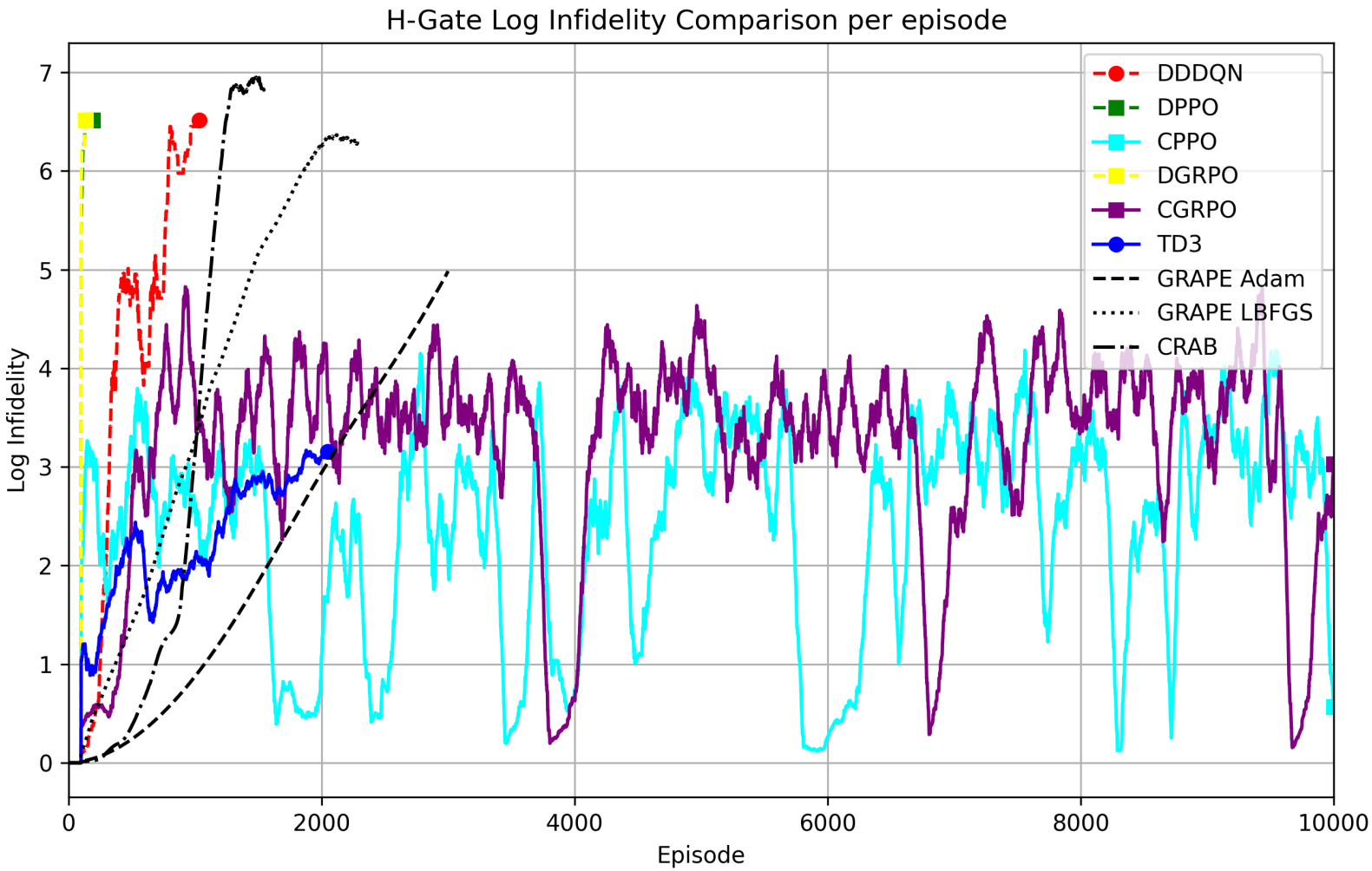
Optimal Quantum Control

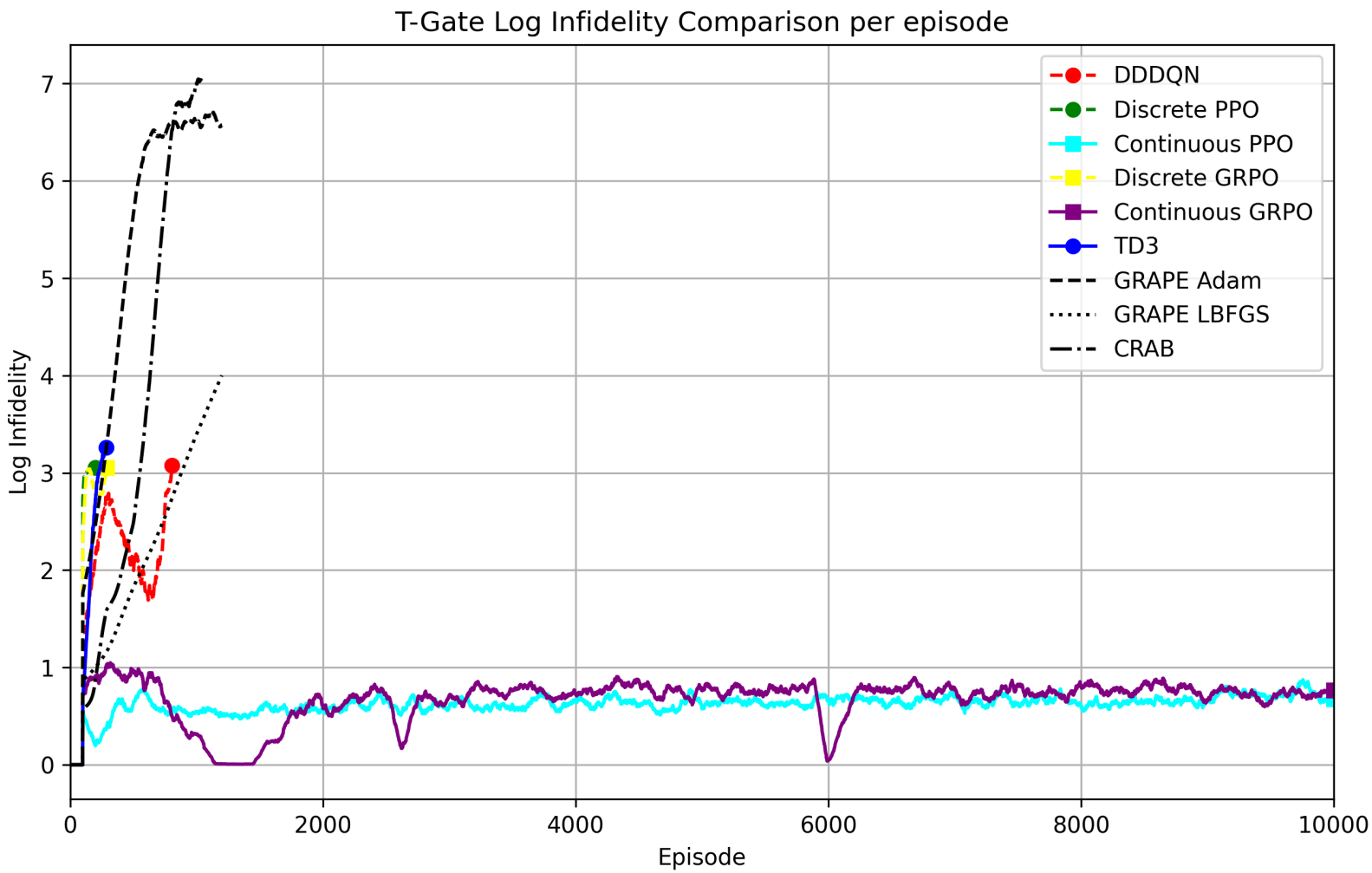


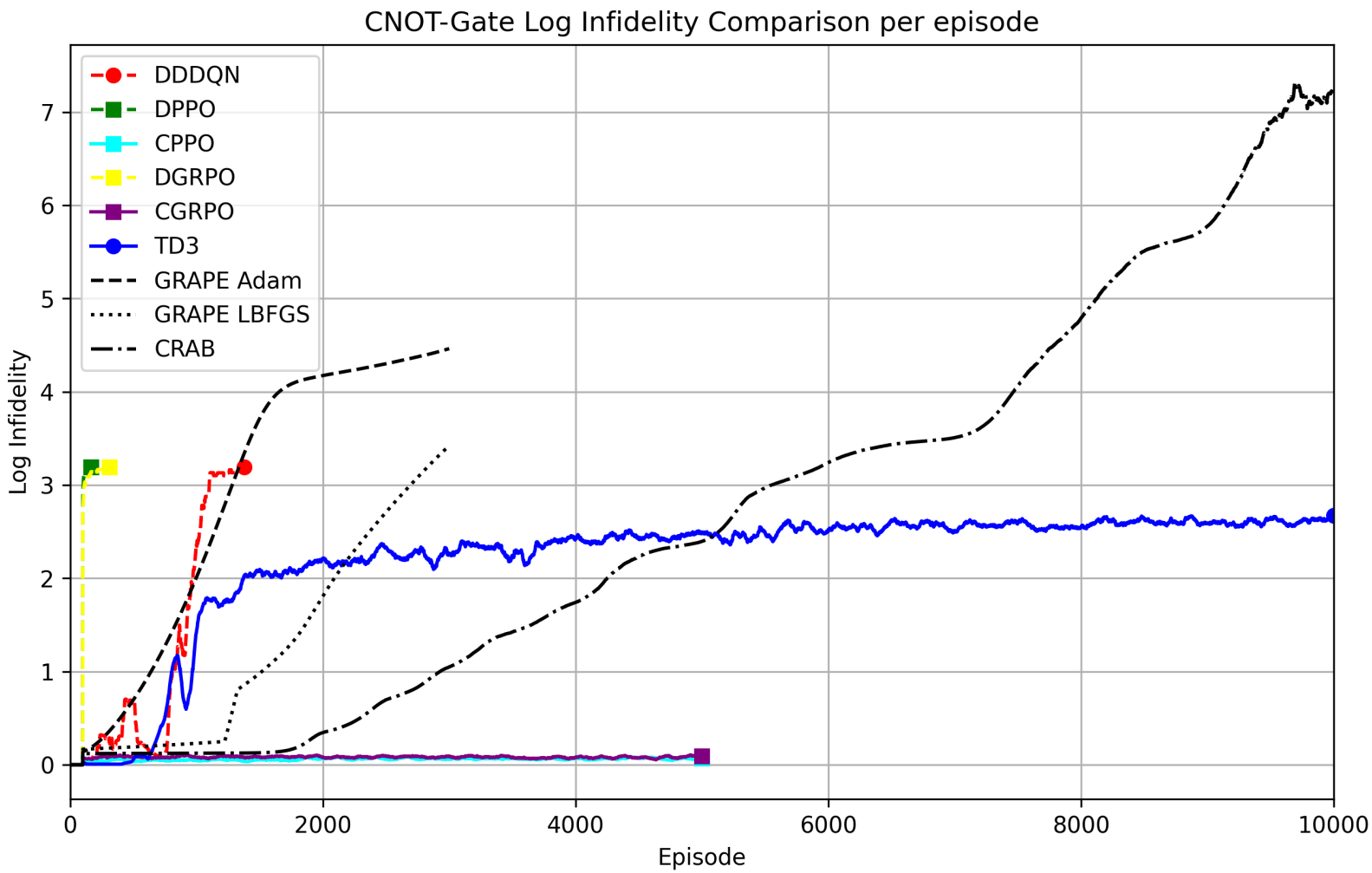

T-Gate Optimal Control Methods
Loss
Fidelity
Infidelity
Loss
Fidelity
Infidelity
Iteration
Iteration
Iteration
GRAPE AdamW
GRAPE LBFGS
CRAB
GRAPE AdamW
GRAPE LBFGS
CRAB
GRAPE AdamW
GRAPE LBFGS
CRAB


CNOT-Gate Optimal Control Methods
Loss
Fidelity
Infidelity
GRAPE AdamW
GRAPE LBFGS
CRAB
Iteration
Loss
Fidelity
Infidelity

Comparisons




T-Gate Log Infidelity Comparison per episode
DDDQN
Discrete PPO
Continuous PPO
Discrete GRPO
Continuous GRPO
TD3
GRAPE Adam
GRAPE LBFGS
CRAB
Log Infidelity
Episode
0
1
2
3
4
5
6
7
0
2000
4000
6000
8000
10000


CNOT-Gate Log Infidelity Comparison per episode
DDDQN
DPPO
CPPO
DGRPO
CGRPO
TD3
GRAPE Adam
GRAPE LBFGS
CRAB
Log Infidelity
Episode
0
1
2
3
4
5
6
7
0
2000
4000
6000
8000
10000

Conclusions

Applications

1. Algorithms:
   Quantum Fourier Transform
   Grover's Search
   VQA

2. Scalability:
   HRL
   MARL
   Large Circuit Compilation

3. Error Correction/Mitigation:
   FTQC
   QEC
   Logical Gate Optimization and Compilation

Future Work

1. Implementation of RL-optimized quantum gates on real quantum hardware.
2. Hybrid quantum-classical RL methods. An approach is quantum reinforcement learning (QRL).
3. Multi-agent reinforcement learning or hierarchical reinforcement learning exploration.
4. RL-driven error mitigation techniques that enable RL agents to dynamically adjust gate parameters to compensate for decoherence.
5. RL policies to adapt control sequences in real time based on measured noise fluctuations in quantum hardware.
6. Improve fault-tolerant operations by optimizing quantum error correction strategies via RL.

Acknowledgments

DEMOCRITUS UNIVERSITY OF THRACE | DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

MSc in QUANTUM COMPUTING AND QUANTUM TECHNOLOGIES

NATIONAL CENTRE FOR SCIENTIFIC RESEARCH "DEMOKRITOS"

Thanks for your attention

Questions?

DEMOCRITUS UNIVERSITY OF THRACE | DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

MSc in QUANTUM COMPUTING AND QUANTUM TECHNOLOGIES

NATIONAL CENTRE FOR SCIENTIFIC RESEARCH "DEMOKRITOS"