Quantum Entanglement Swapping Management with Reinforcement Learning, Open AI - GYM

Antoine Cousson & Ramon Aparicio Pardo
Inria and I3S , Sophia-Antipolis, France
raparicio@i3s.unice.fr
antoine.cousson@etu.unice.fr

1. Introduction

1.1 Entanglement as fundamental resource [1]

Quantum communication are based on **entanglement**. When two qubits, the quantum counterparts of a classical bit, are entangled, their individual states cannot be separately described: a state change, i.e., a qubit reading, in one of them implicitly comes with a change in the other one, regardless the physical distance between them. Thus, the *readings* at the two entangled qubits *exhibit non-classical correlations* that can be used to design new *applications* not possible with classical communication, such us *quantum cryptography* or *distributed quantum computation.*

1.2 Elementary link vs virtual link generation [1]

An **elementary link** is an entanglement between two qubits located at two physically separated nodes (e.g., between nodes A and B in Fig.1). Its *success probability* P_e exponentially decreases with distance, which means that short long distance entanglements (e.g., between A and B, or B and C, in Fig. 1) have more likely to succeed than long distance entanglements (e.g., between A and C, in Fig. 1). To overcome this issue, we can create a **virtual link** (e.g. A-C) over two elementary links (e.g., A-B and B-C) via **entanglement swapping.** This process consumes the previously generated elementary links on a path between two end-points to produce a new entangled pair between the two remote ends. When this process must be successively iterated to create very long distance entanglements, the intermediate generated entanglements must be stored on the so-called **quantum memories** to be consumed later.

1.3 Quantum memory lifetimes [1]

The probability that a qubit stored in a quantum memory is still, after a certain time, in its original state (e.g., an entangled state) decreases with time. This probability is referred as to *memory efficiency* η_m [2], and its decay is known as **decoherence.** This process is the consequence of the progressive interactions of the quantum memory with the environment, since a memory cannot be perfectly isolated from it. The *entanglement swapping success probability* P_s depends on the *memory efficiency* η_m of the oldest loaded quantum memory involved in the swapping.

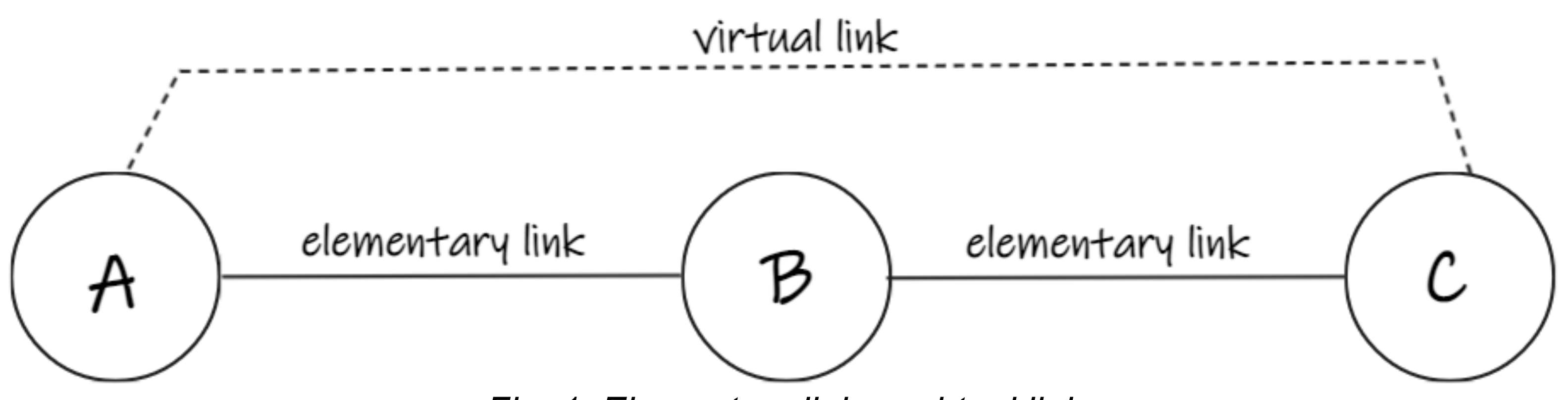

Fig. 1. Elementary link vs virtual link

2. Problem Statement as Markov Decision Process (MDP)

The management of a *virtual link generation over two elementary links via entanglement swapping* can be formulated as a Markov Decision Process (MDP) [3]. In this process, we aim to maximize the number of successful entanglement swaps per time unit, i.e., *the virtual link generation rate*. To generate the virtual link, two elementary links must have been successfully created before attempting the entanglement swapping with probability P_s. The older the first generated elementary link is, the more likely the swapping will fail (see Section 1). Then, after a so-called *cutoff time* t_c, discarding the oldest elementary link and *resetting* it becomes beneficial. Unfortunately, this reset is a stochastic process with *success probability* P_e, which must be repeated till success, delaying the swapping attempt. Thus, the *cutoff time* t_c of the elementary link has to be carefully selected to reduce the time between two successful swaps and maximize the virtual link generation rate.

Now, we describe the aforementioned process as a MDP. A managing agent can apply a certain ***action*** *a* after observing the current ***state*** *s*. The execution of this action will trigger a transition into a new state *s'* with a certain probability. The agent receives a ***reward*** *r* based on the "quality" of the pair *(s,a)* to maximize the objective. In our case, state, action space and reward can be characterized as follows.

The system state (action) consists of the concatenation of the states (actions) of the two elementary links and the virtual link. The ***state*** of each link is a vector *s= [x,m],* where

- ***x*** = {1 if entanglement is active; 1, otherwise}
- ***m*** is the entanglement age (-1, if entanglement is inactive)

The ***action space*** of each link can take two values:

- **0: WAIT**, i.e., to do nothing.
- **1: (RE)SET,** i.e., to (re)try the link generation with a success probability P_e (P_s) for a elementary (virtual) link.

The ***reward*** is defined as:

- ***r*** = {1 if entanglement swapping succeeds; 0, otherwise}

3. Reinforcement Learning (RL)

If we do not possess models characterizing precisely the *elementary link generation probability* P_e and the *memory efficiency* η_m (and, thus, the *entanglement swapping success probability* P_s), the state transition probabilities are unknown. In this case, we can apply Deep Reinforcement Learning (DRL) [4],[5] to find a policy (set of actions depending on the current state) that maximizes the *virtual link generation rate.* Here, this policy consists of finding the best *cutoff time* t_c for the oldest elementary link (see Section 2).

To do that, we use `OpenAI Gym` [6] to program in `Python` an environment modelling the MDP as described in Section 2. The agent is also programmed in `Python` as a neural network that outputs an action *a* based on the input state *s.* The neural network is trained using the DRL algorithm called Deep Q-Network (DQN) [5] provided by the `OpenAI Baselines` library [7]. DQN learns a "Quality" value (the Q-value) for each pair *(s,a)*.

Below, we depict the training performance in Figures 2 and 3. Fig. 2 shows the temporal evolution of the *Temporal Difference (TD) error*: the difference between the actual Q-value outputted by the agent and a estimate of this Q-value. We see that the TD error quickly converges to 0. Fig. 3 depicts the episode reward evolution with time. An episode reward is the sum of the rewards *r* produced during all the steps in a training episode. We observe that the average episode reward increases with time, stabilizing after 600 episodes.

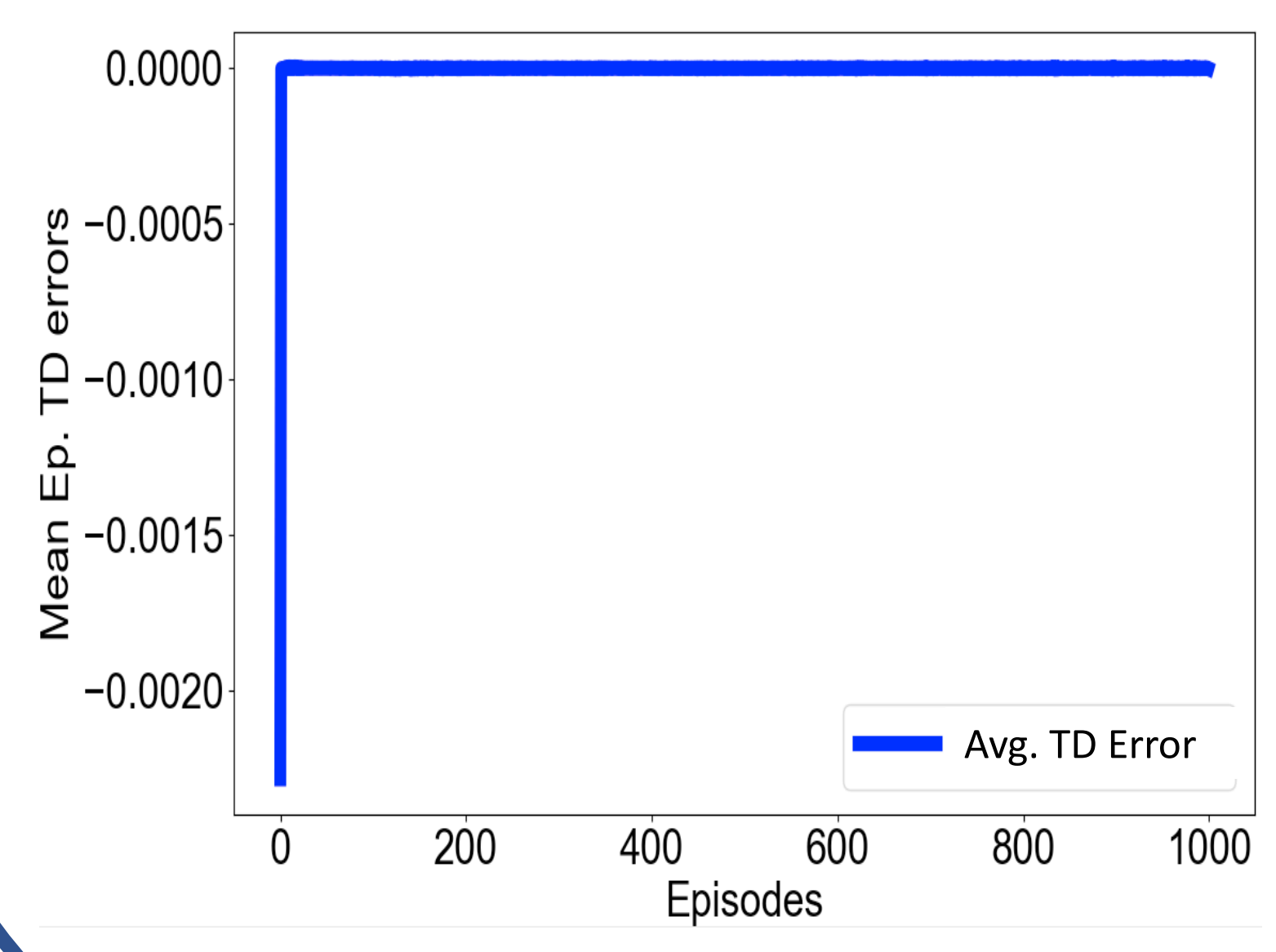

Fig. 2. TD error evolution

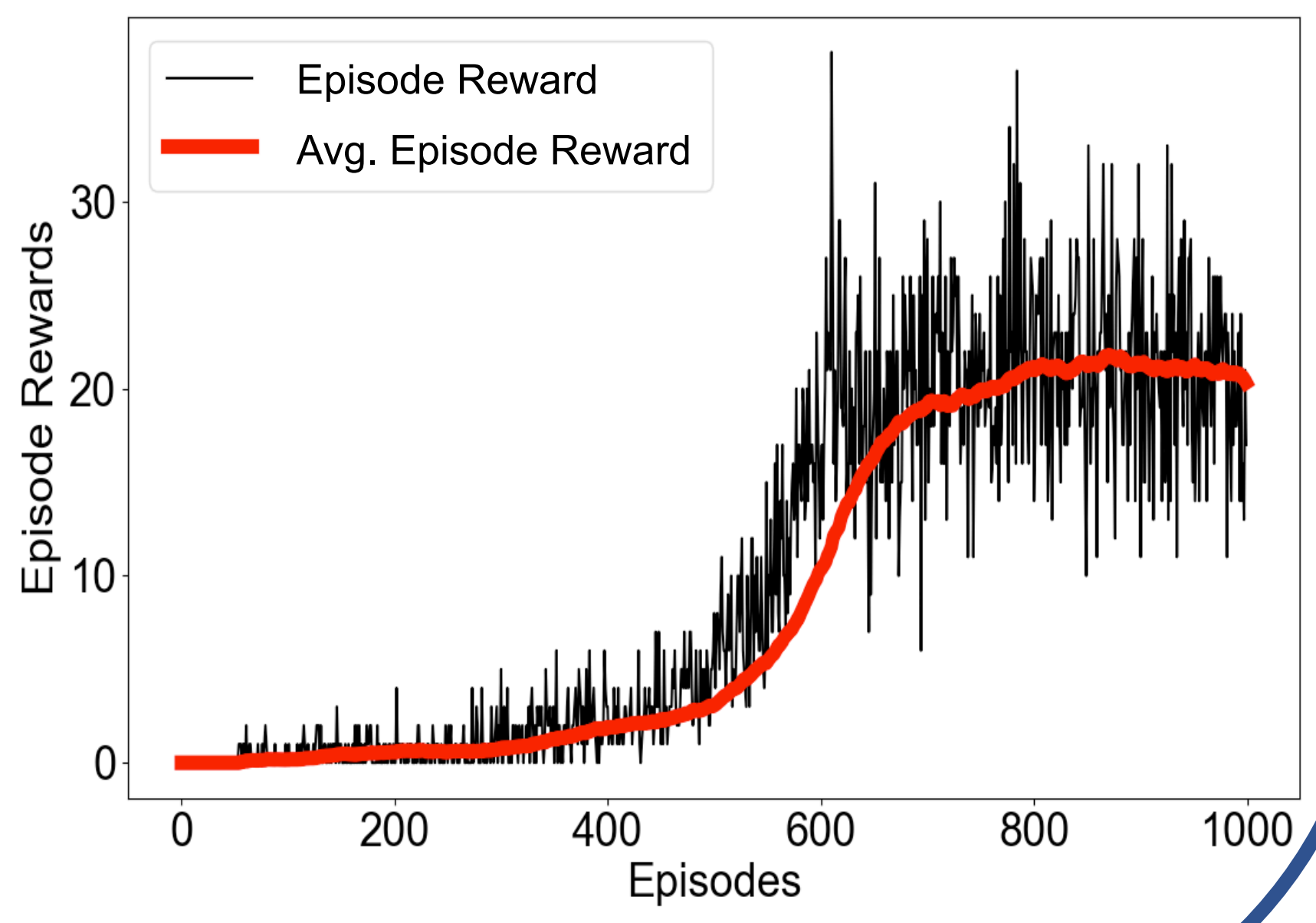

Fig. 3. Episode reward evolution

4. Results

In Fig. 4, we compare the policy learned by the DRL agent with two benchmarks:

- The ***Inf-cutoff-time*** *policy*: the cutoff time of the oldest link is set to infinity, then, when it is set up, we keep it till the second succeeds without taking care of the decoherence. This is the by-default policy considered by the State-of-the-Art [8],[9]. It represents a ***lower bound*** on the optimal policy.
- The ***opt-cutoff-time*** ***policy***: the optimal cutoff time of the oldest link is found by *brute force.* This process cannot scale up with larger problem instances. It trivially represents an upper **bound** on the policy learned by the DRL agent.

We test the policies by simulating the MDP process 1000 times. An "Episode" is a simulation instance. Each episode is divided in 10 000 steps.

We observe that the ***DRL agent*** clearly outperforms the **Inf-cutoff-time** policy and is close to the **opt-cutoff-time policy.**

The found cutoff times are:

- **DRL policy** : 146.0 steps
- **opt-cutoff-time policy:** 108 steps

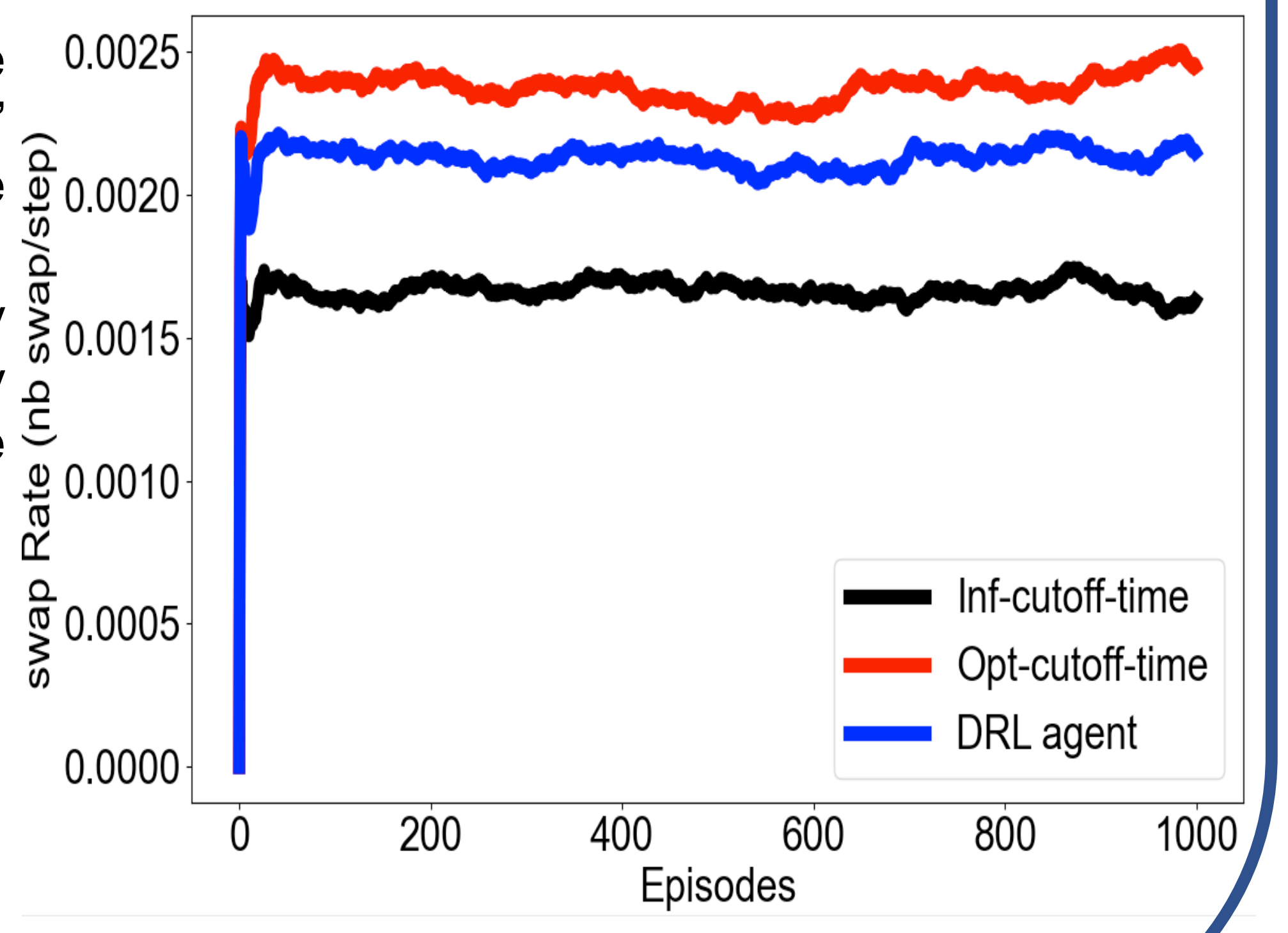

Fig. 4. DRL agent vs benchmarks

6. Conclusion

In this study, we explore if DRL can be used to learn policies maximizing the virtual link generation rate via entanglement swapping.

In the scenario where a model of the entanglement success probabilities is not known, we have obtained some first results pointing out that we can "discover" a close-to-optimal policy outperforming the the State-of-the-Art *Inf-cutoff-time* policy.

7. References

1. W. Kozlowski, et al. "Architectural principles for a quantum internet." *Internet Engineering Task Force, Internet-Draft draft-irtfqirg-principles-03* (2020).
2. A. Ortu, et al. "Storage of photonic time-bin qubits for up to 20 ms in a rare-earth doped crystal." *npj Quantum Information, vol., pp.* 8.1, 2022
3. S. Khatri, *Towards a General Framework for Practical Quantum Network Protocols,* Diss. Louisiana State University and Agricultural & Mechanical College, 2021.
4. Y. Bengio, *Learning deep architectures for AI*. Now Publishers, 2009.
5. V. Mnih, K. Kavukcuoglu *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, 2015
6. "Gym: Gym toolkit for creating reinforcement learning environments," 12/09/22. [Online]. Available: https://gym.openai.com
7. "Openai baselines: high-quality implementations of reinforcement learning algorithms," 12/09/22. [Online]. Available: https://github.com/openai/baselines
8. L. Gyongyosi, and S. Imre. "Advances in the quantum internet." *Communications of the ACM,* vol 65.8, pp 52-63, 2022.
9. S. Shi, and C. Qian. "Concurrent entanglement routing for quantum networks: Model and designs," Proc. SIGCOMM, 2020.

This work has been supported by the French government, through the UCA DS4H Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-17-EURE-0004.