# A Heuristic Approach for Resource Generation in a Quantum Network with Purifications

**Qu Tianchen**[1]

**Supervisor(s): Gayane Vardoyan**[1,2]**, Bethany Davies**[1,2]

[1]**EEMCS, Delft University of Technology, The Netherlands**

[2]**QuTech, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Qu Tianchen
Final project course: CSE3000 Research Project
Thesis committee: Gayane Vardoyan, Bethany Davies, Rihan Hai

## Abstract

Many quantum internet applications need access to multiple entangled links existing simultaneously. This requires the generation of multiple entangled links within a time window. Using single entangled link generation protocol, we model this task as a Markov decision process and propose a heuristic-based policy to generate them. This policy chooses different configuration parameters depending on the number of links in the register. Entanglement purification is also incorporated into this heuristic in different ways. To compare, we choose two baseline policies including a previously studied fixed configuration parameter policy. We show that this algorithm completes the generation up to six times faster than the baselines in our simulations and that using entanglement purification can further improve its performance.


## 1 Introduction

In the domain of quantum internet, an entangled link is two remote qubits that are entangled to each other which share a correlation stronger than classical bits [1][2]. Such links can be used as a vital resource in quantum internet [3]. The simultaneous availability of multiple entangled links is required for many applications that are not feasible otherwise in classical communications. This includes Blind Quantum Computation(BQC) [4] as well as extending entangled links via repeater chains [5]. However, the entanglement generation faces various challenges. Thus finding an efficient protocol for generating multiple entangled links with minimised time becomes a stepping stone toward achieving a full-fledged quantum internet [3].

The two main obstacles here are generation success probability and noise. In a typical heralded entanglement generation scheme which is commonly used today [6], the success probability of generating an entangled link can be as low as $5 \times 10^{-5}$ [7]. Even after heralding multiple generations, we still cannot guarantee that they ensure a successful generation. At the same time, the links we generate are never perfect. After the entangled link is established and stored in a quantum register, it also suffers from decoherence that arises from the interaction between the register and the environment [1]. This decreases the quality of the link over time until the link is too noisy to be kept useful. Thus, we must find a way of generating entangled links before the existing links are discarded due to low quality.

In an experimental setting for entanglement generation, several generation actions can be chosen by selecting different configuration parameters. This will affect two quantities: the successful generation probability, and the quality of the link. The goal is to simultaneously establish $n$ entangled links up to a certain quality threshold over the communication channel. Previously, this problem was characterised under the assumption that the instrument can generate entangled links with a fixed success probability and quality (fixed configuration parameters) [8]. And we are tempted to ask: can we improve the performance by using different configuration parameters cleverly?

In this work, we model the process of generating multiple entangled links as a Markov decision process (MDP). We propose a policy on this MDP with a trimmed action space and a heuristic that chooses actions based on estimating the time needed to generate future links. In our simulations, this can improve the performance up to six times compared to a fixed configuration parameter approach.

We also consider the effect of entanglement purification on entanglement generation. Purification is a process that can bring multiple low-fidelity entangled links to higher qualities with some probability. We have developed a way to incorporate two-to-one purification schemes into our heuristic. Specifically, we consider the Extreme Photon Loss (EPL-D) scheme [9][10] and the DEJMPS scheme [11] in our simulation results to show the improvement that purification can bring.

We first cover the background in Section 2. Here the model of interest is defined as well as several useful concepts throughout the paper. We also justify the assumptions we have made for the model. Next, we introduce our heuristic and the policy in Section 3. Here we explain the heuristic policy and how purification protocols can be added to our policy in different ways. The simulation results are illustrated in Section 4, accompanied by discussions and analysis of the results. Finally, the conclusion and future directions are in Section 5.

## 2 Preliminaries

### 2.1 Background

**Entangled link**

A quantum state can be represented by a unit vector in the Hilbert space $|\psi\rangle \in \mathcal{H}$ ; e.g, the state of a qubit is a vector with dimension $d = 2$. Quantum mechanics postulate demands that for two systems $A, B$ with corresponding Hilbert space $\mathcal{H}_A, \mathcal{H}_B$, the Hilbert space of the compound system would be their tensor product:$\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$. A perfect entangled link $|\Psi\rangle$ is a maximally entangled state in a two qubits system $\mathcal{H}_A \otimes \mathcal{H}_B$ which can be written as

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|\psi_0\rangle|\phi_0\rangle + |\psi_1\rangle|\phi_1\rangle) \tag{1}$$

for some orthonormal basis $\{|\psi_0\rangle, |\psi_1\rangle\} \in \mathcal{H}_A, \{|\phi_0\rangle, |\phi_1\rangle\} \in \mathcal{H}_B$. On the other hand, when we are uncertain about the specific state a system is in, we can express the state in a more general way as a classical probabilistic mixture of states known as the mixed state:

$$\rho = \sum_i p_i \left|\psi_i\right\rangle\!\left\langle\psi_i\right|, \quad \sum_i p_i = 1, p_i \geq 0. \tag{2}$$

In reality, the entangled links we generate are always noisy and take the form of a certain mixed state. We define the quality of a link as the 'closeness' between the perfect entanglement state and the link known as the fidelity between them:

$$F(|\Psi\rangle\langle\Psi|, \rho) = \langle\Psi|\rho|\Psi\rangle \in [0, 1]. \tag{3}$$

If not specified otherwise, fidelity in this paper refers to the fidelity with the perfect entanglement state.

**Success probability and fidelity trade-off**

At each generation attempt, we conduct multiple entanglement generations that in total have a certain success probability $p$ to generate a link with fidelity $F$. All the attempts will cost a constant amount of time $\Delta t$. In a typical heralded entanglement scheme, we can continuously tune the success probability $p$ and fidelity $F$ with a linear trade-off between them:

$$F = 1 - \lambda p, \quad p, F \in (0, 1). \tag{4}$$

The derivation of this trade-off relation and the formula for $\lambda$ are specified in Appendix A. A typical $\lambda$ can take a value between 0.1 to five (e.g., in [6] we have $\lambda = 2$). For the rest of the paper, we assume that all our generation attempts follow this trade-off relation. However, other possible relations can be derived. We show a different trade-off relation based on interpolation in Appendix B.

**Noise model and fidelity bins**

After an entangled link is established and stored in the quantum register, the noise arising from its interaction with the environment is generally complicated. The exact noise will heavily depend on the actual realisation and calibration of the platform. However, it has been shown that if we assume the noise to be Markovian, then we can always use the depolarization noise model as a worst-case model to describe the system [12]. Depolarization noise channel can be seen as Pauli operators randomly acting on the state in a Poisson process, characterized by a single positive decay parameter $\Gamma'$:

$$\mathcal{E}(\rho) = e^{-\Gamma' t}\rho + (1 - e^{-\Gamma' t})\frac{I}{4}. \tag{5}$$

Together with the fidelity definition in (3), we can see that the fidelity decays exponentially towards $\frac{1}{4}$ (the fidelity of a maximally mixed state):

$$F(t) = \left(F_0 - \frac{1}{4}\right) e^{-\Gamma' t} + \frac{1}{4}. \tag{6}$$

Given that each timestep is constant, we can discretize the depolarization process based on each subsequent generation attempt after a link is established:

$$F(n) = \left(F_0 - \frac{1}{4}\right) e^{-\Gamma n} + \frac{1}{4}. \tag{7}$$

Here we define the discrete decay rate $\Gamma = \Gamma' \Delta t$ which is referred to below as the decay rate. This tells us the fidelity of a state after $n$ generation attempts given its initial state fidelity $F_0$.

Low-fidelity entangled links have been subjected to a lot of noise. For example, a state with fidelity lower than 0.5 can be a separable state that does not have any entanglement. Therefore, we require that the fidelities of the links we want to be higher than a certain threshold $F_{th}$. Any link lower than this threshold is not useful anymore and is discarded. Due to the depolarizing noise mentioned above, an entangled link in the register only has a limited amount of time before its fidelity goes below the threshold. This allows us to perform a limited amount of entanglement generations.

Suppose the entangled link that we have generated has initial fidelity $F$. Using (7), we can derive how many timesteps $n$ it took before the link decays below the threshold $F_{th}$ and is discarded:

$$(F - \frac{1}{4})e^{-\Gamma n} + \frac{1}{4} \geq F_{th} \Rightarrow n < \frac{1}{\Gamma} \ln \frac{F - \frac{1}{4}}{F_{th} - \frac{1}{4}}. \tag{8}$$

We thus define the maximum number of timesteps a link can have before its fidelity drops lower than $F_{th}$ as its fidelity bin $n(F)$:

$$n(F) = \left\lfloor \frac{1}{\Gamma} \ln \frac{F_0 - \frac{1}{4}}{F_{th} - \frac{1}{4}} \right\rfloor. \tag{9}$$

A visualization for the fidelity bin can be found in Figure 1

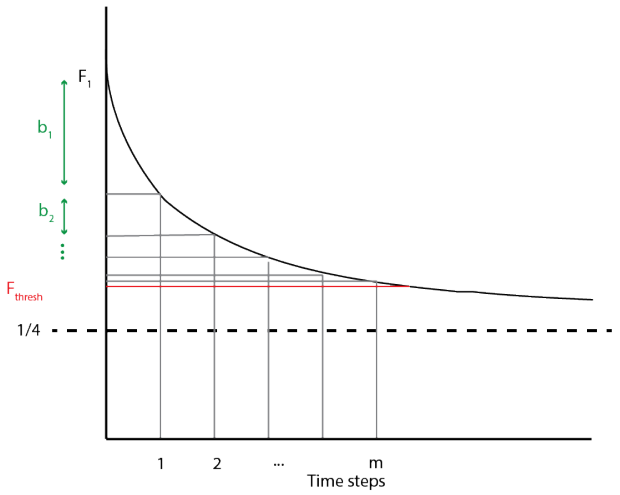


Figure 1: Each $b_i$ here is a fidelity bin. We can see that the fidelity ranges for different fidelity bins are different since fidelity decays exponentially with depolarization noise. Adapted from B. Davies with permission.

This fidelity bin can be interpreted as the lifespan of a link, which will decrease by one after each generation attempt. It also signals how many entanglement generations we can do before this link is discarded. If a link is in fidelity bin zero, this means that once the next generation attempt is finished, its fidelity will be below the threshold and being discarded.

The binning process also assigns two links with different fidelity to be within the same bin. This means that their fidelity will not make a difference to our system since the generation attempts do not cost an arbitrarily small amount of time. We can thus describe the state of a link by its corresponding fidelity bin $n(F)$.

## 2.2 A Markov decision model

In this section, we define the system formally as an MDP based on the observations above. To summarize, our system can be defined by the following parameters:

- $s$, the number of entangled links we want to generate;
- $F_{th}$, the fidelity threshold for the entangled links we want;
- $\Gamma$, the decay rate for the quantum register;

- $\{(p, F) \in (0, 1)^2\}$, the entanglement generation scheme.

An MDP is a 6-tuple $(\mathcal{S}, \mathcal{A}, P_a(s, s'), s_{initial}, S_{terminal}, R_a)$ where

- $\mathcal{S}$ is the state space which consists of all possible states in the system. Here we can define it as

$$\mathcal{S} = \{T : |T| \leq s, \forall n_i \in T, 0 \leq n_i \leq n(1)\}, \quad (10)$$

  where the state $T$ here is the set of fidelity bins for all existing entangled links in the register. Note that the state size will not exceed the total number of required links $s$ and the fidelity bin is between zero and the maximal fidelity bin $n(1)$ (the lifespan for a perfect entangled link);

- $\mathcal{A}$ is the action space, representing all possible actions we can do within a timestep. We directly call the fidelity of the link that an action can create as the fidelity of an action. The Heralded entanglement generation scheme will give us a continuous action space with parameter $\lambda$:

$$\mathcal{A} = \{(p_a, F_a) : p_a, F_a \in (0, 1), F_a = 1 - \lambda p_a\}. \quad (11)$$

  In experiments we might also want to restrict ourselves to a smaller action space with a restricted $p \in (a, b), \quad 0 \leq a < b \leq 1$:

$$\mathcal{A} = \{(p_a, F_a) : p \in (a, b), F_a \in (0, 1), F_a = 1 - \lambda p_a\}; \quad (12)$$

- $P_a(s, s')$ is the transition function representing the probability distribution of the next state $s'$ given the previous state $s$ and action $a$. In our system, this transition characterises the following steps:
  - A decay process where each fidelity bin in the state will degrade by one. Any link with fidelity bin 0 will be discarded, e.g., $\{0, 2, 3, 5\} \to \{1, 2, 4\}$,
  - Based on the action $(p_a, F_a)$ we have chosen, a new link is added with probability $p_a$: if the generation attempt is successful, a link with fidelity bin $n(F_a)$ will be added to the register, e.g., (success)$\{1, 2, 4\} \to \{1, 2, 4, n(F_a)\}$;
- $s_{initial}$ is the initial state of the MDP with no links in the quantum register $s_{initial} = \emptyset$;
- $S_{terminal}$ is the set of terminal states where the MDP ends with $s$ links in the register $S_{terminal} = \{T \in \mathcal{S} : |T| = s\}$.

Since we use a heuristic approach in this paper, we do not need to define the reward function $R_a$.

### 2.3 Purification

Purification, also called entanglement distillation, is a process that transforms multiple noisy entangled links into links with higher fidelity via local operation and classical communication (LOCC) [1][13]. In this paper, we utilize two well-studied purification schemes. Both of them purify two identical copies into a single link with some success probability. Failing the purification process will result in the loss of all input states. They also require only one round of classical communication and the time for the purification process is negligible in our model.

The single-click protocol for entanglement generation that we specified in Appendix A generates links with fidelity $F_0$ in state $\rho = F_0 |\Psi^+\rangle\langle\Psi^+| + (1 - F_0) |00\rangle\langle 00|$ where the Bell state $|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ is a maximally entangled state. This can be transformed into R-states with equal fidelities by applying $X$ gate locally on both sides:

$$\rho \xrightarrow{X_1 \otimes X_2} F_0 \left|\Psi^+\right\rangle\left\langle\Psi^+\right| + (1 - F_0) |11\rangle\langle 11|. \quad (13)$$

EPL-D purification scheme [9] takes in such R-states [10] and returns a perfect entangled link $|\Psi^+\rangle\langle\Psi^+|$ with probability

$$p_{epl} = \frac{F_0^2}{2}. \quad (14)$$

Notice that the EPL-D scheme only works for R states and we can only apply it immediately after the links are generated.

To purify links in the register that have gone through stochastic noise, we can first transform them into a Werner state $\rho' = \frac{4F_0 - 1}{3} |\Psi^+\rangle\langle\Psi^+| + \frac{1 - F_0}{3} I$ with equal fidelity $F_0$ via local operations. Then we can apply DEJMPS purification protocol [11]. This purification process has a success probability

$$p_D = \frac{8F_0^2 - 4F_0 + 5}{9} \quad (15)$$

to generate a link with fidelity

$$F_D = \frac{10F_0^2 - 2F_0 + 1}{8F_0^2 - 4F_0 + 5}. \quad (16)$$

The exact steps of carrying out the two purification schemes are specified in Appendix C.

## 3 Heuristic and Policy

### 3.1 Action space reduction

Notice that the definition of fidelity bins in (9) contains a floor function. For two different actions $(p_a, F_a), (p_b, F_b)$ that are rounded down to the same fidelity bin $n(F_a) = n(F_b)$, they share the same lifespan and only differ in their corresponding success probabilities. Therefore, within the same fidelity bin, the action with maximum fidelity will be strictly better than other actions.

Given the action space in (12), we can define the optimal action within some fidelity bin $n$ as:

$$(p_n, F_n)_{optimal} = \underset{a}{\text{argmax}}\{p_a : (p_a, F_a) \in \mathcal{A}, n(F_a) = n\}. \quad (17)$$

Since the trade-off is linear, maximizing the probability means that this optimal action produces a link with minimal fidelity: the lower bound fidelity to have fidelity bin $n$. This implies that after it decays to fidelity bin 0, its fidelity will be exactly $F_{th}$. Note that we can only define an optimal action if the corresponding fidelity bin is achievable in the original action space: $n(F_{min}) \leq n \leq n(F_{max})$.

We can thus restrict the whole action space to a finite optimal action subset

$$\mathcal{A}_{optimal} = \{(p_n, F_n)_{optimal} : n(F_{min}) \leq n \leq n(F_{max})\}. \quad (18)$$

We can see that the discrete nature of the batched attempts not only gives a finite state space but a discrete optimal action space as well. This leaves only a discrete set of fidelities desirable to generate. Such restriction simplifies the action space and reduces the complexity of the MDP dynamics. It can also remove the unnecessary fidelity requirement within the same fidelity bin in the heuristic policy below.

### 3.2 Look-ahead heuristic and policy

**Look ahead**

One difficulty in finding a desirable policy for the MDP is the big state space. As the size of the state space grows exponentially with the number of links.[1]

To avoid complicated analysis of the state space, we adopt a heuristic called look-ahead which only considers the number of links in the current state. Here we look ahead to derive the current action based on future expectations instead of looking back on different fidelity bins of the links in the current state.

The basic observation here is that the current link we want to generate needs to survive when all necessary links have been generated. This is a reasonable requirement since it would be pointless if we tried to generate a link that would be discarded by the end of the generation. We denote the fidelity bin of the link we want to generate as $n_{current}$ and the estimated time ahead to generate the rest of all the links as $t_{ahead}$. This heuristic can be expressed as $n_{current}$ needs to be proportional to $t_{ahead}$:

$$n_{current} \propto t_{ahead}. \tag{19}$$

Qualitatively speaking, the link wants a long lifespan at the start since it takes more time to generate other links in the future. On the other hand, the last few links that we want to generate can have lower fidelity bins (thus higher success probabilities) to cost less generation time.

**Look-ahead policy**

Based on the heuristic above, we can derive the policy in a backward manner with the algorithm below.[2] The three parameters here are the number of links we need to generate $s$, the optimal action space we derived in action space reduction $\mathcal{A}_{optimal}$, and the proportion factor $\alpha$ for the relation in (19): $n_{current} = \alpha \cdot t_{ahead}$.

This new proportion factor $\alpha$ can be interpreted as how 'secure' we want the current link to be. Policy with a high $\alpha$ value tends to generate links with higher fidelity bins. Since the time we need to generate the links ahead is a distribution associated with our estimation for the time ahead, higher $\alpha$ grants the current link a bigger chance of surviving till the last generation. On the other hand, small $\alpha$ generates the current link with a higher success probability and thus costs less average time to generate it.

[1] We show this in Appendix D

[2] In general, we cannot assume that there exists an action that has the fidelity bin $\lfloor \alpha \cdot t_{ahead} \rfloor\}$. In this case, we can assign the action with the closest fidelity bin to the required bin. An example is shown in Figure 4 where we hit the lower and upper bound of the action space.

The returned 'policy' $\pi' : \mathbb{N} \to \mathcal{A}_{optimal}$ will tell us the action $\pi'(i)$ to carry out for generating the $i_{th}$ link in the register. In other words, we can write the whole policy $\pi : \mathcal{S} \to \mathcal{A}_{optimal}$ as $\pi(T) = \pi'(|T| + 1)$, $|T|$ is the number of links for a register state $T$. We also denote the estimated time ahead after the $i_{th}$ link is generated as $t_{ahead}(i)$

**Algorithm 1** Heuristic Policy$(s, \mathcal{A}_{optimal}, \alpha)$

$t_{ahead}(s) \leftarrow 0$ ▷ no need to generate future links
$\pi'(s) \leftarrow \underset{a}{\text{argmax}}\{p_a : (p_a, F_a) \in \mathcal{A}_{optimal}, F_a \geq F_{th}\}$
**for** $i = \{s-1, s-2, \ldots, 2, 1\}$ **do**
  $t_{ahead}(i) \leftarrow t_{ahead}(i+1) + \frac{1}{p_{\pi(i+1)}}$ ▷ increase $t_{ahead}$
  $\pi'(i) \leftarrow \text{arg}_a\{n(F_a) = \lfloor \alpha \cdot t_{ahead} \rfloor\}$ ▷ the heuristic
**return** $\pi'$

We start from generating the last link when we already have $s-1$ link in the register. The optimal action here would be the action with maximum success probability as long as its fidelity is above $F_{th}$.

To see this, suppose the optimal policy carries out an action with non-maximum success probability $p_a < p_{max}$ when there are $s-1$ links in the register. If it succeeds, the generation ends immediately just like succeeding with maximum success probability action, and in both cases the future generation time is zero. If it fails, it will end up with the same state as the maximum success probability action fails. From the Markovian property of the system, the same state will give the same future generation time. However, $p_a$ has less probability of success which will end the generation immediately. Therefore, this action takes a longer time to finish the generation on average.

Thus we assign the policy for generating the last link $\pi'(s) = \underset{a}{\text{argmax}}\{p_a : (p_a, F_a) \in \mathcal{A}_{optimal}, F_a \geq F_{th}\}$. Since we do not need any future attempts once this link is generated, the estimated time ahead when this link is generated is $t_{ahead}(s) = 0$.

Next, the time to generate a new link for an action with probability $p$ is a geometric distribution $P(T_p = a) = (1-p)^a p$. The average time for it is $\mathbb{E}[T_p] = \frac{1}{p}$. If the future policies $\pi'(i+1), \pi'(i+2), \ldots, \pi'(s)$ are settled, we can calculate the average time for carrying them out as

$$t_{ahead}(i) = \sum_{k=i+1}^{s} \frac{1}{p_{\pi'(k)}}. \tag{20}$$

We then use the heuristic $\pi'(i) = \text{arg}_a\{n(F_a) = \lfloor \alpha \cdot t_{ahead}(i) \rfloor\}$ to assign the action for the current policy. The floor function here removes unnecessary fidelity requirements. In this way, each previous policy can be derived iteratively.

We can see that both $t_{ahead}(i)$ and $\pi'(i)$ do not depend on the fidelity bin of the fidelity bins of the links in the current register. This allows them to be precomputed before the actual generation starts.

Also, note that $t_{ahead}(i)$ is the average time for successfully generating $s-i$ links in the future. It does not guarantee

that all of these links will survive in the register by the end of the generation.

### 3.3 Incorporating purification

Both purification protocols we mentioned in the previous section require two identical copies of entangled links. Therefore, we need to modify our actions slightly. We assume that the success of each action will now create two links that are ready for purification afterwards. Based on this change, we give two different ways of including purification

**Purification on arrival**

For each successful generation attempt $(p_a, F_a) \in \mathcal{A}$, we can carry out a purification process right away. This allows us to define a new action for each old action:

$$(p_a, F_a) \xrightarrow{purify} (p_a p_p(F_a), F_p(F_a)). \tag{21}$$

This new action will succeed if both the generation and purification process are successful and will produce one link in that case. Here $p_p(F), F_p(F)$ are the functions representing the purification success probability and resulting fidelity with input fidelity $F$. In this way, a purification scheme will create a new set of actions $\mathcal{A}_p$ that has the same size as the original action space $\mathcal{A}_{optimal}$.

However, we cannot directly apply the previous algorithm to the system with extended action space $\mathcal{A}'$. This is because actions that produce one and two links are not directly comparable.

Generating more links per action reduces the time for future generation attempts. Suppose we have $i-1$ links in memory, an action that generates two links has the future generation time $t_{ahead}(i+1)$. An action with purification only gives one link, and it has the future time estimation $t_{ahead}(i)$ in this case. The heuristic thus gives us two actions – an action $a_{nop}$ without purification and an action $a_p$ with purification:

$$a_{nop} = \arg_{a \in \mathcal{A}_{optimal}} \{n(F_a) = \lfloor \alpha \cdot t_{ahead}(i+1) \rfloor\}, \tag{22}$$

$$a_p = \arg_{a' \in \mathcal{A}_p} \{n(F_a) = \lfloor \alpha \cdot t_{ahead}(i) \rfloor\}. \tag{23}$$

We then choose the action that will give a smaller time ahead increase for $t_{ahead}(i-1)$: $\pi'(i) = \text{argmin}_{a_p, a_{nop}} \left( \frac{1}{p_{a_{nop}}} + t_{ahead}(i+1), \frac{1}{p_{a_p}} + t_{ahead}(i) \right)$. This reduces the generation time for generating the current link. With (20), this comparison can also be written as

$$\frac{1}{p_{a_{nop}}} \sim \frac{1}{p_{a_p}} + \frac{1}{p_{\pi'(i+1)}}. \tag{24}$$

The overall policy algorithm is:

In our simulation results, we purify under the EPL-D scheme. The overall success probability of the new action follows from (4) and (14):

$$p'_a = p_a p_p(F_a) = \frac{p_a F_a^2}{2} = \frac{F_a^2 - F_a^3}{2\lambda}. \tag{25}$$

Since this will produce a perfect link with fidelity 1 regardless of the input states, we choose the action with fidelity $F_a = \frac{2}{3}$ to maximise the success probability.[3] Therefore we

[3]If in experimental setting $F_a = \frac{2}{3}$ is not achievable, we can always find another value that maximises the probability

**Algorithm 2** Policy with Purification$(s, \mathcal{A}_{optimal}, \alpha)$

$t_{ahead}(s) \leftarrow 0$
$\pi'(s) \leftarrow \underset{a}{\text{argmax}}\{p_a : (p_a, F_a) \in \mathcal{A}_{optimal}, F_a \geq F_{th}\}$
**for** $i = \{s-1, s-2, \ldots, 2, 1\}$ **do**
  $t_{ahead}(i) \leftarrow t_{ahead}(i+1) + \frac{1}{p_{\pi(i+1)}}$
  $a_{nop} \leftarrow \arg_{a \in \mathcal{A}_{optimal}} \{n(F_a) = \lfloor \alpha \cdot t_{ahead}(i+1) \rfloor\}$
  $a_p \leftarrow \arg_{a' \in \mathcal{A}_p} \{n(F_a) = \lfloor \alpha \cdot t_{ahead}(i) \rfloor\}$
  **if** $\frac{1}{p_{a_{nop}}} < \frac{1}{p_{a_p}} + \frac{1}{p_{\pi'(i+1)}}$ **then**
    $\pi'(i) \leftarrow a_{nop}$
  **else**
    $\pi'(i) \leftarrow a_p$
    **return** $\pi'$

can view EPL-D as adding a single new action in the action space $\mathcal{A}_p = \{(p'_{max}, 1)\}$ where $p'_{max}$ is the maximum success probability for all actions with purification. Since we only have one action for purification that produces perfect link, we only need to evaluate $a_{nop}$ here and compare with $(p'_{max}, 1)$ using (24).

**Purification in the end**

We can also incorporate purification as a modification on the transition function of the MDP: Whenever two links reach fidelity threshold $F_{th}$ and are about to expire, we automatically purify them during that timestep. In this case, purification strictly improves the performance since the two links would expire otherwise. We have also tried different thresholds for purifying two states, but simulation results showed that $F_{th}$ gives the best performance in most cases. However, this approach gives us less insight into the dynamics of the system and we use it as a comparison to the above case.

Since the links in the register with threshold fidelity suffer from depolarizing noise, we assume them to be Werner states $\rho = \frac{4F_{th}-1}{3} |\Psi^+\rangle\langle\Psi^+| + \frac{1-F_{th}}{3} I$ and adopt DEJMPS protocol for the two links.

## 4 Results

### 4.1 Heuristic policy without purification

We first illustrate the properties of the heuristic without considering purification. We compare the heuristic's performance to the random action and fixed action policies, then we show how changing the number of links we want changes the performance of the heuristic.

With an action space where $p \in (a, b)$, the two baselines that we choose here are

- random action: At each timestep, the policy chooses randomly between two extreme actions (with maximum and minimum success probability) in the action space $\mathcal{A}$,
- fixed action: At each timestep, the policy always chooses a fixed action with the mean success probability $\frac{a+b}{2}$.

In the case of generating four links, the heuristic costs less than half the time to finish the generation compared to the two baselines as shown in Figure 2. We can see that in this specific

case, the performance has a sweet spot when $\alpha$ is at around 1.1-1.26. This can be viewed as a balance between generating high-quality links and generating links quickly. We also notice that the policy varies discretely with changing $\alpha$. This is due to our restricted action space $\mathcal{A}_{optimal}$ being discrete. However, applying the whole action space $\mathcal{A}$ will also have the same discrete jumps as what we have seen in Figure 2. This jump arises because the fidelities of actions are assigned in discrete fidelity bins in our model.

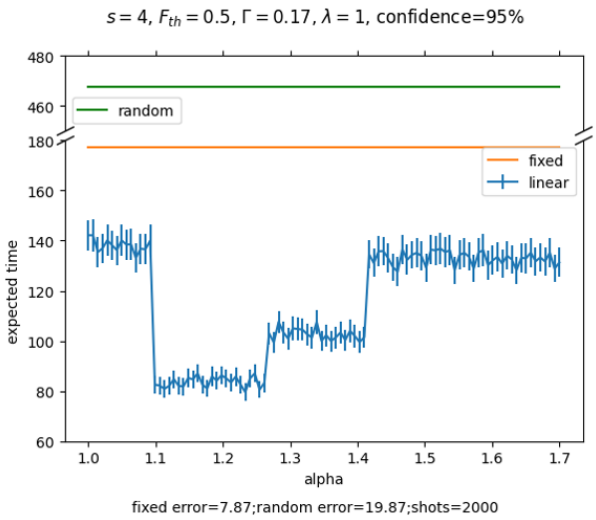


Figure 2: This is how the performance for generating four links changes when we tune the proportional factor $\alpha$. It is possible to tune $\alpha$ below 1, but that in general performs poorly. Tuning $\alpha$ too low will also cause the policy not being able to produce $s$ links at all. The fixed error and random error at the bottom are the errors with a 95% confidence interval for the two baseline policies.

The heuristic works better for a less near-term network where more links are required and the system has a lower decay rate[4]. We show the result for generating 20 links with a lower decay rate in Figure 3. Here the heuristic is six times faster than the fixed action policy and 45 times faster than the random action policy. This shows that random action and fixed action are far from optimal in the regime of large link numbers $s$. At the same time, the action space and state space grow larger accordingly. Thus searching for an optimal policy via policy iteration or machine learning would be computationally costly. Therefore the heuristic can be particularly helpful in this case.

We also witness the performance in Figure 3 with a very sharp increase when $\alpha$ is small. We observe that the generation time follows a short-tailed distribution regardless of the policy that we choose. Since in our heuristic $\lfloor \alpha \cdot t_{ahead} \rfloor$ can be seen as a cut-off that guarantees the link will survive if the generation time for the rest of the links does not exceed the cut-off, when we vary $\alpha$ too low, the probability of survival drops sharply. This also shows that optimizing $\alpha$ is a very important step for this heuristic.

[4]A lower decay rate implies a register with less decoherence noise or the time for each generation attempt is shorter.

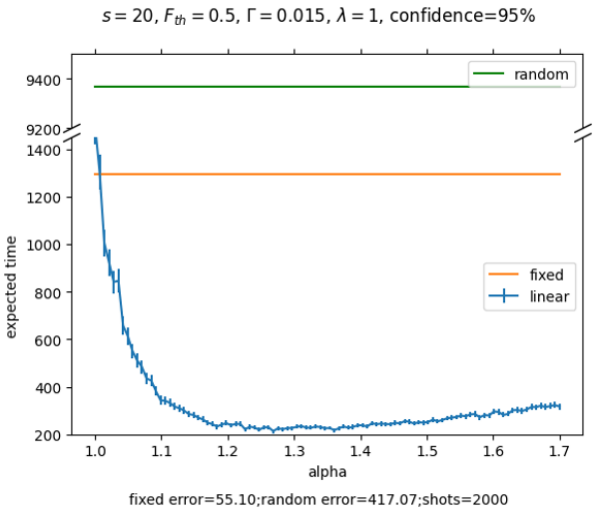


Figure 3: Here we show the case for generating 20 links. We can see that the expected time ratio gets widened compared to the 4 links' case. The heuristic policy can be up to six times faster than the fixed action policy and 45 times faster than the random action policy. However, when $\alpha = 1$, the heuristic policy would perform even worse than the fixed action policy. The fixed error and random error at the bottom are the errors with a 95% confidence interval for the two baseline policies.

The performance of the heuristic when we vary the number of links $s$ is shown in Figure 4. The blue line shows the performance and the red line shows the bin $n(F_{\pi'(1)})$ for the new action that is appended for each policy. Since $t_{ahead}$ is estimated backwards, every time we want to generate an additional link, we only append a new action at the beginning of the previous policy. For example, if the policy for $s = 2$ is $[a_2, a_1]$, the policy for $s = 3$ would be $[a_3, a_2, a_1]$ for some action $a_3$ at the beginning. Therefore, the policy for generating $s$ links can be seen as the partial red line on the left side of $x = s$. We see that as $t_{ahead}$ grows, no action can offer the corresponding fidelity bins at some point, and the action would be fixed to the one with the highest fidelity bin afterwards. This appears to be the start of bad performance for the heuristic. Any increase in the number of links we want will result in a fixed action and the expected generation time will grow rapidly afterwards. The performance would then appear similar to the fixed action case [8] where the expected time is bounded exponentially in the worst case. This may be the limit for the heuristic. On the other hand, it may also indicate the limit of the system configuration itself: we are hitting some hardware limitations for generating more than a certain amount of links efficiently.

## 4.2 Heuristic policy with purification

To see the further improvement from purification, we compare the two different purification approaches with the original heuristic policy. Figure 5 shows the performance for all the different approaches with varying numbers of links to generate. Here the performance is optimized over $\alpha$.

The first thing we realize during our research is that purifi-

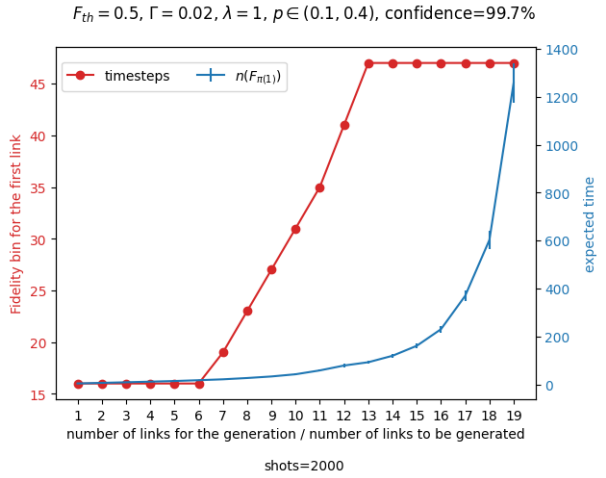


Figure 4: For the blue line, the figure shows the performance for generating different numbers of links. For the red line, it shows the action we carry out given the number of links we still need to generate. Since the policy $\pi'$ only depends on the number of links, the policy for generating $s$ links is the partial red line on the left side of $x = s$. When the number of links $s$ is high enough(in the figure its about $s = 13$), the policy has to choose the highest fidelity bin action to start with constantly if we increase $s$ even higher. At this point, the expected time of the heuristic will start to increase significantly.

cation depends on the actual fidelity of the link. This means that the fidelity bin itself is no longer enough to characterize the system dynamics. The links need to have higher fidelities in order to have a higher success probability during purification. For DEJMPS protocol, the fidelity jump also depends on the input fidelities from (16). This means that purification works better when we have higher threshold $F_{th}$ and lower decay rate respectively. This makes purification more useful for less near-term quantum networks. In Figure 5, we set the threshold to be 0.9.

According to Figure 5, purification on arrival does not perform as well as purification in the end. Specificly when $s = 16$, it is even worse than the baseline without purification. To our surprise, a small modification for the purification on arrival approach changes its performance. We can see that the modified purification outperforms both approaches steadily and has the least expected time.

The modification is on (24) and we show the whole algorithm in Appendix E. We assume that the next action after the purification action is similar to the purification action $\pi'(i+1) = a_p$. Then we only need to compare $2p_{a_{nop}} \sim p_{a_p}$ and choose the action that is bigger in the comparison:

$$\pi'(i) = \begin{cases} a_{nop} & 2p_{a_{nop}} > p_{a_p} \\ a_p & 2p_{a_{nop}} \leq p_{a_p} \end{cases}. \tag{26}$$

The new policy afterwards removes occasional purification actions when there are many links in the register. While in the earlier stages of the generation, the policy is still very similar to the policy without modification. The improvement might be because policies with more purification actions take more steps to finish, and this leads to a higher variance for the generation time. When multiple links are in the register, higer variance makes it easier to lose the existing links.

The expected time ratios between no purification and the three purification approachs are shown in Figure 6. When generating odd numbers of links, purification gives a higher improvement ratio. This is most probably because actions without purification only generate two links, and purifying two links to one link would help obtain the additional one link of the $2n + 1$ links we want. For an even number of links, as the number of links increases, the expected time for the baseline can still be 50.4% higher than the expected time for the modified purification approach.

However, the two purification-on-arrival approaches don't always perform well. We have seen cases where they can have the same policy as the baseline: the heuristic simply regards the purification actions as not worthy to be used here. This gives zero improvement since purification is completely overlooked by the heuristic. In this case, purification in the end would be a better choice. The fact that it strictly gives improvement under any configuration makes it more robust compared to the other two approaches. Another potential advantage of this approach is that fidelity drops exponentially under depolarizing noise and makes the fidelity decrease to slow down over time. We can see that the fidelity bins become smaller when they are close to the threshold in Figure 1. Therefore, carrying out purification at the threshold grants bigger fidelity bin leaps.

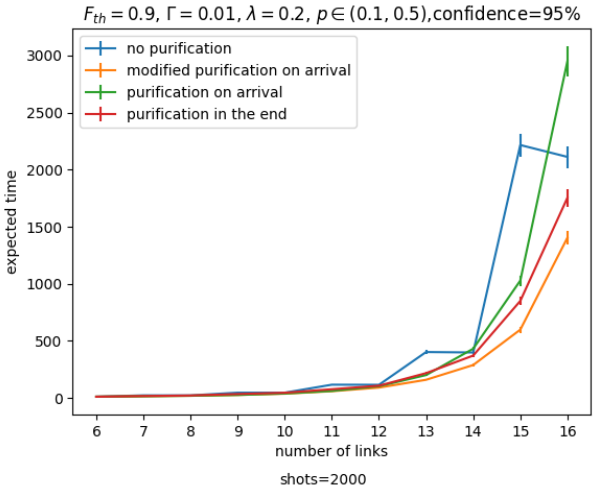


Figure 5: For each data point we optimize a different $\alpha \in (0.9, 1.7)$ for the shortest expected time. The baseline is less stable since each action will create two links, this means that the performance would be stepwise for every two more links to generate. On the other hand, the curve for policies with purification are much smoother.

## 5 Conclusion

In summary, we have managed to model the entanglement generation process as an MDP and proposed a heuristic-based policy for it. The heuristic demands the lifespan of a link to be proportional to the estimated future generation time. As shown by the simulation results, the heuristic policy is able to

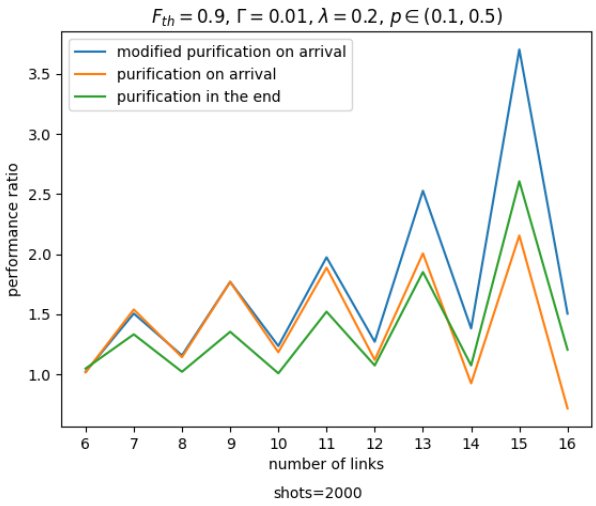


Figure 6: This is the performance ratio between no purification and the three purification approaches. The zig-zag line is a result of the two link platform for the baseline performance. We can see this in Figure 5

.

finish generation significantly faster than fixed action policy, which was investigated in previous work [8]. This improvement is also robust under different parameters ranging from a near-term quantum network generating four entangled links to the task of generating 20 links. Between different parameters, we find the heuristic to perform particularly well when a larger number of links needs to be generated. On top of that, we propose three different ways of incorporating different purification schemes(EPL-D, DEJMPS) and evaluate their improvements to the original heuristic.

### Further directions

So far, our policy is agnostic to the fidelity bins for the links in the register. Although this has helped us avoid the complexity of the state space, it also makes the heuristic too granular to capture specific cases. This includes situations where previous generations have taken too much time. Many links in the register are then about to expire. We would prefer to start generating high-fidelity links directly and 'discard' these links with low-fidelity bin. A possible alternative policy can take such a current state into account when we are estimating the time ahead.

With regard to purification, our model and heuristic have only utilized some specific protocols. Purification is a vast field in quantum communication and quantum internet that can be applied in various ways. We have only considered the case of purifying two identical copies. The next possible step can be using protocols which purify two links with different fidelities or purify with more than two links. We also assume the purification process takes negligible time compared to the heralded generation. But this is not true for general purification where multiple rounds of communication are needed and we cannot execute them within a single timestep.

We also assume the trade-off function to be linear in our work. As the success probability increases, the linear approximation would be less accurate for heralded generation and a full investigation for this accuracy is needed. For example, in the case of generating two links for heralded entanglement generation, a nonlinear trade-off function would be more suitable for the analysis, the performance under such trade-off functions remains to be seen.

## 6 Responsible Research

Since the research that we have conducted in this paper intends to broaden our knowledge in the public domain, it is important to show that the results and conclusions that we have made are reproducible and reliable.

For the reproducibility of our simulations in Section 4, we have explained all our assumptions in the main body of this paper and we have published our code on Github [14] with the fully open-source MIT License to illustrate the simulation process. This also includes original figures that we used in the paper as well as other figures that are not adopted in this paper for cross-referencing. This allows researchers to reproduce and develop upon our results for future uses. The code includes simulations and plot generation for all the figures in this paper (except the figure that is adapted from other people with permission) and allows the researchers to generate different plots under different parameters. The researchers can also find various MDPs in the repository where they can test their own policies. However, it is important to note that we simulate a probabilistic MDP and future researchers may not achieve the exact same results. Therefore our simulations also include error bars and the number of shots for each simulations for further reference.

## Acknowledgement

Firstly, I would like to thank my supervisor professor Gayane Vardoyan and my daily supervisor Bethany Davies for providing this project. My work would have been impossible without their invaluable guidance for the past few months. I would also like to thank my friend Aksel and all the group members of our project for all the discussions we have had. Many thanks to professor Rihan Hai for reviewing and examining my thesis. Last but not least, I'm grateful to my parents. Their endless support has given me courage and strength in this process and all the 22 past years that I have had.

## Appendix A. Trade-off between success probability and Fidelity

In a single-click protocol that we considered here [7], the two remote qubits will be prepared at a superposition state $\sqrt{\alpha}\,|0\rangle + \sqrt{1-\alpha}\,|1\rangle$. The parameter $\alpha$ is named brightness and state $|0\rangle$ is called bright state. we can drive the bright state $|0\rangle$ via optical transition to an excited state $|e\rangle$ which will decay and emit a single photon [7].

Upon generation, two wave-pulse will be applied simultaneously at the two qubits to a superposition of emitting a photon and not emitting a photon. Then both photons will travel through a beamsplitter where the path information is removed. Finally, detecting a single photon will produce an entangled state $|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ [7].

However, the detector cannot tell the difference between one photon emitted and two photons emitted but one is lost. In the case of high photon loss $\eta \ll 1$, the resulting state that we produce in this way would be $\rho = (1-\alpha)\left|\Psi^+\right\rangle\!\left\langle\Psi^+\right| + \alpha\left|00\right\rangle\!\left\langle 00\right|$, and the success probability would be [7]

$$p_{succ} = \alpha\eta|\zeta(t')|^2. \tag{27}$$

with $|\zeta(t')|^2$ is a constant related to the optical mode. This will give us an entanglement link with fidelity [7]

$$F = 1 - \alpha. \tag{28}$$

When we batch $M$ single-photon generations with $Mp_{succ} \ll 1$, the probability of having at least one success would be [8]

$$p = 1 - (1-p_{succ})^M \simeq Mp_{succ}. \tag{29}$$

Combining (27), (28) and (29), we arrive at the trade-off function for a single-click-based entanglement generation platform with a tunable brightness parameter [8]:

$$F = 1 - \lambda p, \quad \lambda = \frac{\alpha}{Mp_{succ}} = \frac{1}{M\eta|\zeta(t')|^2}. \tag{30}$$

The batch treatment of sending photons also results in natural discrete timesteps for the system with each entanglement generation attempt costing a constant amount of time $\Delta t = Mt_g$ where $t_g$ is the time for a single generation. We may argue that in the case of batch generation, we can always stop the current batch of generation once an entangled link is established. This can arguably reduce the average time for the generation attempt. However, this introduces stochastic uncertainty. For the simplicity of our analysis, we do not consider them in this work.

## Appendix B. Interpolating Between Two Discrete Actions

Suppose we have two original actions $\{(p_1, F_1), (p_2, F_2)\}$, we can define a continuous family of actions $\{(p_q, F_q) : q \in (0,1)\}$ as the probabilistic mixture of the two actions. We first conduct a Bernoulli test with probability $q$ and perform action 1 if it is tail and action 2 otherwise:

1. Sample $x$ from $X \sim Bernoulli(q)$;
2. Conduct action 1 if $x = 0$, conduct action 2 if $x = 1$ ;
3. If the link is established, assign its bin as $n_q = n(F_q)$.

Then the success probability for generating a link would be

$$P_q(success) = qp_1 + (1-q)p_2. \tag{31}$$

To derive the fidelity $F_q$, let's denote the link for action 1 is $\rho_1$ and the link for action 2 is $\rho_2$, then the resulting state of the new action would be a mixture state of $\rho_1, \rho_2$ with probability conditioned on successful generation:

$$\rho_q = \rho_1 P_q(x=0|success) + \rho_2 P_q(x=1|success) \tag{32}$$

$$= \frac{qp_1\rho_1 + (1-q)p_2\rho_2}{qp_1 + (1-q)p_2}. \tag{33}$$

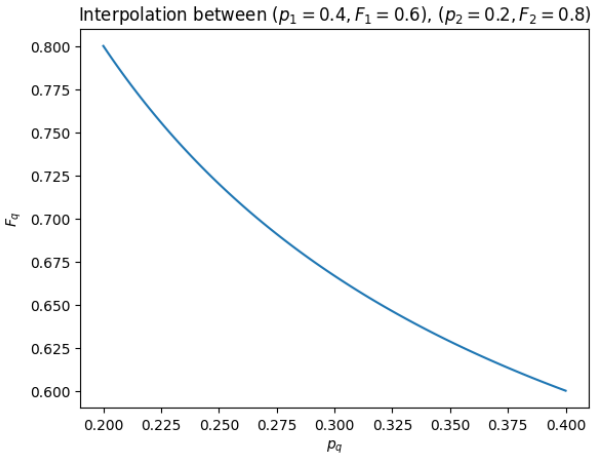


Figure 7: This plots the $F_q - p_q$ trade-off when we are interpolating between two actions specified in the figure. Here we have $\lambda = 1$

Since fidelity is a linear function of the density matrix $F(a\rho_1 + b\rho_2, \left|\phi^+\right\rangle\!\left\langle\phi^+\right|) = aF_1 + bF_2$, its fidelity can be expressed as

$$F_q = F(\rho_q, \left|\phi^+\right\rangle\!\left\langle\phi^+\right|) = \frac{qp_1F_1 + (1-q)p_2F_2}{qp_1 + (1-q)p_2}. \tag{34}$$

In this case, the $F_q - p_q$ trade-off is

$$F_q(p_q) = \frac{1}{p_q}\frac{p_1p_2(F_2 - F_1)}{p_1 - p_2} + \frac{p_1F_1 - p_2F_2}{p_1 - p_2} \tag{35}$$

$$= \frac{\lambda p_1 p_2}{p_q} + 1 - \lambda(p_1 + p_2). \tag{36}$$

With this interpolation, we can always achieve a continuous action space with only two discrete actions. This would allow us to greatly simplify our experimental setting to two fixed brightness parameters described in Appendix A. However, the native linear trade-off performs better than the interpolation method we introduced here as we can see in Figure 7. Due to the discrete nature of the fidelity bins, we can argue that under certain parameters and smaller gaps between two actions, the performance can be equivalent to the linear trade-off action space. This would allow us to perform a discrete set of actions with interpolation to achieve the same performance as native continuous action space.

There is also an interesting discovery about the average fidelity bins under interpolation. Suppose the memory is informed about the choice of actions in the process and assigns fidelity bin $n = n(F_1)$ for the link generated with action 1 and assigns $n = n(F_2)$ for action 2, we can still calculate the average fidelity bin overall $\bar{n}_q$ as:

$$\bar{n}_q = n(F_1)P_q(x=0|success) + n(F_2)P_q(x=1|success). \tag{37}$$

We can see that this is smaller than $n_q$ due to the concavity of the fidelity bin function in (9):

$$\bar{n}_q = n(F_1)P_q(x=0|success) + n(F_2)P_q(x=1|success) \tag{38}$$

$$\leq n(F_1P_q(x=0|success) + F_2P_q(x=1|success)) = n_q. \tag{39}$$

Thus forgetting the actual choice we have made can increase the lifespan of our link.

This idea of interpolation is developed independently from the work in optimizing purification [10] where they adopted interpolation between different purification protocols. They also showed that extrapolation is possible towards fidelity 0 with similar ideas.

## Appendix C. Entanglement Purification Schemes

We reiterate the two entanglement purification schemes that are specified in [10].

### EPL-D scheme

Input: two identical copies of R-states

Apply bilocal CNOT gates between the two copies.
Measure the target qubits and communicate the results.
**if** The measured flags are 11 **then**
    Output the source (first) copy

### DEJMPS scheme

Although DEJMPS can be applied to a wider range of states, here we only consider Werner states as inputs. DEJMPS also requires the input fidelity to be larger than 0.5.

Input: two identical copies of Werner states with fidelity higher than 0.5

Rotate both qubits on Alice's side by $\frac{\pi}{2}$ around the X axis and by $-\frac{\pi}{2}$ on Bob's side
Apply bilocal CNOT gates between the two copies
Measure the target qubits and communicate the results.
**if** The measured flags are 00 or 11 **then**
    Output the source (first) copy

## A Appendix D. Lower Bound Estimation for the State Space Size

When the number of required links increases, the maximum fidelity bin also needs to increase respectively in order to allow such a number of links to be generated. For example, if the highest fidelity action has fidelity bin four, it is impossible to generate 20 links. Suppose the maximum fidelity bin and the number of links have a fixed ratio $n(F_{max}) = \lfloor as \rfloor, a > 1$, the state space size has a lower bound:

$$|\mathcal{S}| = \sum_{i=0}^{s} \binom{n(F_{max})}{i} > \binom{\lfloor as \rfloor}{s}. \tag{40}$$

If $a \geq 2$, then the lower bound for the space size can be estimated as

$$\binom{\lfloor as \rfloor}{s} \geq \binom{2s}{s} > \left(\frac{2s}{s}\right)^s = 2^s \tag{41}$$

which grows exponentially for $s$.

If $a < 2$, then we can relax the combinatorics using the fact that $\binom{m}{n} \geq \binom{m-l}{n-l}; m, n, l \in \mathcal{N}; m \geq n \geq l$ by setting $m = \lfloor as \rfloor, n = s, l = 2s - \lfloor as \rfloor$:

$$\binom{\lfloor as \rfloor}{s} \geq \binom{(2(\lfloor as \rfloor - s)}{\lfloor as \rfloor - s} > 2^{\lfloor as \rfloor - s}. \tag{42}$$

We can see that in this case the lower bound also grows exponentially.

## Appendix E. Modified Purification on Arrival

**Algorithm 3** Modified Purification$(s, \mathcal{A}_{optimal}, \alpha)$

$t_{ahead}(s) \leftarrow 0$
$\pi'(s) \leftarrow \underset{a}{\text{argmax}}\{p_a : (p_a, F_a) \in \mathcal{A}_{optimal}, F_a \geq F_{th}\}$
$a_p \leftarrow (p'_{max}, 1)$ ▷ Only one purification action
**for** $i = \{s-1, s-2, \ldots, 2, 1\}$ **do**
    $t_{ahead}(i) \leftarrow t_{ahead}(i+1) + \frac{1}{p_{\pi(i+1)}}$
    $a_{nop} \leftarrow \arg_{a \in \mathcal{A}_{optimal}}\{n(F_a) = \lfloor \alpha \cdot t_{ahead}(i+1) \rfloor\}$
    **if** $2p_{a_{nop}} > p_{a_p}$ **then**
        $\pi'(i) \leftarrow a_{nop}$
    **else**
        $\pi'(i) \leftarrow a_p$
        **return** $\pi'$

## References


[1] M. A. Nielsen and I. Chuang. *Quantum computation and quantum information*. University Press, Cambridge, 10th anniversary edition edition, 2010.

[2] Mark M Wilde. From Classical to Quantum Shannon Theory. 2019.

[3] Stephanie Wehner, David Elkouss, and Ronald Hanson. Quantum internet: A vision for the road ahead. *Science*, 362(6412), 2018.

[4] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, October 2009.

[5] Álvaro G. Iñesta, Gayane Vardoyan, Lara Scavuzzo, and Stephanie Wehner. Optimal entanglement distribution policies in homogeneous repeater chains with cutoffs. *npj Quantum Information*, 9(1):46, May 2023.

[6] M. Pompili, C. Delle Donne, I. Te Raa, B. Van Der Vecht, M. Skrzypczyk, G. Ferreira, L. De Kluijver, A. J. Stolk, S. L. N. Hermans, P. Pawełczak, W. Kozlowski, R. Hanson, and S. Wehner. Experimental demonstration of entanglement delivery using a quantum network stack. *npj Quantum Information*, 8(1):121, October 2022.

[7] S L N Hermans, M Pompili, L Dos Santos Martins, A R-P Montblanch, H K C Beukers, S Baier, J Borregaard, and R Hanson. Entangling remote qubits using the single-photon protocol: an in-depth theoretical and experimental study. *New Journal of Physics*, 25(1):013011, January 2023.

[8] Bethany Davies, Thomas Beauchamp, Gayane Vardoyan, and Stephanie Wehner. Tools for the analysis of quantum protocols requiring state generation within a time window, April 2023. arXiv:2304.12673 [quant-ph].

[9] Charles H. Bennett, David P. DiVincenzo, John A. Smolin, and William K. Wootters. Mixed-state entanglement and quantum error correction. *Phys. Rev. A*, 54:3824–3851, Nov 1996.

[10] Filip Rozpędek, Thomas Schiet, Le Phuc Thinh, David Elkouss, Andrew C. Doherty, and Stephanie Wehner. Optimizing practical entanglement distillation. *Phys. Rev. A*, 97(6):062333, June 2018.

[11] David Deutsch, Artur Ekert, Richard Jozsa, Chiara Macchiavello, Sandu Popescu, and Anna Sanpera. Quantum privacy amplification and the security of quantum cryptography over noisy channels. *Phys. Rev. Lett.*, 77:2818–2821, Sep 1996.

[12] W. Dür, M. Hein, J. I. Cirac, and H.-J. Briegel. Standard forms of noisy quantum operations via depolarization. *Physical Review A*, 72(5):052326, November 2005. arXiv:quant-ph/0507134.

[13] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 1 edition, April 2018.

[14] T.Qu. performance evaluation for the heuristic policy. https://github.com/qutianchen/heuristic-multiple-entanglement-generation, 2024.