Streaming quantum state purification

Andrew M. Childs[1,2], Honghao Fu[1,2,3,6], Debbie Leung[1,4], Zhi Li[4], Maris Ozols[5], and Vedang Vyas[1]

[1]Institute for Quantum Computing, University of Waterloo, Waterloo, ON, N2L 3G1, Canada
[2]Department of Computer Science, Institute for Advanced Computer Studies, and Joint Center for Quantum Information and Computer Science, University of Maryland, College Park, MD 20742, USA
[3]Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, USA
[4]Perimeter Institute for Theoretical Physics, 31 Caroline St. N., Waterloo, ON N2L 2Y5, Canada
[5]QuSoft and University of Amsterdam, Science Park 123, 1098 XG, Amsterdam, the Netherlands
[6]Concordia Institute of Information Systems Engineering, Concordia University, 1455 Blvd. De Maisonneuve Ouest, Montreal, QC H3G 1M8, Canada

Quantum state purification is the task of recovering a nearly pure copy of an unknown pure quantum state using multiple noisy copies of the state. This basic task has applications to quantum communication over noisy channels and quantum computation with imperfect devices, but has primarily been studied previously for the case of qubits. We derive an efficient purification procedure based on the swap test for qudits of any dimension, starting with any initial error parameter. Treating the initial error parameter and the dimension as constants, we show that our procedure has sample complexity asymptotically optimal in the final error parameter. Our protocol has a simple recursive structure that can be applied when the states are provided one at a time in a streaming fashion, requiring only a small quantum memory to implement.

Contents

1 Introduction

Quantum states are notoriously susceptible to the effects of decoherence. Thus, a basic challenge in quantum information processing is to find ways of protecting quantum systems from noise, or of removing noise that has already occurred. In this paper we study the latter approach, aiming to (partially) reverse the effect of decoherence and produce less noisy states out of mixed ones, a process that we call *purification*.

Since noise makes quantum states less distinguishable, it is impossible to purify a single copy of a noisy state. However, given multiple copies of a noisy state, we can try to reconstruct a single copy that is closer to the original pure state. In general, we expect the quality of the reconstructed state to be higher if we are able to use more copies of the noisy state. We would like to understand the number of samples that suffice to produce a high-fidelity copy of the original state, and to give efficient procedures for carrying out such purification.

More precisely, we focus on depolarizing noise and consider the following formulation of the purification problem. Let $|\psi\rangle \in \mathbb{C}^d$ be an unknown pure d-dimensional state (or *qudit*). In the *qudit purification problem*, we are given multiple noisy copies of $|\psi\rangle$ that are all of the form

$$\rho(\delta) := (1-\delta)|\psi\rangle\langle\psi| + \delta\,\frac{I}{d}, \tag{1}$$

where the *error parameter* $\delta \in (0,1)$ represents the probability that the state is depolarized. If we need to compare error parameters across different dimensions, we write $\delta^{(d)}$ to specify the underlying dimension.

Let $\mathcal{P}$ denote a purification procedure, which is a quantum operation that maps N copies of $\rho(\delta)$ to a single qudit. (Note that the implementation of $\mathcal{P}$ can make use of additional quantum workspace.) We refer to N as the *sample complexity* of $\mathcal{P}$. The goal is to produce a single qudit that closely approximates the desired (but unknown) target state $|\psi\rangle$. In particular, we aim to produce a state whose fidelity with the ideal pure state $|\psi\rangle$ is within some specified tolerance, using the smallest possible sample complexity N. Our streaming protocol produces a state of the form $\rho(\epsilon)$ for some small $\epsilon > 0$, which has fidelity $1-(1-\frac{1}{d})\epsilon$ with $|\psi\rangle$, so it is convenient to focus on the output error parameter ϵ.

The purification problem was first studied over two decades ago. Inspired by studies of entanglement purification [BBP+96], Cirac, Ekert, and Macchiavello studied the problem of purifying a depolarized qubit and found the optimal purification procedure [CEM99]. Later, Keyl and Werner studied qubit purification under different criteria, including the possibility of producing multiple copies of the output state and measuring the fidelity either by comparing a single output state or selecting all the output states [KW01]. Subsequent work studied probabilistic quantum state purification using semidefinite programming [Fiu04]. However, until recently, little was known about how to purify higher-dimensional quantum states.

Our main contribution is the first concrete and efficient qudit purification procedure, which achieves the following.

Theorem 1 (Informal)**.** *For any fixed input error parameter $\delta \in (0,1)$, there is a protocol that produces a state with output error parameter at most $\epsilon \in (0,1)$ using $O(1/\epsilon)$ samples of the d-dimensional input state and $O(\frac{1}{\epsilon}\log d)$ elementary gates.*

We present this protocol in Section 2 and analyze its performance in Section 3, establishing a formal version of the above result (Theorem 9 and Corollary 11).

In Section 4, we describe several applications of our results. In Section 4.1, we use our procedure to solve a version of Simon's problem with a faulty oracle. In particular, we show that if the oracle depolarizes its output by a constant amount, then the problem can still be solved with only quadratic overhead, preserving the exponential quantum speedup. In Section 4.2, we discuss the relationship between mixedness testing (introduced and studied in [MW16, OW15b, Wri16]), quantum state purification, and quantum state tomography. In some sense, these problems are increasingly difficult. Our protocol implies a dimension-independent upper bound for the sample complexity of mixedness testing for depolarized states, below the worst-case lower bound shown in [OW15b]. On the other hand, quantum state tomography can be used for quantum state purification, even in a streaming fashion. We compare the performance of this approach and purification based on the recursive swap test. The swap test is more efficient for large dimension and fixed input noise, whereas state tomography is more efficient for small dimension and very small signal. In Section 4.3, we briefly discuss the relationship between quantum state purification and the quantum majority vote problem for qubits, including how the former provides a higher-dimensional generalization of the latter.

Rather than applying Schur–Weyl duality globally as in [CEM99], our streaming purification protocol recursively applies the swap test to pairs of quantum states. This approach has two advantages. First, our procedure readily works for quantum states of any dimension. Second, the swap test is much easier to implement and analyze than a general Schur basis measurement. In particular, the protocol can be implemented in a scenario where the states arrive in an online fashion, using a quantum memory of size only logarithmic in the total number of states used.

Since our protocol does not respect the full permutation symmetry of the problem, it is not precisely optimal. However, suppose we fix the dimension and the initial error parameter. Then, in the qubit case, our sample complexity matches that of the optimal protocol [CEM99] up to factors involving these constants, suggesting that the procedure performs well. Furthermore, in Section 5 we prove a lower bound showing that our protocol has optimal scaling in the final error parameter up to factors involving the other constants.

The optimal purification protocol should respect the permutation symmetry of the input and hence can be characterized using Schur–Weyl duality. Some of the authors developed a framework for studying such optimal purification procedures, which led to a recently reported optimal protocol [LFIC24] (discussed further below). We briefly describe this related work in Section 6 along with a summary of the results and a discussion of some open questions.

Since this manuscript was first made available as an arXiv preprint, our purification protocol has been applied to more general noisy states, with applications in the problem of principal eigenstate classical shadows [GPS24]. Additional results connecting purification and shadow estimation can be found in [ZL22, LLY+24]. Furthermore, alternative streaming purification protocols have been developed using semidefinite programming and block encoding techniques [YCH+24, YKLO24].

After our initial journal submission, [LFIC24] reported an optimal purification procedure for arbitrary qudit states using Schur–Weyl duality. In particular, the optimal sample complexity is now known for arbitrary dimension. A comparison with this optimal result shows that our method based on the recursive swap test has optimal expected sample complexity as a function of the dimension (in addition to the final error parameter), up to a constant factor, for initial error $\delta < 1/2$. Since the optimal protocol involves performing

the Schur transform [BCH06] and applying a change of basis between different irreducible representations of the unitary group, it is unclear whether the optimal protocol can be implemented efficiently. Even if it can, it is unlikely to be practical. Moreover, the optimal protocol cannot be used as a gadget in the streaming model because its output is not a depolarized state, unlike the swap-test gadget. Hence, it requires a large quantum memory to store N input states. In contrast, our streaming protocol only needs a quantum memory of size $O(\log(N))$ and very simple operations. Therefore, our approach will likely be the method of choice in practice.

2 Purification using the swap test

In this section, we present a recursive purification procedure based on the swap test. We review the swap test in Section 2.1 and define a gadget that uses it to project onto the symmetric subspace in Section 2.2, and then we specialize the analysis to the case where the two input copies have the same purity. Next, we define our recursive purification procedure in Section 2.3. Finally, we describe a stack-based implementation of this procedure in Section 2.4, providing a bound on the space complexity.

2.1 The swap test

The *swap test* [BCWDW01] is given by the circuit shown in Fig. 1. It takes two input

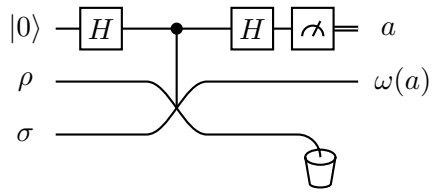
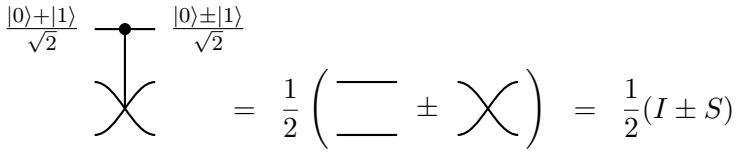


Figure 1: Quantum circuit for the swap test on arbitrary input states ρ and σ. The gate in the middle denotes the controlled-swap operation while $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ is the Hadamard gate.

registers of the same dimension, and requires one ancilla qubit initialized to $|0\rangle$. The gate after the first Hadamard gate is the controlled-swap gate $|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes S$, where I and S denote respectively the identity gate and the *swap gate* ($|i\rangle \otimes |j\rangle \mapsto |j\rangle \otimes |i\rangle$) acting on the last two registers. Since the control qubit is prepared in the state $H|0\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and then measured in the Hadamard basis $\frac{|0\rangle\pm|1\rangle}{\sqrt{2}}$, we can use the identity



(2)

to eliminate the control qubit when analyzing the output. Thus for joint input μ and measurement outcome $a \in \{0, 1\}$ (see Fig. 1), the corresponding sub-normalized post-measurement state on the last two registers is

$$\frac{I + (-1)^a S}{2}\, \mu\, \frac{I + (-1)^a S}{2}, \tag{3}$$

and the trace gives the probability of the outcome a. Note that $\frac{1}{2}(I + S)$ and $\frac{1}{2}(I - S)$ project onto the *symmetric* and the *anti-symmetric subspace*, respectively.

The swap test is a standard procedure for comparing two unknown *pure* quantum states $|\psi_{1,2}\rangle$; in this case, $\mu = |\psi_1\rangle\langle\psi_1| \otimes |\psi_2\rangle\langle\psi_2|$. The probability to obtain the outcome $a = 0$ decreases from 1 to 1/2 as the fidelity between the inputs decreases from 1 to 0 (i.e, as the inputs change from being identical to being orthogonal). If we instead apply the swap test to two copies of a mixed state ρ (i.e., $\mu = \rho \otimes \rho$), the postmeasurement state for the $a = 0$ outcome has increased weights for matching eigenvectors on the two inputs, suggesting that the swap test can be used for purification. The swap test also potentially entangles the two output quantum registers, so our purification algorithm keeps only one of the two quantum output registers.

We now make this intuition concrete. Starting from a product input $\mu = \rho \otimes \sigma$ in eq. (3) and discarding the last register yields the sub-normalized state

$$\frac{1}{4}(\rho + \sigma + (-1)^a \rho\sigma + (-1)^a \sigma\rho), \tag{4}$$

where the last two terms are obtained from the identity $\mathrm{Tr}_2[(\rho \otimes \sigma)S] = \rho\sigma$.

The probability of obtaining outcome a is given by the trace of eq. (4),

$$\frac{1}{2}(1 + (-1)^a \,\mathrm{Tr}(\rho\sigma)), \tag{5}$$

and the normalized output state $\omega(a)$ in Fig. 1 is

$$\omega(a) = \frac{1}{2} \cdot \frac{\rho + \sigma + (-1)^a(\rho\sigma + \sigma\rho)}{1 + (-1)^a \,\mathrm{Tr}(\rho\sigma)}. \tag{6}$$

Note that the probability of the $a = 0$ outcome is at least 1/2 for a product input. Note also that the swap test can be implemented with $O(\log d)$ quantum gates since a swap gate between qudits can be decomposed into swaps of constituent qubits.

2.2 The swap test gadget

We are particularly interested in the output state $\omega(0)$ that corresponds to the measurement outcome $a = 0$. This state is equal to the normalized projection of $\rho \otimes \sigma$ onto the symmetric subspace. Algorithm 1 introduces a Swap *gadget* that performs the swap test on fresh copies of the input states ρ and σ until it succeeds in obtaining the $a = 0$ outcome.

Algorithm 1 Swap gadget.

This procedure has access to a stream of inputs ρ and σ, and uses as many copies of them as necessary.

```
procedure Swap(ρ, σ)
    repeat
        Ask for one fresh copy of ρ and σ.
        Apply the swap test shown in Fig. 1 and denote the measurement outcome by a.
    until a = 0
    return the state ω(0) of the output register
```

We denote the final output of the Swap gadget by

$$\mathrm{Swap}(\rho, \sigma) := \frac{1}{2} \cdot \frac{\rho + \sigma + \rho\sigma + \sigma\rho}{1 + \mathrm{Tr}(\rho\sigma)}, \tag{7}$$

which is just $\omega(0)$ from eq. (6). The expected number of copies of each input state consumed by this gadget is given by the inverse of its success probability. By eq. (5), this is $2/(1+\mathrm{Tr}(\rho\sigma))$.

We focus on applying the swap test gadget when $\rho = \sigma$. From eq. (5), the outcome $a = 0$ in the swap test occurs with probability

$$\frac{1}{2}(1+\mathrm{Tr}(\rho^2)) \tag{8}$$

and, according to eq. (7), the normalized output state is

$$\textsc{Swap}(\rho,\rho) = \frac{\rho+\rho^2}{1+\mathrm{Tr}(\rho^2)}. \tag{9}$$

Note that $\textsc{Swap}(\rho,\rho)$ shares the same eigenvectors with ρ and ρ^2. If the eigenvalues of ρ are $\lambda_1 > \lambda_2 \geq \lambda_2 \geq \cdots \geq \lambda_d$, then the eigenvalues of $\textsc{Swap}(\rho,\rho)$ are

$$\lambda_i' = \frac{\lambda_i+\lambda_i^2}{1+\sum_{j=1}^d \lambda_j^2}. \tag{10}$$

In particular,

$$\lambda_1' - \lambda_1 = \frac{\lambda_1+\lambda_1^2}{1+\sum_{j=1}^d \lambda_j^2} - \lambda_1 = \frac{\lambda_1^2\sum_{j=1}^d \lambda_j - \lambda_1\sum_{j=1}^d \lambda_j^2}{1+\sum_{j=1}^d \lambda_j^2} = \frac{\lambda_1\sum_{j=2}^d(\lambda_1-\lambda_j)\lambda_j}{1+\sum_{j=1}^d \lambda_j^2}, \tag{11}$$

which is positive unless ρ is pure or maximally mixed. In particular, the following corollary holds for the noisy states we want to purify.

Corollary 2. *If $\delta \in (0,1)$ then $\textsc{Swap}(\rho(\delta),\rho(\delta)) = \rho(\delta')$ where $\delta' < \delta$.*

This corollary suggests a simple recursive algorithm that applies the Swap gadget on a pair of identical inputs produced in the previous level of the recursion. Corollary 2 then guarantees that at each subsequent level of recursion, the states become purer. Before we formally state and analyze this recursive procedure, let us first derive the success probability of the swap test on two copies of $\rho(\delta)$, the expected number of copies of $\rho(\delta)$ consumed by the Swap gadget, and the error parameter δ' of the output state $\rho(\delta')$. The success probability of the swap test can be found from eq. (8) and the spectrum of $\rho(\delta)$:

$$\begin{aligned} p(\delta) &:= \frac{1}{2}\left(1+\left((1-\delta)+\frac{\delta}{d}\right)^2+(d-1)\left(\frac{\delta}{d}\right)^2\right) \\ &= 1-\left(1-\frac{1}{d}\right)\delta+\frac{1}{2}\left(1-\frac{1}{d}\right)\delta^2. \end{aligned} \tag{12}$$

Accordingly, the expected number of copies of $\rho(\delta)$ consumed by the Swap gadget is

$$\frac{2}{p(\delta)} = \frac{2d}{d-(d-1)\delta+\frac{1}{2}(d-1)\delta^2}. \tag{13}$$

The output state of the Swap gadget is $\textsc{Swap}\big(\rho(\delta),\rho(\delta)\big) = \rho(\delta')$, where δ' can be obtained from eq. (10):

$$\frac{\delta'}{d} = \lambda_2' = \frac{\frac{\delta}{d}+\left(\frac{\delta}{d}\right)^2}{2p(\delta)}, \tag{14}$$

so

$$\delta' = \frac{\delta + \delta^2/d}{2p(\delta)} = \frac{\delta + \delta^2/d}{2 - 2\left(1 - \frac{1}{d}\right)\delta + \left(1 - \frac{1}{d}\right)\delta^2}. \tag{15}$$

If the inputs are only similar but not identical, the above analysis holds approximately. In Appendix A, we evaluate the output error parameter of the SWAP gadget when the inputs are $\rho(\delta_1)$ and $\rho(\delta_2)$ for arbitrarily δ_1, δ_2. We present the region of (δ_1, δ_2) for which the output has better purity than both inputs, which includes the line $\delta_1 = \delta_2$. For large δ_1, δ_2, the region becomes narrower as d increases. We thus design the protocol to apply the swap test to states of equal noise parameter.

2.3 Recursive purification based on the swap test

Given access to many copies of $\rho(\delta)$ from eq. (1), with a given value of $\delta \in (0, 1)$, our goal is to produce a single copy of $\rho(\epsilon)$, for some desired target parameter $\epsilon \ll \delta$. Since our purification procedure (Algorithm 2) invokes itself recursively, it will be convenient to denote the initial state by $\rho_0 := \rho(\delta)$ and the state produced at the ith level of the recursion by ρ_i. We choose the total depth of the recursion later to guarantee a desired level of purity in the final output.

Before stating our algorithm more formally, we emphasize one slightly unusual aspect: even though we fix the total depth of the recursion, the number of calls *within* each level is not known in advance. This is because each instance of the algorithm calls an instance in the previous recursion level an unknown number of times (as many times as necessary for the SWAP gadget to succeed). We analyze the expected number of calls and the total number of input states consumed in Section 3.3.

Algorithm 2 Recursive purification based on the swap test.

This procedure purifies a stream of states ρ_0 by recursively calling the SWAP gadget (Algorithm 1).

```
1: procedure PURIFY(n)
2:     if n = 0 then                          If at the bottom level of recursion,
3:         return ρ_0                         request one copy of ρ_0 from the stream and return it.
4:     else                                   Otherwise call the next level of recursion
5:         return SWAP(PURIFY(n − 1), PURIFY(n − 1))   until the SWAP gadget succeeds.
```

Let $\rho_n := \text{PURIFY}(n)$ denote the output of a purification procedure of depth n. We can represent this procedure as a full binary tree of height n whose 2^n leaves correspond to the original input states $\rho_0 = \rho(\delta)$ and whose root corresponds to the final output state ρ_n. For any $i \in \{0, \ldots, n\}$, there are 2^{n-i} calls to $\text{PURIFY}(i)$, each producing a copy of ρ_i by combining two states from the previous level (or by directly consuming one copy of ρ_0 when $i = 0$). Every time a SWAP gadget fails, the recursion is restarted at that level. This triggers a cascade of calls going down all the way to $\text{PURIFY}(0)$, which requests a fresh copy of ρ_0. Hence, a failure at level i results in at least 2^i fresh copies of ρ_0 being requested. The procedure terminates once the outermost SWAP gadget succeeds.

According to Corollary 2, the state ρ_i is of the form $\rho(\delta_i)$ for some error parameter $\delta_i < \delta_{i-1}$. Denote by p_i the probability of success of the SWAP gadget in $\text{PURIFY}(i)$ (in other words, p_i is the probability of getting outcome $a = 0$ when the swap test is applied on two copies of ρ_{i-1}). According to eqs. (12) and (15), the parameter p_i depends on δ_i,

while δ_i in turn is found via a recurrence relation:

$$p_i = P(\delta_{i-1}, d), \qquad \delta_i = \Delta(\delta_{i-1}, d), \qquad \delta_0 = \delta, \tag{16}$$

where the functions $P, \Delta \colon (0,1) \times [2,\infty) \to (0,1)$ are defined as follows:

$$P(\delta, d) := 1 - \Big(1 - \frac{1}{d}\Big)\delta + \frac{1}{2}\Big(1 - \frac{1}{d}\Big)\delta^2, \qquad \Delta(\delta, d) := \frac{\delta + \delta^2/d}{2P(\delta, d)}. \tag{17}$$

For any values of the initial purity $\delta_0 \in (0,1)$ and dimension $d \geq 2$, these recursive relations define a pair of sequences $(p_1, p_2, \dots)$ and $(\delta_0, \delta_1, \dots)$. Our goal is to understand the asymptotic scaling of the parameters p_i and δ_i, and to determine a value n such that $\delta_n \leq \epsilon$, for some given $\epsilon > 0$.

2.4 Non-recursive stack-based implementation

It is not immediately obvious from the recursive description of Purify(n) in Algorithm 2 that it uses $n+1$ qudits of quantum memory plus one ancilla qubit. Algorithm 3 provides an alternative stack-based implementation of the same procedure that makes this clear.

Algorithm 3 Stack-based implementation of Algorithm 2.

Global variables:

- ▷ qudit[$\cdot$] – *an array of qudits that simulates a stack,*
- ▷ purity[$\cdot$] – *an array of integers that store the purity level of each qudit,*
- ▷ k – *an integer that indicates the current position in the stack.*

This procedure requests a fresh copy of ρ_0 from the stream and stores it in the stack.

procedure FetchNewCopy
  k++ *Move the stack pointer forward by one.*
  qudit[k] $= \rho_0$ *Fetch a fresh copy of ρ_0 and store it in the current stack position.*
  purity[k] $= 0$ *Set its purity level to 0.*

Non-recursive implementation of the Purify(n) procedure from Algorithm 2.

procedure Purify(n)
  $k = 0$ *Initially the stack is empty.*
  purity[0] $= -1$ *Set the purity level of empty stack to a non-existing value.*
  repeat *Repeat until n levels of recursion are reached.*
    FetchNewCopy *Request two fresh copies of ρ_0*
    FetchNewCopy *and put them in the stack.*
    repeat *Keep merging the two top-most qudits.*
      Apply the swap test circuit shown in Fig. 1 to qudit[$k-1$] and qudit[k], store the result in qudit[$k-1$], and let a be the measurement outcome.
      if $a = 0$ **then** *If the swap test succeeded,*
        $k = k - 1$ *move the pointer backwards by one (i.e., keep the improved qudit)*
        purity[k]++ *and increase its purity level by one.*
      else *If the swap test failed,*
        $k = k - 2$ *move the pointer backwards by two (i.e., discard both qudits).*
    until $a = 1$ or purity[$k-1$] $\neq$ purity[k] *Stop if swap test failed or different purities.*
  until purity[1] $= n$ *Stop when n levels of recursion are reached.*
  return qudit[1] *Return the only remaining qudit.*

Note that throughout the execution of Algorithm 3, $k \geq 0$, purity$[0] = -1$, and purity$[1] \geq$ purity$[2] \geq \cdots \geq$ purity$[k]$ with at most one equality. The last statement can be verified by noting that it is satisfied initially, and that it is preserved whenever any of the purity values are updated. The fact that at most two purities can be equal implies the stated upper bound for the memory required. Just as in the recursive formulation in Algorithm 2, the swap test is always applied to two states of equal purity. In the first pass through the repeat loop in line 7, the swap test is applied to two new copies of ρ_0. In subsequent passes, the condition in line 14 ensures that the next inputs to the swap test have equal purity.

3 Analysis of recursive purification

The goal of this section is to analyze the complexity of the recursive purification procedure PURIFY(n) defined in Algorithm 2. Most of the analysis deals with understanding the recurrence relations in eqs. (16) and (17). First, in Section 3.1 we establish monotonicity of the functions P and Δ from eq. (17) and the corresponding parameters p_i and δ_i. In Section 3.2, we analyze the asymptotic scaling of the error parameter δ_i. Finally, in Section 3.3, we analyze the expected sample complexity of PURIFY(n).

3.1 Monotonicity relations

First we prove monotonicity of the functions P and Δ in eq. (17).

Lemma 3. *For any $\delta \in (0,1)$ and $d \geq 2$, the function $P(\delta, d)$ is strictly decreasing in both variables, and the function $\Delta(\delta, d)$ is strictly increasing in both variables.*

Proof. To verify the monotonicity of P, note that

$$\frac{\partial P}{\partial \delta} = \Big(1 - \frac{1}{d}\Big)(-1+\delta) < 0, \qquad \frac{\partial P}{\partial d} = \Big(-1 + \frac{\delta}{2}\Big)\frac{\delta}{d^2} < 0. \tag{18}$$

Similarly, the partial derivatives of Δ satisfy

$$\frac{\partial \Delta}{\partial \delta} = \frac{1 + 2\delta/d}{2P(\delta,d)} - \frac{\delta + \delta^2/d}{2P(\delta,d)^2} \cdot \frac{\partial}{\partial \delta} P(\delta, d) = \frac{(d+2\delta)P(\delta,d) + \delta(d+\delta)(1-1/d)(1-\delta)}{2dP(\delta,d)^2} > 0, \tag{19}$$

$$\frac{\partial \Delta}{\partial d} = \frac{-\delta^2/d^2}{2P(\delta,d)} - \frac{\delta + \delta^2/d}{2P(\delta,d)^2} \cdot \frac{\partial}{\partial d} P(\delta, d) = \frac{(1-\delta)\delta^3}{4d^2 P(\delta,d)^2} > 0, \tag{20}$$

which completes the proof. □

Thanks to the recurrence relations in eq. (16), the parameters p_i and δ_i inherit the monotonicity of P and Δ from Lemma 3. We denote them by $p_i^{(d)}$ and $\delta_i^{(d)}$ to explicitly indicate their dependence on d. The following lemma shows that both parameters are strictly monotonic in the dimension d.

Lemma 4. *For any $i \geq 1$, the success probability $p_i^{(d)}$ is monotonically strictly decreasing in d and the error parameter $\delta_i^{(d)}$ is monotonically strictly increasing in d, for all integer values of $d \geq 2$.*

Proof. Recall from eq. (16) that $\delta_0^{(d)} = \delta = \delta_0^{(d+1)}$, irrespective of d. Using the recurrence from eq. (16) and the strict monotonicity of Δ in d from Lemma 3,

$$\delta_1^{(d)} = \Delta(\delta_0^{(d)}, d) < \Delta(\delta_0^{(d+1)}, d+1) = \delta_1^{(d+1)}. \tag{21}$$

Using this as a basis for induction, the strict monotonicity of Δ in both arguments implies that

$$\delta_i^{(d)} = \Delta(\delta_{i-1}^{(d)}, d) < \Delta(\delta_{i-1}^{(d+1)}, d+1) = \delta_i^{(d+1)} \tag{22}$$

for all $i \geq 1$. We can promote this to a similar inequality for $p_i^{(d)}$ using the recurrence from eq. (16). Indeed, this recurrence together with the strict monotonicity of P in both arguments implies that

$$p_i^{(d)} = P(\delta_{i-1}^{(d)}, d) > P(\delta_{i-1}^{(d+1)}, d+1) = p_i^{(d+1)} \tag{23}$$

for all $i \geq 1$. □

3.2 Asymptotics of the error parameter

Our goal is to upper bound the error sequence $\delta_i^{(d)}$ as a function of $\delta = \delta_0^{(d)}$, d, and i. We also define

$$\delta_i^{(\infty)} := \lim_{d\to\infty} \delta_i^{(d)} \tag{24}$$

for all $i \geq 0$. We can gain some insight and motivate our analysis by examining numerical plots of $\delta_i^{(d)}$ as functions of i, for some representative values of d and a common $\delta_0^{(d)}$ that is very close to 1, as shown in Fig. 2. To understand the error sequence with a smaller value of $\delta_0^{(d)}$, one can simply translate the plot horizontally, starting at a value of i corresponding to the desired initial error.

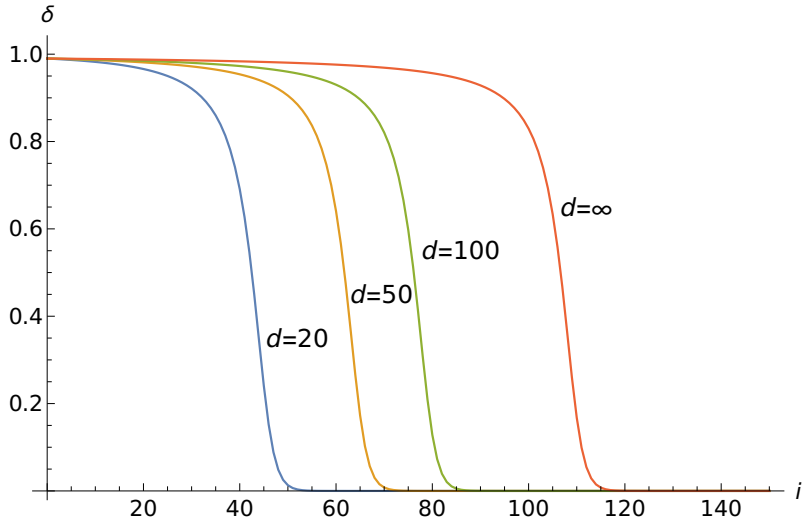


Figure 2: Numerical plots of $\delta_i^{(d)}$ for $d = 20, 50, 100$, and of $\delta_i^{(\infty)}$, with initial error $\delta_0^{(d)} = 0.99$.

For each d, the sequence $\delta_i^{(d)}$ features a slow decline from the initial value $\delta \approx 1$, until it drops to about 0.9. Then the sequence drops rapidly to about 0.05 before leveling off and decreasing exponentially. We can establish asymptotic bounds by separately considering the subsequences with $\delta_i^{(d)} \geq 1/2$ and $\delta_i^{(d)} < 1/2$. However, tighter bounds can be obtained by avoiding the threshold $\delta_i^{(d)} \approx 1/2$. Thus we consider the subsequences with $\delta_i^{(d)} \leq 1/3$, $1/3 < \delta_i^{(d)} \leq 2/3$, and $2/3 < \delta_i^{(d)}$. Depending on how small δ is, we can combine some or all of the above estimates to obtain the number of iterations needed to suppress the error from δ to ϵ for arbitrary d. We then use these results to bound the sample complexity

in Section 3.3. Note that the thresholds 1/3 and 2/3 are arbitrary, and can be further optimized as a function of d, but we leave such fine tuning to the interested reader.

Our analysis exploits the monotonicity established in Section 3.1: for a common $\delta_0^{(d)}$, for all i, d, we have $\delta_i^{(d)} \le \delta_i^{(d+1)} \le \delta_i^{(\infty)}$, since $\Delta(\delta_i^{(d)}, d)$ is an increasing function of d and of $\delta_i^{(d)}$. We divide our analysis into parts corresponding to the three subsequences:

1. We establish a tight upper bound for $\delta_i^{(d)}$ for $\delta_0^{(d)} = \delta \le 1/3$ for all $d \ge 2$. This bound is roughly exponentially decreasing in i.
2. For any $d \ge 2$, we show that 3 to 5 iterations are enough to reduce $\delta_i^{(d)}$ from 2/3 to 1/3.
3. For $\delta_0^{(d)} > 2/3$, and specifically for $\delta_0^{(d)} \approx 1$, we are interested in upper bounds on i^* so that $\delta_{i^*}^{(d)} \approx 2/3$. We further divide this step into two parts.
   (a) First, we upper bound i^* for $\delta_0^{(\infty)}$. Due to Lemma 4, the same upper bound holds for other $\delta_0^{(d)}$. Combining parts 1, 2, and 3(a) (in Section 3.2.1, Section 3.2.2, and Section 3.2.3, respectively), we can upper bound $\delta_i^{(d)}$ for all δ, i, and d. In particular, we estimate the number of iterations required (as a function of $\delta = \delta_0$ and ϵ) to purify copies of $\rho(\delta)$ into $\rho(\epsilon)$. The number of iterations is a gentle function of δ and ϵ, and the large-d limit only introduces an additive constant.
   (b) Then we tighten the estimate of i^* for $\delta_i^{(d)}$ with techniques similar to those in part 3(a), but with more complex arithmetic. The resulting estimate applies to all $d \ge 2$ and $\delta > 2/3$. It improves upon the bound in part 3(a) for small $d(1-\delta)$, and approaches the bound in part 3(a) for large $d(1-\delta)$.

3.2.1 Part 1

If the initial error parameter $\delta_0^{(d)}$ is sufficiently small, subsequent values $\delta_i^{(d)}$ decrease roughly exponentially in i.

Lemma 5. *For all $d \ge 2$, if $\delta = \delta_0^{(d)} \le 1/2$, then for all $i \ge 0$,*

$$\delta_i^{(d)} \le \frac{\delta}{2^i(1-2\delta)+2\delta}. \tag{25}$$

Proof. We upper bound $\delta_i^{(d)}$ by another sequence η_i defined as follows:

$$\eta_0 := \delta, \quad \eta_i := \frac{\eta_{i-1}}{2-2\eta_{i-1}} \quad \text{for } i \ge 1. \tag{26}$$

To see that $\delta_i^{(d)} \le \eta_i$ for all $i \ge 0$, suppose $\delta = \delta_0^{(d)} = \delta_0^{(\infty)} = \eta_0$. From Lemma 4, $\delta_i^{(d)} \le \delta_i^{(\infty)}$ for all $i \ge 0$. Let us now prove by induction that $\delta_i^{(\infty)} \le \eta_i$ for all $i \ge 0$. The base case is immediate. The induction hypothesis says $\delta_r^{(\infty)} \le \eta_r$ for some $r \ge 0$, so from eq. (16), we have

$$\delta_{r+1}^{(\infty)} = \Delta(\delta_r^{(\infty)}, \infty) \le \Delta(\eta_r, \infty) \tag{27}$$

$$= \frac{\eta_r}{2-2\eta_r+\eta_r^2} \le \frac{\eta_r}{2-2\eta_r} = \eta_{r+1} \tag{28}$$

where the second line comes from eq. (15). Altogether, this shows $\delta_i^{(d)} \le \eta_i$ for all $i \ge 0$.

The recurrence η_i admits an exact solution. Indeed, note that

$$\frac{1}{\eta_i} = \frac{2}{\eta_{i-1}} - 2, \tag{29}$$

which yields the following closed-form expression for η_i:

$$\frac{1}{\eta_i} = 2^i\left(\frac{1}{\eta_0} - 2\right) + 2. \tag{30}$$

Inverting and substituting $\eta_0 = \delta$, we find that

$$\eta_i = \frac{\delta}{2^i(1-2\delta)+2\delta}. \tag{31}$$

and the result follows. □

Note that under the assumption $\delta \leq 1/2$, we have $0 \leq \eta_i \leq 1$ for all $i \geq 0$.

To suppress the error from $\delta \in (0, 1/2)$ to ϵ, since $\eta_i < \frac{\delta}{2^i(1-2\delta)}$, it suffices to iterate $\log_2 \frac{\delta}{(1-2\delta)\epsilon}$ times. In particular, for $\delta \leq 1/3$, it suffices to iterate $\log_2 \frac{1}{\epsilon}$ times.

3.2.2 Part 2

Using monotonicity from Lemma 4 and by considering $\delta_i^{(\infty)}$, a brute-force calculation shows that 5 iterations are sufficient to suppress the error from at most $2/3$ to at most $1/3$ for all $d \geq 2$. For smaller values of d, as few as 3 iterations suffice.

3.2.3 Part 3(a)

For $\delta \approx 1$, Fig. 2 suggests that the error suppression is slow until after about $\frac{1}{1-\delta}$ iterations, when the error rapidly drops from 0.9 to 0.1. We now study how quickly $\delta_i^{(d)}$ can be upper bounded by $2/3$ starting from an arbitrary $\delta_0^{(d)} \approx 1$.

We first consider the simpler sequence $\delta_i^{(\infty)}$. Note from eq. (17) that

$$\lim_{d\to\infty} \Delta(x, d) = \frac{x}{2 - 2x + x^2}, \tag{32}$$

so $\delta_i^{(\infty)}$ obeys the following recurrence:

$$\delta_0^{(\infty)} := \delta, \quad \delta_i^{(\infty)} := \frac{\delta_{i-1}^{(\infty)}}{2 - 2\delta_{i-1}^{(\infty)} + \delta_{i-1}^{(\infty)2}} \quad \text{for } i \geq 1. \tag{33}$$

Because the function $(2 - 2x + x^2)^{-1}$ is monotonically increasing for $x \in [0, 1]$, and $\delta_i^{(\infty)}$ is monotonically decreasing in i, the following upper bound holds by induction:

$$\delta_i^{(\infty)} \leq \frac{\delta_{i-1}^{(\infty)}}{2 - 2\delta + \delta^2} \leq \frac{\delta}{(2 - 2\delta + \delta^2)^i}. \tag{34}$$

If $\delta > 2/3$ is constant, the above is an exponentially decreasing function in i, which may be sufficient for many purposes. However, this bound does not have tight dependence on δ.

We can instead bound $\delta_i^{(\infty)}$ by considering two other recurrences, defined as follows. Let i^* be the smallest i so that $\delta_{i^*+1}^{(\infty)} < 2/3$. We first define

$$\kappa_i^{(\infty)} = 1 - \delta_i^{(\infty)}, \tag{35}$$

so that

$$\kappa_0^{(\infty)} = 1 - \delta, \quad \kappa_i^{(\infty)} = 1 - \delta_i^{(\infty)} = 1 - \frac{\delta_{i-1}^{(\infty)}}{2 - 2\delta_{i-1}^{(\infty)} + \delta_{i-1}^{(\infty)2}} = \frac{\kappa_{i-1}^{(\infty)} + \kappa_{i-1}^{(\infty)2}}{1 + \kappa_{i-1}^{(\infty)2}}. \tag{36}$$

Let

$$g^{(\infty)}(x) = \frac{x + x^2}{1 + x^2}, \tag{37}$$

so $\kappa_i^{(\infty)} = g^{(\infty)}(\kappa_{i-1}^{(\infty)})$. Furthermore, for $i \in \{0, 1, \ldots, i^*\}$, define an inverse recurrence for $\kappa_i^{(\infty)}$ as

$$\mu_0^{(\infty)} = \kappa_{i^*}^{(\infty)}, \quad \mu_i^{(\infty)} = h^{(\infty)}(\mu_{i-1}^{(\infty)}) \tag{38}$$

where $h^{(\infty)}(y)$ is the inverse of $y = g^{(\infty)}(x) = \frac{x+x^2}{1+x^2}$ (which is well-defined for all $x \in [0, 1]$), so indeed, $\mu_i^{(\infty)} = \kappa_{i^*-i}^{(\infty)}$. In particular, $\mu_{i^*}^{(\infty)} = 1 - \delta$. The definition of i^* and the relationships between $\delta_i^{(d)}, \kappa_i^{(d)}, \mu_i^{(d)}$ for a representative $d = 20$ are summarized in Fig. 3.

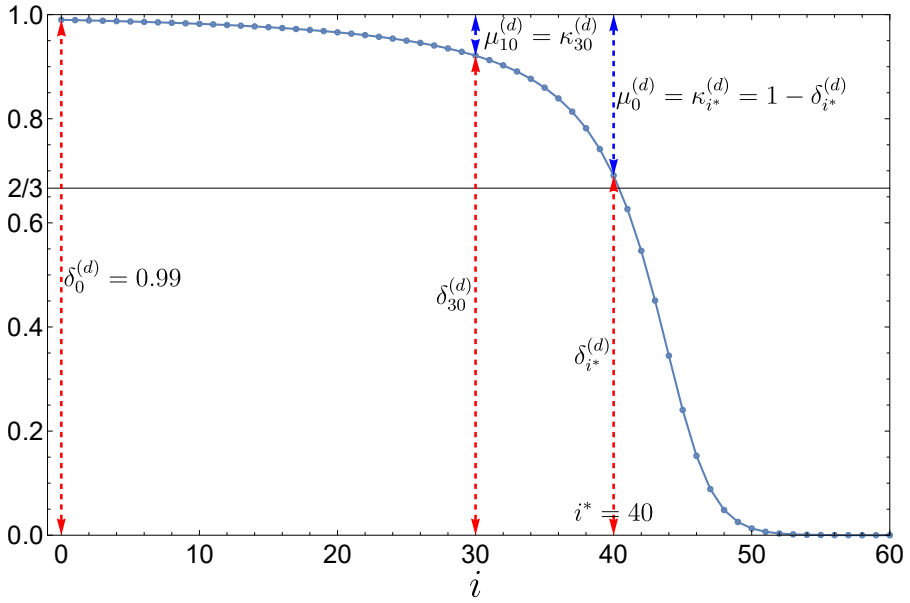


Figure 3: A diagrammatic summary of the definition of i^*, and the relationships between $\delta_i^{(d)}, \kappa_i^{(d)}$, and $\mu_{i^*-i}^{(d)}$, illustrated for $d = 20$ and $\delta = 0.99$. The light blue curve shows $\delta_i^{(d)}$ as a function of i. Since i^* is the smallest i so that $\delta_{i+1}^{\infty} < 2/3$, in this case, $i^* = 40$. For each i, $\kappa_i^{(d)} = 1 - \delta_i^{(d)}$ is the distance from the light blue curve to the value 1. Finally, $\mu_i^{(d)} = \kappa_{i^*-i}$ is a recurrence that iterates from the right to the left (starting from i^*). Note that the index j for $\mu_j^{(d)}$ is related to the label i for the abscissa as $j + i = i^*$.

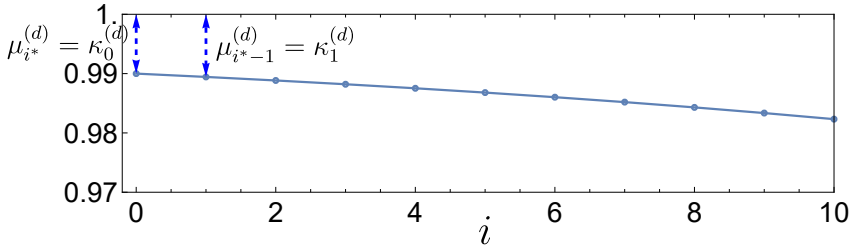


Figure 4: Zoomed-in view of the upper left corner of Fig. 3.

The main idea of the analysis is as follows. We will derive upper bounds for $\mu_i^{(\infty)}$ as a function of i. In particular, if we prove an upper bound υ for $\mu_i^{(\infty)}$, we have a lower bound $1-\upsilon$ for $\delta_{i^*-i}^{(\infty)}$, which means that $i+1$ iterations are sufficient to reduce $\delta_{i^*-i}^{(\infty)}$ from at least $1-\upsilon$ to $\delta_{i^*+1}^{(\infty)} < 2/3$. In particular, if n satisfies $\mu_n^{(\infty)} \le 1-\delta$, then $n+1$ iterations are sufficient to reduce the initial error $\delta_0^{(\infty)} = \delta$ to $\delta_{n+1}^{(\infty)} < 2/3$, so $i^* < n$. In particular, we show the following.[1]

Lemma 6. *If* $\mu_0^{(\infty)} \le 1/3$*, then for all* $i \ge 1$*,*

$$\mu_i^{(\infty)} < \frac{1}{i} + \frac{2\ln i}{i^2}. \tag{39}$$

Proof. The proof consists of several claims. In Claim 2, we obtain an inequality which can be used to upper bound $\mu_i^{(\infty)}$ using an upper bound for $\mu_{i-1}^{(\infty)}$. Then, we start with a crude upper bound on all $\mu_k^{(\infty)}$ in Claim 1, and recursively improve the upper bound in Claims 3 to 5, using Claim 2.

By taking the derivative of $g^{(\infty)}(x)$ defined in eq. (37), we can verify that $g^{(\infty)}(x)$ is *strictly* increasing, continuous, and bounded on $[0, 2.4]$. Therefore, its inverse $h^{(\infty)}(y)$ has the same properties on $[0, 1.2]$. We can also obtain a closed-form expression for $h^{(\infty)}(y)$ by inverting $y = \frac{x+x^2}{1+x^2}$ using the quadratic formula,

$$x = \frac{-1 + \sqrt{1 + 4y(1-y)}}{2(1-y)}, \tag{40}$$

keeping only the solution corresponding to the case of interest, $0 < x, y < 1/3$ (because $1/3 > \mu_0^{(\infty)}$ and $\mu_i^{(\infty)}$ decreases with i while remaining positive).

Claim 1. *We have* $\mu_1^{(\infty)} \le 0.2808$*,* $\mu_2^{(\infty)} \le 0.2396$*, and* $\mu_i^{(\infty)} \le 1/10$ *for all* $i \ge 10$*.*

Proof of Claim 1. This follows from the monotonicity of $h^{(\infty)}$ and its closed-form expression, using $\mu_0^{(\infty)} \le 1/3$. (*Claim*) □

[1] In version 1 of the arXiv preprint of this manuscript, we invoked a theorem from [Ste96] (case 2), which implies

$$\mu_i^{(\infty)} = \frac{1}{i} + \frac{2\ln i}{i^2} + o\Big(\frac{\ln i}{i^2}\Big),$$

and provided a self-contained derivation using Taylor series expansion in $\mu_i^{(\infty)}$. The strict bound in Lemma 6 is obtained by different techniques, most crucially by strengthening Eq. (54) in version 1 to the inequality in Claim 2, and by replacing the approximation $\mu_i^{(\infty)} \approx \frac{1}{i}$ with recursively better upper bounds in Claims 3 to 5, using Claim 2.

Claim 2. *For all $i \geq 3$,*

$$\frac{1}{\mu_i^{(\infty)}} > \frac{1}{\mu_{i-1}^{(\infty)}} + 1 - 2\mu_{i-1}^{(\infty)}. \tag{41}$$

Proof of Claim 2. We can rephrase the claim as

$$\mu_i^{(\infty)} = h^{(\infty)}(\mu_{i-1}^{(\infty)}) < \frac{1}{\frac{1}{\mu_{i-1}^{(\infty)}} + 1 - 2\mu_{i-1}^{(\infty)}}. \tag{42}$$

For $i \geq 3$, $\frac{1}{\frac{1}{\mu_{i-1}^{(\infty)}}+1-2\mu_{i-1}^{(\infty)}} < 0.22$, so $g^{(\infty)}$ is invertible. Inserting the identity function $h^{(\infty)} \circ g^{(\infty)}$ on the right-hand side and using the monotonicity of $h^{(\infty)}$, it suffices to show

$$\mu_{i-1}^{(\infty)} < g^{(\infty)}\left(\frac{1}{\frac{1}{\mu_{i-1}^{(\infty)}} + 1 - 2\mu_{i-1}^{(\infty)}}\right). \tag{43}$$

The above steps translate the analysis from $h^{(\infty)}$ (involving a square root) to $g^{(\infty)}$ (which involves simpler rational expressions and squares).

For $x > 0$, $g^{(\infty)}(x) = \frac{x+x^2}{1+x^2} = \frac{1+\frac{1}{x}}{1+\frac{1}{x^2}}$, so eq. (43) is equivalent to

$$0 < \frac{1 + \left(\frac{1}{\mu_{i-1}^{(\infty)}} + 1 - 2\mu_{i-1}^{(\infty)}\right)}{1 + \left(\frac{1}{\mu_{i-1}^{(\infty)}} + 1 - 2\mu_{i-1}^{(\infty)}\right)^2} - \mu_{i-1}^{(\infty)}, \tag{44}$$

or equivalently

$$0 < \left(\frac{1}{\mu_{i-1}^{(\infty)}} + 2 - 2\mu_{i-1}^{(\infty)}\right) - \mu_{i-1}^{(\infty)} - \mu_{i-1}^{(\infty)}\left(\frac{1}{\mu_{i-1}^{(\infty)}} + 1 - 2\mu_{i-1}^{(\infty)}\right)^2 = 4\mu_{i-1}^{(\infty)2}\left(1 - \mu_{i-1}^{(\infty)}\right), \tag{45}$$

which clearly holds. (*Claim*) □

Note that Claim 2 shows an upper bound on $\mu_i^{(\infty)}$ using any upper bound on $\mu_{i-1}^{(\infty)}$. We use this strategy to recursively establish tight bounds in the following claims.

Claim 3. *For all $i \geq 3$,*

$$\frac{1}{\mu_i^{(\infty)}} > \frac{4}{5}\left(i + \frac{5}{2}\right). \tag{46}$$

Proof of Claim 3. The claim can be directly verified for $1 \leq i \leq 10$ using the upper bound $\mu_0^{(\infty)} \leq 1/3$ and the closed-form expression for $h^{(\infty)}$. For $i \geq 11$, using Claim 1, $1 - 2\mu_{i-1}^{(\infty)} \geq \frac{4}{5}$. Substituting this into Claim 2 gives

$$\frac{1}{\mu_i^{(\infty)}} > \frac{1}{\mu_{i-1}^{(\infty)}} + \frac{4}{5} > \frac{1}{\mu_{i-2}^{(\infty)}} + 2 \times \frac{4}{5} > \cdots > \frac{1}{\mu_{10}^{(\infty)}} + (i-10) \times \frac{4}{5} > \frac{4}{5}\left(i + \frac{5}{2}\right), \tag{47}$$

where we have also used Claim 1 for $\mu_{10}^{(\infty)}$ in the last step. (*Claim*) □

Claim 3 improves the bound in Claim 1. We can again use Claim 2, now together with Claim 3, to get another improved bound.

Claim 4. *For all* $i \geq 1$,

$$\mu_i^{(\infty)} < \frac{1}{i} + \frac{5}{2}\frac{\ln i}{i^2}. \tag{48}$$

Proof of Claim 4. The $i = 1, 2$ cases follow from Claim 1. Using Claims 2 and 3, for all $i \geq 3$,

$$\frac{1}{\mu_i^{(\infty)}} > \frac{1}{\mu_{i-1}^{(\infty)}} + 1 - 2\mu_{i-1}^{(\infty)} > \frac{1}{\mu_{i-1}^{(\infty)}} + 1 - \frac{5}{2}\frac{1}{\left(i + \frac{5}{2}\right)} > \frac{1}{\mu_2^{(\infty)}} + (i-2) - \frac{5}{2}\sum_{k=2}^{i-1}\frac{1}{\left(k + \frac{5}{2}\right)}. \tag{49}$$

Note that

$$\frac{1}{x + \frac{1}{2}} \leq \ln(x+1) - \ln(x) \text{ for } x \geq 1, \tag{50}$$

so

$$\begin{aligned}
\frac{1}{\mu_i^{(\infty)}} &> \frac{1}{\mu_2^{(\infty)}} + (i-2) - \frac{5}{2}\sum_{k=2}^{i-1}\left[\ln(k+3) - \ln(k+2)\right] && (51)\\
&= \frac{1}{\mu_2^{(\infty)}} + (i-2) - \frac{5}{2}\left[\ln(i+2) - \ln 4\right] && (52)\\
&> i - \frac{5}{2}\ln(i+2) + 5.6408, && (53)\\
&> i - \frac{5}{2}\ln i + 4, && (54)
\end{aligned}$$

where we use the numerical bound of $\mu_2^{(\infty)}$ and $\ln(i+2) - \ln i = \ln(1 + \frac{2}{i}) \leq \ln\frac{5}{3}$ in the last two steps. It remains to show

$$\frac{1}{i - \frac{5}{2}\ln i + 4} < \frac{1}{i} + \frac{5}{2}\frac{\ln i}{i^2}, \tag{55}$$

which is equivalent to

$$16i + 40\ln i > 25\ln^2 i. \tag{56}$$

This holds since $16x > 25\ln^2 x$ for all $x \geq 1$. Indeed, the function $4\sqrt{x} - 5\ln x$ is minimized at $x = \frac{25}{4}$ where it is still positive. (*Claim*) □

We provide the final tightening of the bound that recovers the asymptotic result mentioned in footnote 1 and completes the proof of Lemma 6.

Claim 5. *For all* $i \geq 1$,

$$\mu_i^{(\infty)} < \frac{1}{i} + \frac{2\ln i}{i^2}. \tag{57}$$

Proof of Claim 5. The case of $i \leq 7$ can be checked directly. For all $i \geq 8$, Claims 2 and 4

imply

$$\frac{1}{\mu_i^{(\infty)}} > \frac{1}{\mu_{i-1}^{(\infty)}} + 1 - 2\mu_{i-1}^{(\infty)} > \frac{1}{\mu_{i-1}^{(\infty)}} + 1 - \frac{2}{i-1} - \frac{5\ln(i-1)}{(i-1)^2} \tag{58}$$

$$> \frac{1}{\mu_7^{(\infty)}} + (i-7) - \left(\sum_{k=7}^{i-1} \frac{2}{k} + \frac{5\ln k}{k^2}\right) \tag{59}$$

$$> \frac{1}{\mu_7^{(\infty)}} + (i-7) + 2\ln(6.5) - 2\ln(i-0.5) - \frac{5}{6}(1+\ln 6) \tag{60}$$

$$> i - 2\ln(i) + 2, \tag{61}$$

where we have again used the inequality $\frac{1}{(x+\frac{1}{2})} \leq \ln(x+1) - \ln(x)$, and also the bound $\sum_{k=7}^{\infty} \frac{\ln k}{k^2} < \int_6^{\infty} \frac{\ln x}{x^2} dx = (1+\ln 6)/6$. It remains to show that

$$\frac{1}{i - 2\ln(i) + 2} < \frac{1}{i} + \frac{2\ln i}{i^2}. \tag{62}$$

This is equivalent to $i + 2\ln(i) > 2\ln^2(i)$, which indeed holds for all $i \geq 1$ as one can check by taking derivatives. (*Claim*) □

This completes the proof of the lemma. □

We can obtain an upper bound on the number of iterations n that is sufficient to suppress any initial noise $\delta \in (2/3, 1)$ to less than $1/3$.

Lemma 7. *For any $\delta \in (2/3, 1)$, if an integer n satisfies*

$$n \geq \frac{1}{1-\delta}\left(1 + 2(1-\delta)\ln\left(\frac{1}{1-\delta}\right)\right) = \frac{1}{1-\delta} + 2\ln\left(\frac{1}{1-\delta}\right), \tag{63}$$

then $\mu_n^{(\infty)} \leq 1 - \delta$, and thus $\delta_{n+1}^{(\infty)} < 2/3$.

Proof. Using Lemma 6,

$$\mu_n^{(\infty)} < \frac{1}{n}\left(1 + \frac{2\ln n}{n}\right) < \frac{1}{n}\left(1 + 2(1-\delta)\ln\frac{1}{1-\delta}\right) \leq 1 - \delta, \tag{64}$$

where the second inequality holds because $n > \frac{1}{1-\delta}$ and $\frac{\ln x}{x}$ is decreasing for $x \geq 3$. □

As a comparison, if we use the rough upper bound for $\delta_i^{(\infty)}$ in eq. (34), we conclude that the upper bound is less than $2/3$ after n iterations if

$$n \geq \left\lceil \frac{\ln(3\delta/2)}{\ln(2 - 2\delta + \delta^2)} \right\rceil. \tag{65}$$

When $1 - \delta \ll 1$, $\ln(2 - 2\delta + \delta^2) = \ln(1 + (1-\delta)^2) \approx (1-\delta)^2$, so the upper bound on the number of iterations is roughly

$$n \geq \left\lceil \frac{\ln(3\delta/2)}{(1-\delta)^2} \right\rceil, \tag{66}$$

with quadratically worse dependence on $\frac{1}{1-\delta}$.

Combining parts 1, 2, and 3(a), we have established the following estimate. For all $d \geq 2$, $\delta > 2/3$, and ϵ with $\delta > \epsilon > 0$, the error can be reduced from δ to ϵ with at most

$$\log_2 \frac{1}{\epsilon} + 5 + \frac{1}{1-\delta} + 2\ln\left(\frac{1}{1-\delta}\right) \tag{67}$$

iterations.

3.2.4 Part 3(b)

We now provide a more refined analysis[2] of $\delta_i^{(d)}$ for finite d and $\delta \approx 1$. The method is similar to Part 3(a) above, with more complex arithmetic and one more step to account for a factor that is exponentially decaying in i. Wherever appropriate, we omit straightforward calculations, and ideas covered in Part 3(a) are described in less detail.

The goal is to upper bound the recurrence

$$\delta_0^{(d)} = \delta, \quad \delta_i^{(d)} = \Delta(\delta_{i-1}^{(d)}, d) = \frac{\delta_{i-1}^{(d)} + \frac{1}{d}\,\delta_{i-1}^{(d)2}}{2\Big(1 - \Big(1 - \frac{1}{d}\Big)\delta_{i-1}^{(d)} + \frac{1}{2}\Big(1 - \frac{1}{d}\Big)\delta_{i-1}^{(d)2}\Big)} \tag{68}$$

stated in eqs. (16) and (17). As in part 3(a), let i^* be the smallest integer so that $\delta_{i^*+1}^{(d)} < 2/3$. For $i \in \{0, 1, \ldots, i^*\}$, let

$$\kappa_i^{(d)} = 1 - \delta_i^{(d)}, \tag{69}$$

which implies

$$\kappa_0^{(d)} = 1 - \delta, \quad \kappa_i^{(d)} = 1 - \delta_i^{(d)} = \frac{\Big(1 + \frac{2}{d}\Big)\kappa_{i-1}^{(d)} + \Big(1 - \frac{2}{d}\Big)\kappa_{i-1}^{(d)2}}{\Big(1 + \frac{1}{d}\Big) + \Big(1 - \frac{1}{d}\Big)\kappa_{i-1}^{(d)2}}, \tag{70}$$

where we have first expressed $\delta_i^{(d)}$ in terms of $\delta_{i-1}^{(d)}$ and then substituted $\delta_{i-1}^{(d)} = 1 - \kappa_{i-1}^{(d)}$ to obtain the recurrence. Let $g^{(d)}(x) = \frac{\left(1+\frac{2}{d}\right)x + \left(1-\frac{2}{d}\right)x^2}{\left(1+\frac{1}{d}\right)+\left(1-\frac{1}{d}\right)x^2}$ so that $\kappa_i^{(d)} = g^{(d)}(\kappa_{i-1}^{(d)})$. Taking the derivative of $g^{(d)}(x)$, we can show that $g^{(d)}(x)$ is strictly increasing if $1 + \Big(\frac{d-2}{d+2}\Big)2x - \Big(\frac{d-1}{d+1}\Big)x^2 > 0$. Let S be the set of x satisfying this condition. Note that $[0, 1] \subset S$ for all $d \geq 2$. On the set $g^{(d)}(S)$, define an inverse $h^{(d)}$ of $g^{(d)}$. For $i \in \{0, 1, \ldots, i^*\}$, define an inverse recurrence for $\kappa_i^{(d)}$ as

$$\mu_0^{(d)} = \kappa_{i^*}^{(d)}, \quad \mu_i^{(d)} = h^{(d)}(\mu_{i-1}^{(d)}), \tag{71}$$

so indeed, $\mu_i^{(d)} = \kappa_{i^*-i}^{(d)}$. In particular, $\mu_{i^*}^{(d)} = 1 - \delta$. The definition of i^* and the relations between $\delta_i^{(d)}, \kappa_i^{(d)}, \mu_i^{(d)}$ for a representative $d = 20$ are summarized in Fig. 3.

We can now state the finite-d analogue of Claim 2 in Part 3(a).

Claim 2$^{(d)}$. *For all $i \geq 1$, $d \geq 2$,*

$$\frac{1}{\mu_i^{(d)}} > \frac{1}{a}\,\frac{1}{\mu_{i-1}^{(d)}} + b - 2\,c\,\mu_{i-1}^{(d)}, \tag{72}$$

where

$$a = \frac{d+1}{d+2}, \quad b = \frac{d-2}{d+2}, \quad c = \frac{d^3}{(d+2)^3} \;\; \text{if } d \geq 3, \quad c = \frac{1}{7} \;\; \text{if } d = 2. \tag{73}$$

[2] In the current analysis, Claim 2$^{(d)}$ is the finite-d analogue of Claim 2 in Part 3(a). One can show that Claim 2$^{(d)}$ has the same coefficients as in eq. (74) of Part 3(b) in version 1 of the arXiv preprint of this manuscript. Whereas the latter is an asymptotic expression, Claim 2$^{(d)}$ is a strict inequality. Combining $\delta_i^{(d)} \leq \delta_i^{(\infty)}$ from Part 3(a) and Claim 2$^{(d)}$ gives a bound on the number of iterations that subsumes the bounds in Parts 3(b) and (c) of version 1 of the arXiv preprint, and asymptotically attains the conjectured expression in eq. (99) there.

Proof. Noting the positivity of the right-hand side of eq. (72), we can rephrase our claim as

$$\mu_i^{(d)} = h^{(d)}(\mu_{i-1}^{(d)}) < \frac{1}{\frac{1}{a}\,\frac{1}{\mu_{i-1}^{(d)}} + b - 2\,c\,\mu_{i-1}^{(d)}}\,. \tag{74}$$

Using $a \le 1$, $\mu_{i-1}^{(d)} \le 1/3$, $b \ge 0$, and $c \le 1$, the right-hand side of eq. (74) can be upper bounded by $3/7$, which is within the range S where $g^{(d)}$ is strictly increasing, and thus invertible, so we can insert the identity function $h^{(d)}g^{(d)}$ and apply the monotonicity of $h^{(d)}$ (which follows from that of $g^{(d)}$) to obtain a statement equivalent to eq. (74):

$$\mu_{i-1}^{(d)} < g^{(d)}\left(\frac{1}{\frac{1}{a}\,\frac{1}{\mu_{i-1}^{(d)}} + b - 2\,c\,\mu_{i-1}^{(d)}}\right). \tag{75}$$

As in Part 3(a), the above steps translate the analysis from $h^{(d)}$ to $g^{(d)}$. Furthermore, for $x > 0$, $g^{(d)}(x) = \frac{\left(1+\frac{2}{d}\right)x+\left(1-\frac{2}{d}\right)x^2}{\left(1+\frac{1}{d}\right)+\left(1-\frac{1}{d}\right)x^2} = \frac{(d+2)\frac{1}{x}+(d-2)}{(d+1)\frac{1}{x^2}+(d-1)}$, so eq. (75) is equivalent to

$$\mu_{i-1}^{(d)} < \frac{(d+2)\left(\frac{1}{a}\,\frac{1}{\mu_{i-1}^{(d)}} + b - 2\,c\,\mu_{i-1}^{(d)}\right) + (d-2)}{(d+1)\left(\frac{1}{a}\,\frac{1}{\mu_{i-1}^{(d)}} + b - 2\,c\,\mu_{i-1}^{(d)}\right)^2 + (d-1)}\,, \tag{76}$$

which in turn is equivalent to

$$\begin{aligned} 0 \;&< (d+2)\left(\frac{1}{a}\,\frac{1}{\mu_{i-1}^{(d)}} + b - 2\,c\,\mu_{i-1}^{(d)}\right) \\ &\qquad + (d-2) - \mu_{i-1}^{(d)}\left((d+1)\left(\frac{1}{a}\,\frac{1}{\mu_{i-1}^{(d)}} + b - 2\,c\,\mu_{i-1}^{(d)}\right)^2 + (d-1)\right) \\ &= 2c(d+2)\mu_{i-1}^{(d)} - (d-1)\mu_{i-1}^{(d)} - b^2(d+1)\mu_{i-1}^{(d)} - 4c^2(d+1)\mu_{i-1}^{(d)3} + 4bc(d+1)\mu_{i-1}^{(d)2} \\ &= \left(2c(d+2) - (d-1) - b^2(d+1)\right)\mu_{i-1}^{(d)} + \left(b - c\mu_{i-1}^{(d)}\right)4c(d+1)\mu_{i-1}^{(d)2}, \end{aligned} \tag{77}$$

where the second line is obtained from expanding the first line and using the expressions for a, b to cancel out some of the terms. For $d \ge 3$, the choice of c gives $2c(d+2) - (d-1) - b^2(d+1) = 0$, and since $\mu_{i-1}^{(d)} < 1/3$, we have $b - c\mu_{i-1}^{(d)} > 0$, so the claimed inequality holds. For $d = 2$, $b = 0$. One can substitute the actual values of b, c, d and use $\mu_{i-1}^{(d)} < 1/3$ to verify the above inequality. □

Examining Claim $2^{(d)}$, and observing that $\mu_{i-1}^{(d)} \le \mu_{i-1}^{(\infty)} \to 0$ as $i \to \infty$, the term linear in c has less effect on bounding $\mu_{i-1}^{(d)}$ for large i. Thus we expect $\mu_i^{(d)}$ to decrease at least as quickly as $a^i\mu_0^{(d)}$, with additional suppression due to the term linear in b. This motivates a change of variables

$$\mu_i^{(d)} = a^i\nu_i^{(d)}, \tag{78}$$

and correspondingly, Claim $2^{(d)}$ becomes

$$\frac{1}{\nu_i^{(d)}} > \frac{1}{\nu_{i-1}^{(d)}} + a^i b - 2\,a^i c\,\mu_{i-1}^{(d)}\,. \tag{79}$$

Repeated application of the above gives

$$\frac{1}{\nu_i^{(d)}} > \frac{1}{\nu_0^{(d)}} + b\sum_{k=1}^{i} a^k - 2\,c\sum_{k=1}^{i} a^k \mu_{k-1}^{(d)}\,. \tag{80}$$

Recall that a lower bound on $\frac{1}{\nu_i^{(d)}}$ gives an upper bound on $\nu_i^{(d)}$, and in turn on $\mu_i^{(d)}$. To this end, we take the following steps:

1. We sum the second term of the right-hand side of eq. (80), and apply the upper bound $\mu_k^{(d)} \le \mu_k^{(\infty)} \le \frac{1}{k} + \frac{5}{2}\frac{\ln k}{k^2}$ for $k \ge 1$ from Part 3(a) and $\mu_0^{(d)} \le 1/3$ to the last term to obtain

$$\frac{1}{\nu_i^{(d)}} > \frac{1}{\nu_0^{(d)}} + b\,a\,\frac{1-a^i}{1-a} - 2\,c\sum_{k=1}^{i-1} a^{k+1}\left(\frac{1}{k} + \frac{5}{2}\frac{\ln k}{k^2}\right) - \frac{2}{3}\,c\,a. \tag{81}$$

2. For $0 \le a < 1$, we can upper bound $\sum_{k=1}^{i-1}\frac{a^k}{k}$ by $f(a) = \sum_{k=1}^{\infty}\frac{a^k}{k}$. Note that $\frac{df}{da} = \sum_{k=1}^{\infty} a^{k-1} = \frac{1}{1-a}$, so integrating and using $f(0) = 0$, we find $f(a) = -\ln(1-a) = \ln(d+2) \lesssim \ln(d) + 0.7$.

3. We can also upper bound $\sum_{k=1}^{i-1}\frac{a^k}{k}$ by $\sum_{k=1}^{i-1}\frac{1}{k} \le 1 + \sum_{k=2}^{i-1}\ln(k+\frac{1}{2}) - \ln(k-\frac{1}{2}) = 1 + \ln(i-\frac{1}{2}) - \ln(1.5) < \ln(i-\frac{1}{2}) + 0.6$, where we have used eq. (50).

4. First, $\sum_{k=1}^{i-1} a^{k-1}\frac{\ln k}{k^2} < \sum_{k=2}^{i-1}\frac{\ln k}{k^2} \lesssim 0.1733 + \sum_{k=3}^{i-1}\frac{\ln k}{k^2}$. Then, since $\frac{\ln x}{x^2}$ is decreasing for $x \ge 2$, we have, for all $k \ge 3$, $\frac{\ln k}{k^2} < \int_{k-1}^{k}\frac{\ln x}{x^2}dx$, so $\sum_{k=3}^{i-1}\frac{\ln k}{k^2} \le \int_2^{\infty}\frac{\ln x}{x^2}dx$. Finally, $\int\frac{\ln x}{x^2}dx = -\frac{1+\ln x}{x} + \text{constant}$, so $\int_2^{\infty}\frac{\ln x}{x^2}dx \lesssim 0.8466$. Altogether, we have $\sum_{k=1}^{i-1} a^{k-1}\frac{\ln k}{k^2} < 1.02$.

5. Substituting the bounds from steps 2–4 into eq. (81),

$$\frac{1}{\nu_i^{(d)}} > \frac{1}{\nu_0^{(d)}} + b\,a\,\frac{1-a^i}{1-a} - 2\,c\,a\,\ln(\min\{d,i\}) - 1.4\,c\,a - 5.1\,c\,a^2 - \frac{2}{3}\,c\,a \tag{82}$$

$$> b\,a\,\frac{1-a^i}{1-a} - 2\,c\,a\,\ln(\min\{d,i\}) + 3 - 7.2\,c\,a. \tag{83}$$

Note that eq. (83) holds for all $i \ge 1$. The cases $i = 1, 2$ can be verified directly. Steps 2–4 hold for all $i \ge 3$, except for the expression $\sum_{k=3}^{i-1}\frac{\ln k}{k^2}$ in step 4, which can be set to 0 for $i = 3$ without changing the conclusion. Reverting the change of variables $\mu_i^{(d)} = a^i\nu_i^{(d)}$, we have

$$\frac{1}{\mu_i^{(d)}} > a^{-i}\left(b\,a\,\frac{1-a^i}{1-a} - 2\,c\,a\,\ln(\min\{d,i\}) + 3 - 7.2\,c\,a\right) \tag{84}$$

$$> a^{-i}\left(\frac{ba}{1-a} - 2\,c\,a\,\ln(\min\{d,i\}) + 3 - 7.2\,c\,a\right) - \frac{ba}{1-a}. \tag{85}$$

Recall that $\mu_{i^*}^{(d)} = 1-\delta$, and our ultimate goal is an upper bound on i^*. Also recall that i^*+1 iterations reduce $\delta_0^{(d)} = \delta$ to $\delta_{i^*+1}^{(d)} < \frac{2}{3}$. Furthermore, $i^* \le n^{(\infty)*} := \frac{1}{1-\delta} + 2\ln\left(\frac{1}{1-\delta}\right)$ (the upper bound for i^* for the $d \to \infty$ case in Lemma 7). Incorporating all these observations into eq. (85), we obtain

$$\frac{1}{1-\delta} = \frac{1}{\mu_{i^*}^{(d)}} > a^{-i^*}\left(\frac{ba}{1-a} - 2\,c\,a\,\ln\left(\min\{d, n^{(\infty)*}\}\right) + 3 - 7.2\,c\,a\right) - \frac{ba}{1-a}. \tag{86}$$

This allows us to make a transition from the analysis of $\mu_i^{(d)}$ to i^*. Let

$$\alpha = \frac{ba}{1-a} = \frac{(d-2)(d+1)}{d+2}, \quad \beta = \alpha - 2\,c\,a\,\ln\Big(\min\{d, n^{(\infty)*}\}\Big) + 3 - 7.2\,c\,a. \tag{87}$$

To be concrete, for $d = 2,3,4,5$, $(\alpha,\beta) \approx (0,2.08),(0.8,2.18),(1.67,2.20),(2.57,2.32)$, respectively, and as d increases, $\alpha \approx d$ and $\beta \approx d - 2\ln(\min\{d, n^{(\infty)*}\})$. Equation (86) can now be rephrased as

$$\frac{1}{1-\delta} > a^{-i^*}\ \beta - \alpha, \tag{88}$$

or

$$(a^{-1})^{i^*} < \frac{\frac{1}{1-\delta} + \alpha}{\beta}. \tag{89}$$

This gives an upper bound on i^*.

Lemma 8. *For any $\delta \in (2/3, 1)$, if an integer n satisfies*

$$n > \frac{\ln\left(\frac{\frac{1}{1-\delta}+\alpha}{\beta}\right)}{\ln\frac{1}{a}} = \begin{cases} \frac{\ln\frac{1}{(1-\delta)\beta}}{\ln\frac{4}{3}} \approx 3.476\,\ln\frac{1}{1-\delta} - 2.546 & \text{if } d = 2 \\ \frac{\ln\left(1+\frac{1}{\alpha(1-\delta)}\right)+\ln\frac{\alpha}{\beta}}{\ln\left(1+\frac{1}{d+1}\right)} & \text{if } d \geq 3, \end{cases} \tag{90}$$

then $\delta_n^{(d)} < 2/3$.

We can further understand the above lemma by making some approximations. Except for very small dimension d, we have $a, b, c \approx 1$, $\alpha \approx d$, $\beta \approx d - 2\ln\min\Big\{d, \frac{1}{1-\delta}\Big\}$, and $\frac{\alpha}{\beta} \approx 1 + \frac{2}{d}\ln\min\Big\{d, \frac{1}{1-\delta}\Big\}$. For a small constant dimension d and vanishing $1-\delta$, we have $\frac{1}{\alpha(1-\delta)} \gg 1$. Therefore, $1 + \frac{1}{\alpha(1-\delta)} \approx \frac{1}{\alpha(1-\delta)}$, so the number of iterations is approximately upper bounded by $\frac{\ln\frac{1}{\beta(1-\delta)}}{\ln 1+\frac{1}{d}}$. Furthermore, replacing β by d does not change the leading-order behavior. If instead, the dimension d is large and $1-\delta$ is small but fixed, so $\alpha(1-\delta) \gg 1$, then $\ln\Big(1 + \frac{1}{\alpha(1-\delta)}\Big) \approx \frac{1}{\alpha(1-\delta)}$ while $\ln\Big(1 + \frac{1}{d+1}\Big) \approx \frac{1}{d+1}$, and $\ln\frac{\alpha}{\beta} \approx \frac{2}{d}\ln\frac{1}{1-\delta}$. Equation (90) gives an upper bound for the number of iterations that is approximately $(d+1)\Big(\frac{1}{\alpha(1-\delta)} + \frac{2}{d}\ln\frac{1}{1-\delta}\Big) \approx \frac{1}{1-\delta} + 2\ln\frac{1}{1-\delta}$, which is similar to the bound in part 3(a).

3.3 Expected sample complexity

Since the SWAP gadget in Algorithm 1 is intrinsically probabilistic, the number of states it consumes in any particular run is a random variable. The same applies to our recursive purification procedure PURIFY(n) (Algorithm 2). Hence we are interested in the *expected sample complexity* of PURIFY(n), denoted by $\mathrm{C}(n,d)$, which we define as the expected number of states $\rho_0 = \rho(\delta)$ consumed during the algorithm, where the expectation is over the random measurement outcomes in all the swap tests used by the algorithm, and d is the dimension of $\rho(\delta)$. The sample complexity of PURIFY(n) is characterized by the following theorem.

Theorem 9. *Fix $d \geq 2$ and $\delta \in (0,1)$. Let $|\psi\rangle \in \mathbb{C}^d$ be an unknown state and define $\rho(\delta)$ as in eq. (1). For any $\epsilon > 0$, there exists a procedure that outputs $\rho(\delta')$ with $\delta' \leq \epsilon$ by consuming an expected number of $\mathrm{C}(n,d) < 4^{\min\left\{\frac{1}{1-\delta}+2\ln\left(\frac{1}{1-\delta}\right),(d+2)\ln\left(\frac{1}{1-\delta}\right)\right\}} \times \frac{3630}{\epsilon}$ copies of $\rho(\delta)$.*

Our strategy for proving Theorem 9 is as follows. In Lemma 10, we derive an expression for the expected sample complexity $\mathrm{C}(n,d)$ of $\textsc{Purify}(n)$ in terms of the recursion depth n and the product $\prod_{i=1}^{n} p_i^{(d)}$ of success probabilities $p_i^{(d)}$ of all steps. Then we prove the theorem by lower bounding the product $\prod_{i=1}^{n} p_i^{(d)}$ using eq. (17), Lemma 5, and upper bounds on the number of iterations n from Section 3.2.

Lemma 10. *The expected sample complexity of* $\textsc{Purify}(n)$ *is*

$$\mathrm{C}(n,d) = \frac{2^n}{\prod_{i=1}^{n} p_i^{(d)}}, \tag{91}$$

where the probabilities $p_i^{(d)}$ *are subject to the recurrence in eq. (16).*

Proof. For any $i \geq 1$, the expected number of copies of ρ_{i-1} consumed by the $\textsc{Swap}$ gadget within $\textsc{Purify}(i)$ is $2/p_i^{(d)}$, as in eq. (13). Since each ρ_{i-1} in turn requires an expected number of $2/p_{i-1}^{(d)}$ copies of ρ_{i-2}, the overall number of states ρ_0 consumed is equal to the product of $2/p_i^{(d)}$ across all n levels of the recursion. The expectation is multiplicative because the measurement outcomes at different levels are independent random variables. □

Proof of Theorem 9. To lower bound the denominator of $\mathrm{C}(n,d)$ in eq. (91), we first lower bound the values $p_i^{(d)}$ by bounding eq. (17) as

$$P(\delta,d) = 1 - \Big(1-\frac{1}{d}\Big)\delta + \frac{1}{2}\Big(1-\frac{1}{d}\Big)\delta^2 > \max\Big(1-\delta, \frac{1}{2}\Big(1+\frac{1}{d}\Big)\Big) > \max\Big(1-\delta, \frac{1}{2}\Big). \tag{92}$$

The first bound of $1-\delta$ is immediate, and the second bound is $P(1,d)$, which follows from the monotonicity of $P(\delta,d)$ established in Lemma 3.

We now combine Lemma 10 and eq. (92) to determine the number of iterations of Algorithm 2 to reach the desired target error parameter $\epsilon > 0$, starting from an arbitrary initial error parameter $\delta \in (0,1)$, for an arbitrary $d \geq 2$. Following the structure of Section 3.2, we divide the proof into three cases, depending on the initial error parameter δ.

Case 1: $0 < \delta < 1/2$. Using the final conclusion from part 1, the number of iterations n to achieve a final error parameter ϵ can be chosen to be no more than $\lceil \log_2 \frac{\delta}{(1-2\delta)\epsilon} \rceil \leq \log_2 \frac{\delta}{(1-2\delta)\epsilon} + 1$. We can thus upper bound the numerator of $\mathrm{C}(n,d)$ as follows:

$$2^n \leq 2^{1+\log_2 \frac{\delta}{(1-2\delta)\epsilon}} = 2 \cdot \frac{\delta}{\epsilon} \cdot \frac{1}{1-2\delta}. \tag{93}$$

Using eq. (25) in Lemma 5,

$$\prod_{i=1}^{n} p_i^{(d)} > \prod_{i=1}^{n} \Big(1-\delta_{i-1}^{(d)}\Big) \geq \prod_{i=1}^{n} \Big(1 - \frac{\delta}{2^{i-1}(1-2\delta)+2\delta}\Big) \tag{94}$$

$$= 1 - 2\delta(1-2^{-n}) \tag{95}$$

$$> 1 - 2\delta, \tag{96}$$

where eq. (95) can be proved by induction on n. Combining eqs. (91), (93), and (96),

$$\mathrm{C}(n,d) \leq \frac{2\delta}{\epsilon(1-2\delta)^2}. \tag{97}$$

Case 2: $1/3 \le \delta < 2/3$. From eq. (92) and the monotonicity of $P(\delta, d)$ in Lemma 3, $p_i^{(d)} > P(2/3, \infty) = 5/9$. Combining this with the factor of 2 in the numerator, each iteration uses at most 3.6 noisy states in expectation. At most 5 iterations are sufficient to reduce the error from $2/3$ to less than $1/3$ (using the $d = \infty$ case as an upper bound for all d), so $3.6^5 \le 605$ samples suffice. Applying the analysis of the previous case for $\delta = 1/3$, we see that $6/\epsilon$ samples reduce the error from $1/3$ to at most ϵ. Combining the bounds from these two stages, we have

$$\mathrm{C}(n, d) \le \frac{3630}{\epsilon}. \tag{98}$$

Case 3: $2/3 \le \delta < 1$. Our strategy is to first drive the error parameter below $2/3$ and then apply the bound from case 2. Recall from the previous subsection that this takes $i^* + 1$ iterations, but one iteration is already counted in case 2. Since $p_i^{(d)} > \frac{1}{2}\left(1 + \frac{1}{d}\right)$ from eq. (92), $\mathrm{C}(n, i^*) \le \left(\frac{4}{1+\frac{1}{d}}\right)^{i^*}$. For large enough d, the simpler complexity bound of 4^{i^*} suffices. Parts 3(a) and 3(b) each provide a complexity bound $\mathrm{C}(i^*, d)$ for reducing the error from δ to $2/3$, and we can take the minimum of the two. From Lemma 7 in part 3(a),

$$\mathrm{C}(i^*, d) < 4^{i^*} \le 4^{\frac{1}{1-\delta} + 2\ln\left(\frac{1}{1-\delta}\right)}. \tag{99}$$

For small dimension $d \ge 3$, Part 3(b) provides a sharper bound of $d(1-\delta) \ll 1$. In this case, we can use eq. (89) to obtain

$$\mathrm{C}(i^*, d) < 4^{i^*} = (a^{-1})^{\frac{\ln 4}{\ln \frac{d+2}{d+1}} i^*} < \left(\frac{\frac{1}{1-\delta} + \alpha}{\beta}\right)^{\frac{\ln 4}{\ln \frac{d+2}{d+1}}} < 4^{\frac{\ln\left(\frac{1}{1-\delta} + \frac{\alpha}{\beta}\right)}{\ln \frac{d+2}{d+1}}}, \tag{100}$$

where α, β are as defined in Part 3(b). Combining the above bounds and that from the previous case, we upper bound the sample complexity to reduce the noise parameter from $\delta > 2/3$ to ϵ:

$$\mathrm{C}(n, d) < 4^{\min\left\{ \frac{1}{1-\delta} + 2\ln\left(\frac{1}{1-\delta}\right), \frac{\ln\left(\frac{\frac{1}{1-\delta} + \alpha}{\beta}\right)}{\ln \frac{d+2}{d+1}} \right\}} \times \frac{3630}{\epsilon}. \tag{101}$$

Given d, δ, the above provides a tight bound. However, we can also simplify the second bound by observing

$$\ln\left(\frac{\frac{1}{1-\delta} + \alpha}{\beta}\right) = \ln\left(\frac{1}{1-\delta}\right) + \ln\left(\frac{\alpha(1-\delta)}{\beta} + \frac{1}{\beta}\right). \tag{102}$$

For large d, $\alpha \approx \beta \approx d$, so $\frac{\alpha(1-\delta)}{\beta} + \frac{1}{\beta} \to 1 - \delta < 1/3$. In fact, for all $d \ge 2$,

$$\frac{\alpha(1-\delta)}{\beta} + \frac{1}{\beta} < \alpha/3 + \frac{1}{\beta} \tag{103}$$

$$< \frac{\left(\beta + 2\,c\,a\ \ln(\min\{d, n^{(\infty)*}\}) - 3 + 7.2\,c\,a\right)/3 + 1}{\beta} \tag{104}$$

$$< \frac{1}{3} + \frac{2\,c\,a\ \ln d + 7.2\,c\,a}{3\beta}. \tag{105}$$

As d increases, $ca \to 1$ and β increases roughly as d, and the second term above initially increases to attain a maximum of 0.5512 at $d = 7$ and decreases thereafter, so the last line is upper bounded by 0.8845. Applying this bound to eq. (102) gives

$$\ln\left(\frac{\frac{1}{1-\delta}+\alpha}{\beta}\right) < \ln\left(\frac{1}{1-\delta}\right). \tag{106}$$

Using $\ln(1+x) \geq \frac{x}{x+1}$ to get $\ln\frac{d+2}{d+1} \geq \frac{1}{d+2}$, we have

$$\mathrm{C}(n,d) < 4^{\min\left\{\frac{1}{1-\delta}+2\ln\left(\frac{1}{1-\delta}\right),(d+2)\ln\left(\frac{1}{1-\delta}\right)\right\}} \times \frac{3630}{\epsilon} \tag{107}$$

as claimed. □

As a side note, if $\alpha(1-\delta) \ll 1$, we can obtain a tighter approximation (though not a strict bound) for the sample complexity bound by revising eq. (102) as

$$\ln\left(\frac{\frac{1}{1-\delta}+\alpha}{\beta}\right) = \ln\left(\frac{1}{1-\delta}\right) + \ln\left(\frac{\alpha(1-\delta)}{\beta}+\frac{1}{\beta}\right) \approx \ln\left(\frac{1}{1-\delta}\right) + \ln\left(\frac{1}{\beta}\right) = \ln\left(\frac{1}{(1-\delta)\beta}\right),$$

so

$$\mathrm{C}(n,d) \lesssim 4^{\frac{\ln\frac{1}{(1-\delta)\beta}}{\ln\frac{d+2}{d+1}}} \times \frac{3630}{\epsilon}, \tag{108}$$

which recovers a polynomial dependence on $\frac{1}{1-\delta}$.

We conclude this section by describing the gate complexity of Purify. Since the number of internal nodes in a binary tree is upper bounded by the number of leaves, the total number of Swap gadgets is upper bounded by twice the sample complexity. As mentioned in Section 2.1, the swap test for qudits can be performed using $O(\log d)$ two-qubit gates. Therefore, from eq. (99) we have the following upper bound on the gate complexity of purification for constant δ.

Corollary 11. *For any dimension $d \geq 2$ and any fixed initial error parameter $\delta \in (0,1)$, the expected gate complexity of Purify to achieve an error parameter ϵ is $O(\frac{1}{\epsilon}\log d)$.*

4 Applications

4.1 Simon's problem with a faulty oracle

In this subsection we describe an application of quantum state purification to a query complexity problem with an inherently faulty oracle. Query complexity provides a model of computation in which a black box (or oracle) must be queried to learn information about some input, and the goal is to compute some function of that input using as few queries as possible. It is well known that quantum computers can solve certain problems using dramatically fewer queries than any classical algorithm.

Regev and Schiff [RS08] studied the effect on query complexity of providing an imperfect oracle. They considered the unstructured search problem, where the goal is to determine whether some black-box function $f\colon [M] \to \{0,1\}$ has any input $x \in [M]$ for which $f(x) = 1$. This problem can be solved in $O(\sqrt{M})$ quantum queries [Gro96], whereas a classical computer needs $\Omega(M)$ queries. Regev and Schiff showed that if the black box for f fails to act with some constant probability, the quantum speedup disappears: then

a quantum computer also requires $\Omega(M)$ queries. Given this result, it is natural to ask whether a significant quantum speedup is ever possible with a faulty oracle.

Here we consider the quantum query complexity of Simon's problem [Sim97]. In this problem we are given a black-box function $f\colon \{0,1\}^m \to \{0,1\}^m$ with the promise that there exists a hidden string $s \in \{0,1\}^m \setminus \{0^m\}$ such that $f(x) = f(y)$ if and only if $x = y$ or $x \oplus y = s$. The goal is to find s. With an ideal black box, this problem can be solved with $O(m)$ quantum queries (and poly(m) additional quantum gates), whereas it requires $2^{\Omega(m)}$ classical queries (even for a randomized algorithm).

Now we consider a faulty oracle. If we use the same model as in [RS08], then it is straightforward to see that the quantum query complexity remains $O(m)$. Simon's algorithm employs a subroutine that outputs uniformly random strings y with the property that $y \cdot s = 0$. By sampling $O(m)$ such values y, one can determine s with high probability. If the oracle fails to act, then Simon's algorithm is instead guaranteed to output $y = 0$. While this is uninformative, it is consistent with the condition $y \cdot s = 0$. Thus it suffices to use the same reconstruction procedure to determine s, except that one must take more samples to get sufficiently many nonzero values y. But if the probability of the oracle failing to act is a constant, then the extra query overhead is only constant, and the query complexity remains $O(m)$.

Of course, this analysis is specific to the faulty oracle model where the oracle fails to act with some probability. We now consider instead another natural model of a faulty oracle where, with some fixed probability $\delta \in (0,1)$, the oracle depolarizes its input state. In other words, the oracle acts on its input density matrix ρ according to the quantum channel D_δ where

$$D_\delta(\rho) = (1-\delta) U_f \rho U_f^\dagger + \delta \frac{I}{2^{2m}} \tag{109}$$

where U_f is the unitary operation that reversibly computes f (concretely, $U_f|x,a\rangle = |x, a \oplus f(x)\rangle$). In this case, if we simply apply the main subroutine of Simon's algorithm using the faulty oracle, we obtain a state

$$\rho' = (1-\delta)|\Psi\rangle\langle\Psi| + \delta \frac{I}{2^{2m}} \tag{110}$$

before measuring the first register, where

$$|\Psi\rangle = \frac{1}{2^m} \sum_{x \in \{0,1\}^m} \sum_{\substack{y \in \{0,1\}^m \\ \text{s.t. } y \cdot s = 0}} (-1)^{x \cdot y} |y\rangle |f(x)\rangle \tag{111}$$

is the ideal output.

If we simply measure the first register of ρ', then with probability $1-\delta$ we obtain y satisfying $y \cdot s = 0$, but with probability δ, we obtain a uniformly random string y. Determining s from such samples is an apparently difficult "learning with errors" problem. While s is information-theoretically determined from only $O(m)$ samples—showing that $O(m)$ quantum queries suffice—the best known (classical or quantum) algorithm for performing this reconstruction takes exponential time [Reg10].

We can circumvent this difficulty using quantum state purification. Suppose we apply Purify to copies of ρ', each of which can be produced with one (faulty) query. Since Simon's algorithm uses $O(m)$ queries in the noiseless case, purifying the state to one with $\epsilon = O(1/m)$ ensures that the overall error of Simon's algorithm with a faulty oracle is at most constant. By Theorem 9, the expected number of copies of ρ' consumed by Purify

is $O(\frac{1}{\epsilon}) = O(m)$. We repeat this process $O(m)$ times to obtain sufficiently many samples to determine s. Overall, this gives an algorithm with query complexity $O(m^2)$. Since the purification can be performed efficiently by Corollary 11, this algorithm uses only poly(m) gates.

Note that the same approach can be applied to any quantum query algorithm that applies classical processing to a quantum subroutine that makes a constant number of queries to a depolarized oracle. However, this approach is not applicable to a problem such as unstructured search for which quantum speedup requires high query depth [Zal99].

4.2 Relationship to mixedness testing and quantum state tomography

In the problem of *mixedness testing* [MW16], copies of an unknown d-dimensional quantum state ρ are available, where ρ is promised to be either the maximally mixed state $\frac{I}{d}$ or at least η-far from $\frac{I}{d}$ in trace distance. References [OW15b, Wri16] establish a sample complexity of $\Theta(d/\eta^2)$ copies of ρ for this problem. In particular, the lower bound is proved for a state ρ with eigenvalues $\frac{1}{d}(1 \pm \frac{\eta}{2})$, each with degeneracy $\approx \frac{d}{2}$.

If the state ρ is restricted to the form $\rho(\delta)$ given by eq. (1), where $\delta = 1$ or $\delta \leq 1-\frac{\eta}{2}$, one can solve the mixedness testing problem simply by applying Algorithm 2 as an attempt to purify the samples. If $\delta = 1$, the swap test in each iteration fails with probability $1/2$ and always outputs I/d. If instead, $\delta \leq 1 - \frac{\eta}{2}$, then after $15 + \frac{1}{1-\delta} + 2\ln\left(\frac{1}{1-\delta}\right) \leq 15 + \frac{2}{\eta} + 2\ln\frac{2}{\eta}$ iterations, the algorithm outputs $\rho(\epsilon)$ with $\epsilon \leq 2^{-10}$, and the swap test in the final iteration passes with near certainty. The difference in the probability of passing the swap test in the last iteration can be used to determine whether $\delta = 1$ or $\delta \leq 1 - \frac{\eta}{2}$ with only a few repetitions. The sample complexity found in Section 3.3 has worse dependence on η (compared to $\Theta(d/\eta^2)$ from [OW15b, Wri16]), but it is independent of the dimension d. Therefore, under the promise that the unknown ρ has the form of a depolarized pure state as in eq. (1), the protocol in [OW15b, Wri16] is not optimal in terms of the dimension d.

In the problem of quantum state tomography, copies of an unknown d-dimensional quantum state ρ are available, and a desired accuracy level $\eta > 0$ is specified. The output is a classical description of a d-dimensional quantum state σ such that $\|\rho - \sigma\|_1 \leq \eta$. For this application, we adopt the trace distance as a measure of accuracy which better facilitates the discussion.

One method to perform quantum state purification of an unknown state ρ of the form $\rho(\delta) = (1-\delta)|\psi\rangle\langle\psi| + \delta\,\frac{I}{d}$ given by eq. (1) is to first perform quantum state tomography, obtain an estimate σ of $\rho(\delta)$, and then prepare a copy of the principal eigenvector of σ. Let $|\phi\rangle$ be the principal eigenvector of σ. We now determine how many samples suffice to achieve $\||\psi\rangle\langle\psi| - |\phi\rangle\langle\phi|\|_1 \leq \epsilon$.

Since $|\psi\rangle\langle\psi|$ is scaled by a factor of $1-\delta$ in $\rho(\delta)$, we expect the required accuracy for tomography η to scale linearly in $1-\delta$. Therefore, we let $\eta = (1-\delta)\xi$ and determine ξ as a function of the allowed noise parameter ϵ for the purification. Thus ξ relates the accuracy of tomography to that of purification.

The accuracy of this approach to purification can be understood as follows. If $\|\rho(\delta) - \sigma\|_1 \leq \eta$ and λ is the principal eigenvalue of σ, then

$$\lambda \geq (1-\delta) + \delta/d - (1-\delta)\xi = 1/d + (1-\delta)(1-\xi-1/d). \tag{112}$$

Furthermore, since $|\phi\rangle$ is the principal eigenvector of σ,

$$(1-\delta)\xi \geq \operatorname{Tr}|\phi\rangle\langle\phi|(\sigma - \rho(\delta)) = |\lambda - (1-\delta)|\langle\phi|\psi\rangle|^2 - \delta/d|, \tag{113}$$

where we have used $\|X\|_1 = \max\{\mathrm{Tr}(UX) : U \text{ unitary}\}$ for a square matrix X to obtain the leftmost inequality. Combining the two inequalities above, $|\langle\phi|\psi\rangle|^2 \geq 1 - 2\xi$. The trace distance between $|\psi\rangle$ and $|\phi\rangle$ is $\sqrt{1 - |\langle\phi|\psi\rangle|^2} \leq \sqrt{2\xi} = \epsilon$ if we choose $\xi = \epsilon^2/2$.

Having related ξ to ϵ, it remains to determine the sample complexity for the tomography step. The sample complexity is $O(d^2/\eta^2) = O(d^2(1-\delta)^{-2}\epsilon^{-4})$ if collective measurements are allowed, and $O(d^3/\eta^2) = O(d^3(1-\delta)^{-2}\epsilon^{-4})$ if one is restricted to single-copy (streaming) measurements (see for example [OW15a, Wri16, HHJ+17]). We remark that for $d = 2^n$, these measurements can be chosen to be single-qubit Pauli measurements, which can be efficiently performed. Additional structure of the states $\rho(\delta)$ given by eq. (1), such as low rank of the pure-state component and tracelessness of the noisy component, likely reduce the complexity further. We leave a detailed analysis to future research.

Comparing the sample complexity for purification using Algorithm 2 (see Section 3.3) with that using state tomography, when d is small and $1-\delta$ is vanishing, state tomography has better dependence on $1-\delta$ (even with single-copy measurements). One disadvantage of Algorithm 2 is that many copies of $\rho(\delta)$ are discarded in the process when the swap test fails, until δ drops below $1/2$. This is a major source of inefficiency as the postmeasurement quantum states from failed swap tests still contain significant information about $\rho(\delta)$. In contrast, state tomography does not waste any samples, and the amplification of the small signal has only inverse square dependence on $1-\delta$. However, for any constant δ, or for growing dimension d and small but non-vanishing $1-\delta$, Algorithm 2 has dimension-independent complexity, unlike state tomography (which involves learning the state and necessarily has complexity growing with dimension). One additional disadvantage of tomography is that it can output a target state that has high quantum circuit complexity to prepare.

4.3 Relationship to quantum majority vote problem

Purification is closely related to the following *quantum majority vote* problem [BLM+22]: given an N-qubit state $U^{\otimes N}|x\rangle$ for an unknown unitary $U \in \mathrm{U}(2)$ and bit string $x \in \{0,1\}^N$, output an approximation of $U|\mathrm{MAJ}(x)\rangle$, where $\mathrm{MAJ}(x) \in \{0,1\}$ denotes the majority of x. While the input to this problem is not of the form $\rho^{\otimes N}$, applying a random permutation produces a separable state whose single-qubit marginals are identical and equal to a state ρ as in eq. (1). The optimal quantum majority vote procedure for qubits [BLM+22] appears to coincide with the optimal qubit purification procedure [CEM99]. However, its generalization to qudits has been investigated only in very limited cases [GO22]. Our qudit purification problem can be interpreted as a generalization of quantum majority voting to higher dimensions.

5 Sample complexity lower bound

In this section we prove a lower bound on the cost of purifying a d-dimensional state. In particular, this bound shows that for constant d, the ϵ-dependence of our protocol is optimal.

Lemma 12. *For any constant input error parameter $\delta > 0$, the sample complexity of purifying a d-dimensional state $\rho(\delta)$ with output fidelity $1-\epsilon$ is $\Omega(\frac{1}{d\epsilon})$.*

Proof. We use the fact that a purification procedure for d-dimensional states can also be used to purify 2-dimensional states, a task with known optimality bounds. For clarity, we

write $|1\rangle$ as the target qubit state to be purified, and let $|2\rangle$ denote a state orthonormal to $|1\rangle$. This is simply a choice of notation and entails no loss of generality, as the operations described in the proof do not depend on the states $|1\rangle$ or $|2\rangle$.

We consider two procedures for purifying N samples of the qubit state $\rho^{(2)}(\delta^{(2)}) = (1 - \frac{\delta^{(2)}}{2})|1\rangle\langle 1| + \frac{\delta^{(2)}}{2}|2\rangle\langle 2|$:

1. applying the optimal qubit procedure, denoted by $\mathcal{P}^{(2)}_{\text{opt}}$, on the N samples; or
2. embedding each copy of $\rho^{(2)}(\delta^{(2)})$ in the d-dimensional space and then applying the optimal qudit purification procedure, denoted by $\mathcal{P}^{(d)}_{\text{opt}}$.

We denote the second procedure by $\mathcal{P}^{(2)}_{\text{embed}}$. Clearly it cannot outperform the optimal procedure $\mathcal{P}^{(2)}_{\text{opt}}$.

The embedding operation involves mixing the given state $\rho^{(2)}(\delta^{(2)})$ with the maximally mixed state $\frac{\Lambda}{d-2}$ in the space orthogonal to the support of $\rho^{(2)}(\delta^{(2)})$. The embedded d-dimensional state is

$$\rho^{(d)}(\delta^{(d)}) = (1 - q)\rho^{(2)}(\delta^{(2)}) + \frac{q}{d-2}\Lambda \tag{114}$$

where $\Lambda := \sum_{i=3}^{d} |i\rangle\langle i|$. We want $\rho^{(d)}(\delta^{(d)})$ to be of the form

$$\rho^{(d)}(\delta^{(d)}) = \left(1 - \frac{d-1}{d}\delta^{(d)}\right)|1\rangle\langle 1| + \frac{\delta^{(d)}}{d}\sum_{i=2}^{d}|i\rangle\langle i|. \tag{115}$$

For eq. (114) and eq. (115) to describe the same state, they need to have the same coefficient for each diagonal entry. This requirement leads to three linear equations for $\delta^{(d)}$ and q, in terms of $\delta^{(2)}$. But these three equations are linearly dependent, so it suffices to express them with the following two equations:

$$q = \frac{d-2}{d}\delta^{(d)}, \qquad \delta^{(2)} = \frac{2\delta^{(d)}}{d - (d-2)\delta^{(d)}}, \tag{116}$$

where the parameters q and $\delta^{(2)}$ are expressed in terms of $\delta^{(d)}$ and d. We note that for $\delta^{(2)} \in (0, 1)$, the corresponding q and $\delta^{(d)}$ are also in $(0, 1)$.

Let the final output fidelity be F_{embed} and F_{opt} for $\mathcal{P}^{(2)}_{\text{embed}}$ and $\mathcal{P}^{(2)}_{\text{opt}}$, respectively. Since $\mathcal{P}^{(2)}_{\text{embed}}$ cannot outperform $\mathcal{P}^{(2)}_{\text{opt}}$, we know that $F_{\text{embed}} \leq F_{\text{opt}}$. From the qubit case [CEM99], the final fidelity satisfies $F_{\text{opt}} \leq 1 - \frac{\delta^{(2)}}{2(1-\delta^{(2)})^2}\frac{1}{N}$, so if $F_{\text{embed}} = 1 - \epsilon$, then $F_{\text{opt}} \geq 1 - \epsilon$ and we have

$$N \geq \frac{\delta^{(d)}(d - (d-2)\delta^{(d)})}{d^2(1-\delta^{(d)})^2}\frac{1}{\epsilon} = \frac{\delta^{(d)}}{(1-\delta^{(d)})d\epsilon} + \Theta\left(\frac{1}{d^2\epsilon}\right). \tag{117}$$

This shows that $N \in \Omega(\frac{1}{d\epsilon})$. □

For constant d, the lower bound is $\Omega(\frac{1}{\epsilon})$, which matches our upper bound on the sample complexity of Purify, showing that our approach is asymptotically optimal. As discussed further in the next section, the results of [LFIC24] indicate that the asymptotic scaling with d in our lower bound is not optimal. Moreover, up to small multiplicative constants, the Purify procedure is asymptotically optimal in d and ϵ for $\delta < 1/3$. For $\delta \gg 2/3$, our analysis of the number of iterations i^* to suppress the error down to $2/3$ suggests suboptimality of the dependence on $1-\delta$ relative to the $\Omega((1-\delta)^{-2})$ performance of the optimal protocol.

6 Conclusions

We have established a general purification procedure based on the swap test. This procedure is efficient, with sample complexity $O(\frac{1}{\epsilon})$ for any fixed initial error δ, independent of the dimension d. The simplicity of the procedure makes it easy to implement, and in particular, it requires a quantum memory of size only logarithmic in the number of input states consumed. Note however that it requires a quantum memory with long coherence time since partially purified states must be maintained when the procedure needs to restart.

It is natural to ask whether the purification protocol can be asymptotically improved when the dimension is not fixed. One way of approaching this question, which is also of independent interest, is to characterize the optimal purification protocol. In a related line of work, partially reported in [Fu16], some of the authors developed a framework to study the optimal qudit purification procedure using Schur–Weyl duality. The optimal procedure can be characterized in terms of a sequence of linear programs involving the representation theory of the unitary group. Using the Clebsch–Gordan transform and Gel'fand–Tsetlin patterns, we gain insight into the common structure of all valid purification operations and give a concrete formula for the linear program behind the optimization problem. For qubits, this program can be solved to reproduce the optimal output fidelity

$$F_2 = 1 - \frac{1}{2(N+1)} \cdot \frac{\delta}{(1-\delta)^2} + O\Big(\frac{1}{N^2}\Big) \tag{118}$$

as established in Ref. [CEM99]. For qutrits, we show that the optimal output fidelity is

$$F_3 = 1 - \frac{2}{3(N+1)} \cdot \frac{\delta}{(1-\delta)^2} + O\Big(\frac{1}{N^2}\Big). \tag{119}$$

More generally, our formulation makes it easy to numerically compute the optimal fidelity and determine the optimal purification procedure for any given dimension. Due to the complexity of the Gel'fand–Tsetlin patterns, a general analytical expression for the optimal solution of the linear program was elusive, but this problem was recently solved [LFIC24]. It turns out that the optimal qudit fidelity is

$$F_d = 1 - \frac{d-1}{d(N+1)} \cdot \frac{\delta}{(1-\delta)^2} + O\Big(\frac{1}{N^2}\Big), \tag{120}$$

so for $F_d \geq 1 - \epsilon$, the sample complexity is

$$N \leq \frac{d-1}{d} \cdot \frac{\delta}{\epsilon(1-\delta)^2}. \tag{121}$$

Comparing to the sample complexity in eq. (97), we see that our expected sample complexity is optimal in d and ϵ up to a small multiplicative constant if $\delta < 1/3$.

Simon's problem with a faulty oracle, as described in Section 4.1, is just one application of Purify. This approach can also be applied to quantum algorithms involving parallel or sequential queries of depth $O(1)$ subject to a depolarizing oracle. However, for a quantum algorithm that makes $\omega(1)$ sequential queries, the query overhead of our purification protocol is significant. In future work, it might be interesting to explore query complexity with faulty oracles more generally, or to develop other applications of Purify to quantum information processing tasks.

The swap test, which is a key ingredient in our algorithm, is a special case of generalized phase estimation [Har05, Section 8.1]. Specifically, the swap test corresponds to generalized phase estimation where the underlying group is S_2. This places our algorithm in a wider context and suggests possible generalizations using larger symmetric groups S_k for $k \geq 2$. Such generalizations might provide a tradeoff between sample complexity and memory usage, since generalized phase estimation for S_k operates simultaneously on k samples. It would be interesting to understand the nature of such tradeoffs for purification or for other tasks. In particular, can one approximately implement the Schur transform using less quantum memory?

Our procedure has a pure target state $|\psi\rangle\langle\psi|$ which is also its fixed point. The same procedure cannot be used for mixed target states since its output might be "too pure." Nevertheless, one can still ask how to implement the transformation

$$\left((1-\delta)\rho + \delta\frac{I}{d}\right)^{\otimes N} \mapsto (1-\epsilon)\rho + \epsilon\frac{I}{d} \tag{122}$$

for $\epsilon \ll \delta$ and for sufficiently large N. This purification can be performed using quantum state tomography. For pure states in various regimes, our method is much more efficient than this. Is there a more sample-efficient method for mixed states? Generalization of the swap test to techniques such as generalized phase estimation may provide a solution, but we leave this question for future research.

Finally, a natural generalization of the current work is the purification of a general unknown mixed state ρ with the promise that the principal eigenvalue is non-degenerate. Applications have been found in [GPS24], and ongoing research is addressing this open problem.

Acknowledgements

We thank Daniel Grier, David Gross, Barbara Kraus, Meenu Kumari, Aadil Oufkir, Hakop Pashayan, Luke Schaeffer, and Nengkun Yu for very helpful discussions.

AMC, HF, DL, and VV received support from NSERC during part of this project. AMC also received support from NSF (grants 1526380, 1813814, and QLCI grant OMA-2120757). HF acknowledges the support of the NSF QLCI program (grant OMA-2016245). DL and ZL, via the Perimeter Institute, are supported in part by the Government of Canada through the Department of Innovation, Science and Economic Development and by the Province of Ontario through the Ministry of Colleges and Universities. MO was supported by an NWO Vidi grant (Project No. VI.Vidi.192.109).

References

[BBP+96] Charles H. Bennett, Gilles Brassard, Sandu Popescu, Benjamin Schumacher, John A. Smolin, and William K. Wootters. Purification of noisy entanglement and faithful teleportation via noisy channels. *Physical Review Letters*, 76(5):722, 1996. arXiv:quant-ph/9511027. doi:10.1103/PhysRevLett.76.722.

[BCH06] Dave Bacon, Isaac L. Chuang, and Aram W. Harrow. Efficient quantum circuits for Schur and Clebsch-Gordan transforms. *Phys. Rev. Lett.*, 97(17):170502, Oct 2006. arXiv:quant-ph/0407082. doi:10.1103/PhysRevLett.97.170502.

[BCWDW01] Harry Buhrman, Richard Cleve, John Watrous, and Ronald De Wolf. Quantum fingerprinting. *Physical Review Letters*, 87(16):167902, 2001. arXiv:quant-ph/0102001. doi:10.1103/PhysRevLett.87.167902.

[BLM+22] Harry Buhrman, Noah Linden, Laura Mančinska, Ashley Montanaro, and Maris Ozols. Quantum majority vote, 2022. arXiv:2211.11729. doi:10.48550/arXiv.2211.11729.

[CEM99] J.I. Cirac, A.K. Ekert, and C. Macchiavello. Optimal purification of single qubits. *Physical Review Letters*, 82(21):4344, 1999. arXiv:quant-ph/9812075. doi:10.1103/PhysRevLett.82.4344.

[Fiu04] Jaromír Fiurášek. Optimal probabilistic cloning and purification of quantum states. *Phys. Rev. A*, 70:032308, Sep 2004. arXiv:quant-ph/0403165. doi:10.1103/PhysRevA.70.032308.

[Fu16] Honghao Fu. Quantum state purification. Master's thesis, University of Waterloo, 2016. URL: http://hdl.handle.net/10012/10706.

[GO22] Dmitry Grinko and Maris Ozols. Linear programming with unitary-equivariant constraints, 2022. arXiv:2207.05713. doi:10.48550/arXiv.2207.05713.

[GPS24] Daniel Grier, Hakop Pashayan, and Luke Schaeffer. Principal eigenstate classical shadows, 2024. arXiv:2405.13939. doi:10.48550/arXiv.2405.13939.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 212–219. ACM, 1996. arXiv:quant-ph/9605043. doi/10.1145/237814.237866.

[Har05] Aram W. Harrow. *Applications of coherent classical communication and the Schur transform to quantum information theory*. PhD thesis, MIT, 2005. arXiv:quant-ph/0512255. doi/10.48550/arXiv.quant-ph/0512255.

[HHJ+17] Jeongwan Haah, Aram W. Harrow, Zhengfeng Ji, Xiaodi Wu, and Nengkun Yu. Sample-optimal tomography of quantum states. *IEEE Transactions on Information Theory*, 63(9):5628–5641, 2017. arXiv:1508.01797. doi:10.1109/TIT.2017.2719044.

[KW01] M. Keyl and R.F. Werner. The rate of optimal purification procedures. *Annales Henri Poincare*, 2(1):1–26, 2001. arXiv:quant-ph/9910124. doi/10.1007/PL00001027.

[LFIC24] Zhaoyi Li, Honghao Fu, Takuya Isogawa, and Isaac Chuang. Optimal quantum purity amplification. arXiv:2409.18167. doi/10.48550/arXiv.2409.18167.

[LLY+24] Qing Liu, Zihao Li, Xiao Yuan, Huangjun Zhu, and You Zhou. Auxiliary-free replica shadow estimation. arXiv:2407.20865. doi/10.48550/arXiv.2407.20865.

[MW16] Ashley Montanaro and Ronald de Wolf. *A Survey of Quantum Property Testing*. Number 7 in Graduate Surveys. Theory of Computing Library, 2016. arXiv:1310.2035, doi:10.4086/toc.gs.2016.007.

[OW15a] Ryan O'Donnell and John Wright. Efficient quantum tomography, In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 899-912, New York, NY, USA, 2016. Association for Computing Machinery. arXiv:1508.01907. doi:10.1145/2897518.2897544.

[OW15b] Ryan O'Donnell and John Wright. Quantum spectrum testing. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, page 529–538, New York, NY, USA, 2015. Association for Computing Machinery. arXiv:1501.05028, doi:10.1145/2746539.2746582.

[Reg10] Oded Regev. The learning with errors problem (Invited Survey). *2010 IEEE 25th Annual Conference on Computational Complexity, Cambridge, MA, USA, 2010, pp. 191-204.* doi:10.1109/CCC.2010.26.

[RS08] Oded Regev and Liron Schiff. Impossibility of a quantum speed-up with a faulty oracle. In *Automata, Languages and Programming, ICALP 2008, LNCS, 5125, Springer, Berlin, Heidelberg.*, arXiv:1202.1027. doi:10.1007/978-3-540-70575-8_63.

[Sim97] Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997. doi:10.1137/S0097539796298637.

[Ste96] Stevo Stević. Asymptotic behaviour of a sequence defined by iteration. *Matematički Vesnik*, 48(3-4):99–105, 1996. Asymptotic behavior of a sequence defined by iteration with applications. Colloquium Mathematicum 93 (2002), 267-276. doi:10.4064/cm93-2-6.

[Wri16] John Wright. *How to learn a quantum state.* PhD thesis, Carnegie Mellon University, 2016. URL: https://csd.cmu.edu/academics/doctoral/degrees-conferred/john-wright.

[YCH+24] Hongshun Yao, Yu-Ao Chen, Erdong Huang, Kaichu Chen, Honghao Fu, and Xin Wang. Protocols and trade-offs of quantum state purification, 2024. arXiv:2404.01138. doi/10.48550/arXiv.2404.01138.

[YKLO24] Bo Yang, Elham Kashefi, Dominik Leichtle, and Harold Ollivier. Quantum error suppression with subgroup stabilisation projectors. arXiv:2404.09973. doi/10.48550/arXiv.2404.09973.

[Zal99] Christof Zalka. Grover's quantum searching algorithm is optimal. *Physical Review A*, 60:2746–2751, 1999. arXiv:quant-ph/9711070. doi/10.1103/PhysRevA.60.2746.

[ZL22] You Zhou and Zhenhuan Liu. A hybrid framework for estimating nonlinear functions of quantum states, 2022. arXiv:2208.08416. doi/10.48550/arXiv.2208.08416.

A Output error parameter from swap test as a function of input errors

Here, we analyse the output error parameter when the input states for the swap test gadget (Algorithm 1) are not necessarily the same. For the purpose of purification, we are particularly interested in the special case when both input states, ρ and σ, are of the form

$$\rho(\delta) := (1-\delta)|\psi\rangle\langle\psi| + \delta\,\frac{I}{d} \tag{123}$$

for some *error parameter* $\delta \in (0,1)$, where $|\psi\rangle \in \mathbb{C}^d$ is an unknown but fixed d-dimensional state (the dimension $d \geq 2$ is often implicit). The following claim shows that the SWAP gadget preserves the form of these states, irrespective of the actual state $|\psi\rangle$, and derives the error parameter of the output state as a function of both input error parameters.

Proposition 13. *If the input states of Algorithm 1 are $\rho(\delta_1)$ and $\rho(\delta_2)$ as described in eq. (1), then*

$$\textsc{Swap}(\rho(\delta_1),\rho(\delta_2)) = \rho(\delta') \qquad \textit{where} \qquad \delta' := \frac{\frac{\delta_1+\delta_2}{2} + \frac{\delta_1\delta_2}{d}}{(1+\frac{1}{d}) + (1-\frac{1}{d})(1-\delta_1)(1-\delta_2)}. \tag{124}$$

Proof. Equation (1) shows that both input states are linear combinations of projectors $\{|\psi\rangle\langle\psi|, I\}$. Since this set is closed under matrix multiplication, the output state in eq. (7) is also a linear combination of $|\psi\rangle\langle\psi|$ and I. By construction, the output is positive semidefinite and has unit trace, so it must be of the form $\rho(\delta')$ for some $\delta' \in [0,1]$.

Note that $\rho(\delta_1)$ and $\rho(\delta_2)$ are diagonal in the same basis, so $\rho(\delta_1)\rho(\delta_2) = \rho(\delta_2)\rho(\delta_1)$ and hence the numerator of eq. (7) is $\rho(\delta_1) + \rho(\delta_2) + 2\rho(\delta_1)\rho(\delta_2)$. Using eq. (1), the coefficient of I in this expression is

$$\alpha := \frac{\delta_1}{d} + \frac{\delta_2}{d} + 2\frac{\delta_1\delta_2}{d^2}. \tag{125}$$

The denominator of eq. (7) is given by

$$\begin{aligned}
\beta &:= 1 + \operatorname{Tr}\big(\rho(\delta_1)\rho(\delta_2)\big) \\
&= 1 + (1-\delta_1)(1-\delta_2) + (1-\delta_1)\frac{\delta_2}{d} + \frac{\delta_1}{d}(1-\delta_2) + \frac{\delta_1}{d}\frac{\delta_2}{d}d \\
&= 1 + \Big(1-\frac{1}{d}\Big)(1-\delta_1)(1-\delta_2) + \frac{1}{d}\big((1-\delta_1)+\delta_1\big)\big((1-\delta_2)+\delta_2\big) \\
&= \Big(1+\frac{1}{d}\Big) + \Big(1-\frac{1}{d}\Big)(1-\delta_1)(1-\delta_2).
\end{aligned} \tag{126}$$

The coefficient of I/d for the output state in eq. (7) is thus $\frac{d}{2}\cdot\frac{\alpha}{\beta}$, which agrees with δ' in eq. (124). □

Proposition 13 tells us how the SWAP gadget affects the error parameter δ. In particular, the output state is purer than both input states (i.e., $\delta < \min\{\delta_1,\delta_2\}$) only when δ_1 and δ_2 are sufficiently close. The region of values (δ_1,δ_2) for which this holds is illustrated in Fig. 5 for several different d.

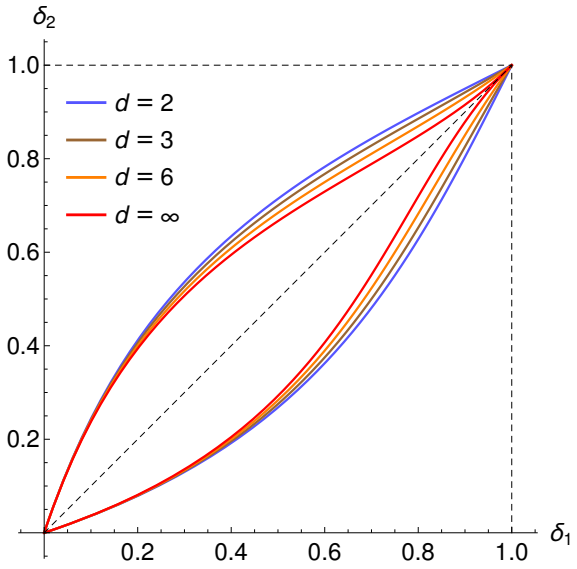


Figure 5: The outer boundary of the region in the (δ_1, δ_2) plane where the SWAP gadget (Algorithm 1) improves the purity of the output compared to both inputs (see Proposition 14), for dimensions $d \in \{2, 3, 6, \infty\}$. As d increases, the region shrinks and converges to the area enclosed by the red curve as $d \to \infty$. For any d, the region contains a substantial area surrounding the $\delta_1 = \delta_2$ line, except for the tip at $\delta_1 = \delta_2 = 1$ which becomes infinitely sharp at $d = \infty$.

Proposition 14. *Let $\delta_1, \delta_2 \in (0, 1)$ and assume (without loss of generality) that $\delta_2 \geq \delta_1$. The output of the SWAP gadget (see Algorithm 1) is purer than both inputs $\rho(\delta_1)$ and $\rho(\delta_2)$ if and only if*

$$\delta_2 - \delta_1 < (1 - \delta_1) \frac{2\delta_1(d - (d-1)\delta_1)}{d + 2\delta_1(d - (d-1)\delta_1)}. \tag{127}$$

Proof. Since we assumed that $\delta_2 \geq \delta_1$, the output is purer than both inputs whenever $\delta_1 > \delta'$ where δ' is given in Proposition 13. The denominator of δ' is always positive, so $\delta_1 > \delta'$ is equivalent to

$$\delta_1 \left(\left(1 + \frac{1}{d}\right) + \left(1 - \frac{1}{d}\right)(1 - \delta_1)(1 - \delta_2) \right) > \frac{\delta_1 + \delta_2}{2} + \frac{\delta_1 \delta_2}{d}. \tag{128}$$

If we multiply both sides by $2d$ and expand some products, we get

$$2\delta_1(d+1) + 2\delta_1(d-1)(1 - \delta_1 - \delta_2 + \delta_1\delta_2) > d(\delta_1 + \delta_2) + 2\delta_1\delta_2, \tag{129}$$

which can be simplified to an affine function in δ_2:

$$(3d\delta_1 - 2(d-1)\delta_1^2) - (d + 2d\delta_1 - 2(d-1)\delta_1^2)\delta_2 > 0. \tag{130}$$

This inequality is equivalent to

$$\frac{3d\delta_1 - 2(d-1)\delta_1^2}{d + 2d\delta_1 - 2(d-1)\delta_1^2} > \delta_2, \tag{131}$$

where the denominator agrees with eq. (127) and is always positive since $\delta_1 > \delta_1^2$. If we subtract δ_1 from both sides of eq. (131), the numerator becomes

$$\begin{aligned} (3d\delta_1 - 2(d-1)\delta_1^2) - (d + 2d\delta_1 - 2(d-1)\delta_1^2)\delta_1 &= 2d\delta_1 - 2(d-1)\delta_1^2 - 2d\delta_1^2 + 2(d-1)\delta_1^3 \\ &= 2\delta_1(1 - \delta_1)(d - (d-1)\delta_1), \end{aligned} \tag{132}$$

which agrees with eq. (127). □

Note that the right-hand side of eq. (127) is strictly positive for any $\delta_1 \in (0,1)$. In particular, if $\delta_1 = \delta_2 \in (0,1)$ then the left-hand side vanishes while the right-hand side is strictly positive. This means that for identical inputs, the output is always[3] purer than both inputs, giving another proof of Corollary 2 (see also Fig. 5).

[3]We exclude the extreme cases when both inputs are completely pure or maximally mixed, since in these cases the purity cannot be improved in principle: a pure state cannot get any purer, and the maximally mixed state has no information about the desired (pure) target state $|\psi\rangle$.