Supplementary material: Disti-mator

A.G. Maity and J.C.A. Casapao

November 2023

1 Bell vectorization

This section summarizes the code in the file `vectorizedBell.py`.

1.1 Bell diagonal states

The Bell-diagonal state is given by

$$\rho_D(\mathbf{q}) = q_1|\Phi^+\rangle\langle\Phi^+| + q_2|\Phi^-\rangle\langle\Phi^-| + q_3|\Psi^+\rangle\langle\Psi^+| + q_4|\Psi^-\rangle\langle\Psi^-|$$

where $\mathbf{q} = [q_1; q_2; q_3; q_4]$ and $q_1+q_2+q_3+q_4 = 1$ (we use semicolons for clarity). Here, we set $|\Phi^\pm\rangle = (|00\rangle \pm |11\rangle)/\sqrt{2}$ and $|\Psi^\pm\rangle = (|01\rangle \pm |10\rangle)/\sqrt{2}$. In this analysis, we assign a source-target pair $\rho_{A_1B_1}(\mathbf{q}) \otimes \rho_{A_2B_2}(\tilde{\mathbf{q}})$, defined over $\mathcal{H}_{A_1} \otimes \mathcal{H}_{B_1} \otimes \mathcal{H}_{A_2} \otimes \mathcal{H}_{B_2}$, to a Bell vector $\mathbf{q} \otimes \tilde{\mathbf{q}}$, defined in $\mathbb{R}^{16}$, where

$$\mathbf{q} \otimes \tilde{\mathbf{q}} := [q_1; q_2; q_3; q_4] \otimes [\tilde{q}_1; \tilde{q}_2; \tilde{q}_3; \tilde{q}_4].$$

We keep the order of the entries based on how the tensor product is evaluated.

1.2 Noise contribution in the state preparation stage

The relevant function here is `noisyPreparation`.

Each distillation round needs N source-target pairs $\rho_{A_1B_1} \otimes \rho_{A_2B_2}$. We suppose that each pair is not prepared at the same time, say the target pair is generated at time t after the source pair. Thus, the source pair have been subjected to noise while being distributed, which we model as a combination of depolarizing and dephasing channels on each side. Let $\lambda = (1 - e^{-\frac{t}{T_1}})$ be the depolarizing parameter, with characteristic time T_1, and let $\zeta = \frac{1}{2}(1 - e^{-\frac{t}{T_2}})$ be the dephasing parameter, with characteristic time T_2. We assume that the time it took to distribute the target pair is significantly smaller than the characteristic times. Then,

$$\rho_{A_1B_1} \otimes \rho_{A_2B_2} \mapsto [(\Delta^{\zeta_A}_{\mathrm{dph},A_1} \circ \Delta^{\lambda_A}_{\mathrm{dpo},A_1}) \otimes (\Delta^{\zeta_B}_{\mathrm{dph},B_1} \circ \Delta^{\lambda_B}_{\mathrm{dpo},B_1})](\rho_{A_1B_1}) \otimes \rho_{A_2B_2} = \rho^*_{A_1B_1} \otimes \rho_{A_2B_2},$$

where

$$\Delta^{\lambda_B}_{\mathrm{dpo},B_1}(\rho_{A_1B_1}) = (1-\lambda_B)\rho_{A_1B_1} + \lambda_B \mathrm{Tr}_{B_1}(\rho_{A_1B_1}) \otimes \frac{I_{B_1}}{2} = (1-\lambda_B)\rho_{A_1B_1} + \lambda_B \frac{I_{A_1B_1}}{4} =: \rho'_{A_1B_1},$$

followed by

$$\Delta^{\zeta_B}_{\mathrm{dph},B_1}(\rho'_{A_1B_1}) = (1-\zeta_B)\rho'_{A_1B_1} + \zeta_B I_{A_1} \otimes Z_{B_1} \rho'_{A_1B_1} I_{A_1} \otimes Z_{B_1} =: \rho''_{A_1B_1}.$$

Then,

$$\Delta^{\lambda_A}_{\mathrm{dpo},A_1}(\rho''_{A_1B_1}) = (1-\lambda_A)\rho''_{A_1B_1} + \lambda_A \frac{I_{A_1}}{2} \otimes \mathrm{Tr}_{A_1}(\rho''_{A_1B_1}) = (1-\lambda_A)\rho_{A_1B_1} + \lambda_A \frac{I_{A_1B_1}}{4} =: \rho'''_{A_1B_1},$$

and finally,

$$\Delta^{\zeta_B}_{\mathrm{dph},A_1}(\rho'''_{A_1B_1}) = (1-\zeta_A)\rho'''_{A_1B_1} + \zeta_A Z_{A_1} \otimes I_{B_1} \rho'''_{A_1B_1} Z_{A_1} \otimes I_{B_1} =: \rho^*_{A_1B_1}.$$

In terms of Bell vectors,

$$\Delta^{\lambda_B}_{\mathrm{dpo}}(\rho) \mapsto \mathbf{q}' := \left[\frac{\lambda_B}{4} + (1-\lambda_B)q_1; \frac{\lambda_B}{4} + (1-\lambda_B)q_2; \frac{\lambda_B}{4} + (1-\lambda_B)q_3; \frac{\lambda_B}{4} + (1-\lambda_B)q_4\right],$$

$$\Delta^{\zeta_B}_{\mathrm{dph}}(\rho) \mapsto \mathbf{q}' := [(1-\zeta_B)q_1 + \zeta_B q_2; (1-\zeta_B)q_2 + \zeta_B q_1; (1-\zeta_B)q_3 + \zeta_B q_4; (1-\zeta_B)q_4 + \zeta_B q_3].$$

1.3 Noisy local rotations

The relevant function here is `noisyRotationX`.

In some of the distillation setups, we require local rotations prior to local controlled gates in order to get the desired distillation statistics. For some Bell-diagonal state ρ_{AB}, we describe the action of a noisy local rotations as

$$\begin{aligned}\Lambda_{R_x}^{m_A,m_B}(\rho_{AB}) = \quad & (1-m_A)(1-m_B) R_x\left(-\frac{\pi}{2}\right) \otimes R_x\left(+\frac{\pi}{2}\right) \rho_{AB} R_x\left(-\frac{\pi}{2}\right)^{\dagger} \otimes R_x\left(+\frac{\pi}{2}\right)^{\dagger} \\ & + [m_A + m_B - m_A m_B] \frac{I_{AB}}{4}\end{aligned}$$

for $0 \leq m_i \leq 1$. Let $\mathbf{q} = [q_1; q_2; q_3; q_4]$ be the associated Bell vector to ρ_{AB}. Then,

$$\Lambda_{R_x}^{m_A,m_B}(\rho_{AB}) \mapsto \mathbf{q}' := (1-m_A)(1-m_B)[q_1; q_4; q_3; q_2] + \frac{m_A + m_B - m_A m_B}{4}[1; 1; 1; 1]$$

1.4 Noisy CNOT

The relevant function here is `noisyCNOT`.

In a distillation setup, Alice and Bob perform a bilocal XOR, i.e., a CNOT operation, on their respective halves. We describe the action of both noisy CNOT operations on $\rho = \rho_{A_1B_1} \otimes \rho_{A_2B_2}$ in terms of the parameters y_A and y_B, where $0 \leq y_i \leq 1$:

$$\begin{aligned}\Lambda_{\mathrm{CNOT}}^{y_A,y_B}(\rho) = \quad & (1-y_A)(1-y_B)\,(\mathrm{CNOT}_{A_1A_2} \otimes \mathrm{CNOT}_{B_1B_2})\,\rho(\,\mathrm{CNOT}_{A_1A_2} \otimes \mathrm{CNOT}_{B_1B_2}) \\ & + [y_A + y_B - y_A y_B] \frac{I_{A_1A_2B_1B_2}}{16}\end{aligned}$$

For noisy CNOT operations, we take the following operation on the Bell vector:

$$\begin{aligned}&(\mathrm{CNOT}_{A_1A_2} \otimes \mathrm{CNOT}_{B_1B_2})\,\rho(\,\mathrm{CNOT}_{A_1A_2} \otimes \mathrm{CNOT}_{B_1B_2}) \\ &\mapsto [q_1\tilde{q}_1; q_2\tilde{q}_2; q_1\tilde{q}_3; q_2\tilde{q}_4; q_2\tilde{q}_1; q_1\tilde{q}_2; q_2\tilde{q}_3; q_1\tilde{q}_4; q_3\tilde{q}_3; q_4\tilde{q}_4; q_3\tilde{q}_1; q_4\tilde{q}_2; q_4\tilde{q}_3; q_3\tilde{q}_4; q_4\tilde{q}_1; q_3\tilde{q}_2]\end{aligned}$$

1.5 Noisy measurements

The relevant functions here are `noisyMeasurementZtarget` and `noisyMeasurementXsource`.

Finally, we consider a noisy Z-measurement scenario. Here, we take the stochastic map $|0\rangle \mapsto |0\rangle$ and $|1\rangle \mapsto |1\rangle$, each with probability η_z, and $|0\rangle \mapsto |1\rangle$ and $|1\rangle \mapsto |0\rangle$, each with probability $1-\eta_z$. So, the associated POVMs when measuring the target pair are

$$\begin{aligned}I_{A_1B_1} \otimes M_{A_2B_2}^{00}(\eta_z^A, \eta_z^B) &= I_{A_1B_1} \otimes \big(\eta_z^A\eta_z^B|00\rangle\langle 00| + (1-\eta_z^A)(1-\eta_z^B)|11\rangle\langle 11| \\ &\quad + \eta_z^A(1-\eta_z^B)|01\rangle\langle 01| + (1-\eta_z^A)\eta_z^B|10\rangle\langle 10|\big)\,, \\ I_{A_1B_1} \otimes M_{A_2B_2}^{11}(\eta_z^A, \eta_z^B) &= I_{A_1B_1} \otimes \big((1-\eta_z^A)(1-\eta_z^B)|00\rangle\langle 00| + \eta_z^A\eta_z^B|11\rangle\langle 11| \\ &\quad + (1-\eta_z^A)\eta_z^B|01\rangle\langle 01| + \eta_z^A(1-\eta_z^B)|10\rangle\langle 10|\big)\,.\end{aligned}$$

Similarly, we consider a noisy X-measurement scenario. Here, we take the stochastic map $|+\rangle \mapsto |+\rangle$ and $|-\rangle \mapsto |-\rangle$, each with probability η_x, and $|+\rangle \mapsto |-\rangle$ and $|-\rangle \mapsto |+\rangle$, each with probability $1-\eta_x$. Thus, the associated POVMs when measuring the source pair are

$$\begin{aligned}M_{A_1B_1}^{++}(\eta_x^A, \eta_x^B) \otimes I_{A_2B_2} &= \big(\eta_x^A\eta_x^B|++\rangle\langle ++| + (1-\eta_x^A)(1-\eta_x^B)|--\rangle\langle --| \\ &\quad + \eta_x^A(1-\eta_x^B)|+-\rangle\langle +-| + (1-\eta_x^A)\eta_x^B|-+\rangle\langle -+|\big) \otimes I_{A_2B_2}, \\ M_{A_1B_1}^{--}(\eta_x^A, \eta_x^B) \otimes I_{A_2B_2} &= \big((1-\eta_x^A)(1-\eta_x^B)|++\rangle\langle ++| + \eta_x^A\eta_x^B|--\rangle\langle --| \\ &\quad + (1-\eta_x^A)\eta_x^B|+-\rangle\langle +-| + \eta_x^A(1-\eta_x^B)|-+\rangle\langle -+|\big) \otimes I_{A_2B_2}.\end{aligned}$$

We have to be careful at this step because of the entangling CNOT operations. That is, if $f^{y_A,y_B}_{\text{CNOT}}$ is the corresponding operation of $\Lambda^{y_A,y_B}_{\text{CNOT}}$ on $\mathbb{R}^{16}$, then $f^{y_A,y_B}_{\text{CNOT}}(\mathbf{q}\otimes\tilde{\mathbf{q}}) = \mathbf{f} \neq \mathbf{q}'\otimes\tilde{\mathbf{q}}'$ following our earlier discussion. Then, for a noisy Z measurement on the target pair,

$$\begin{aligned}
&\mathrm{Tr}(I_{A_1B_1}\otimes[M^{00}_{A_2B_2}(\eta^A_z,\eta^B_z)+M^{11}_{A_2B_2}(\eta^A_z,\eta^B_z)]) \\
&\quad= \quad (1-\eta^B_z-\eta^A_z(1-2\eta^B_z))(f_1+f_2+f_5+f_6+f_9+f_{10}+f_{13}+f_{14}) \\
&\qquad +(\eta^B_z+\eta^A_z(1-2\eta^B_z))(f_3+f_4+f_7+f_8+f_{11}+f_{12}+f_{15}+f_{16}).
\end{aligned}$$

On the other hand, for a noisy X measurement on the source pair,

$$\begin{aligned}
&\mathrm{Tr}([M^{++}_{A_1B_1}(\eta^A_x,\eta^B_x)+M^{--}_{A_1B_1}(\eta^A_x,\eta^B_x)]\otimes I_{A_2B_2}) \\
&\quad= \quad (1-\eta^B_x-\eta^A_x(1-2\eta^B_x))(f_1+f_2+f_3+f_4+f_9+f_{10}+f_{11}+f_{12}) \\
&\qquad +(\eta^B_x+\eta^A_x(1-2\eta^B_x))(f_5+f_6+f_7+f_8+f_{13}+f_{14}+f_{15}+f_{16}),
\end{aligned}$$

where f_i is the i-th component of $\mathbf{f}$. Multiplying each result by 1/2 gives the measurement statistics for $\{00\}$ and $\{++\}$, respectively.

2 Distillation protocols

This section summarizes the code in the file `distillationExperiment.py`.

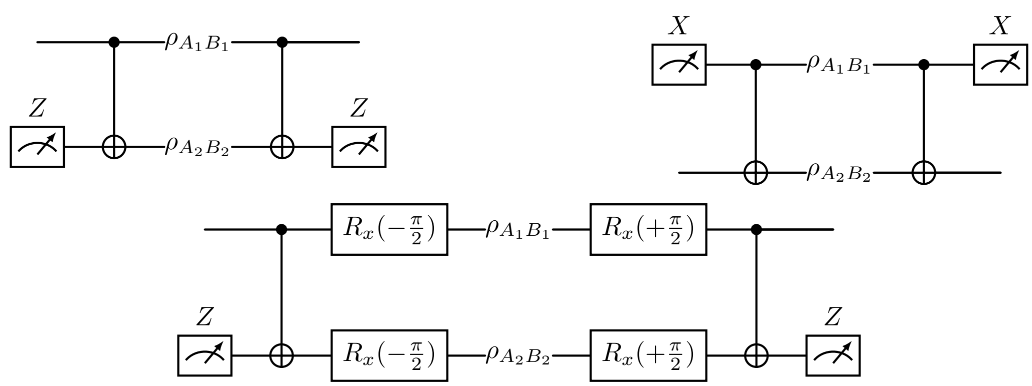

Figure 1: Distillation protocols for this work.

2.1 First distillation statistics

The relevant function here is `protocolExp1`. The first distillation protocol is described in Fig. 1 top left. The corresponding success probability $p^{(1)}$ is described by

$$p^{(1)} = \mathrm{Tr}\Big[I_{A_1B_1}\otimes M^{00}_{A_2B_2}(\eta^A_z,\eta^B_z)\Lambda^{y_A,y_B}_{\text{CNOT}}\Big([(\Delta^{\zeta_A}_{\text{dph},A_1}\circ\Delta^{\lambda_A}_{\text{dpo},A_1})\otimes(\Delta^{\zeta_B}_{\text{dph},B_1}\circ\Delta^{\lambda_B}_{\text{dpo},B_1})](\rho_{A_1B_1})\otimes\rho_{A_2B_2}\Big)\Big].$$

In the noiseless case, and $\rho_{A_2B_2}=\rho_{A_1B_1}(\mathbf{q})$, we find that

$$p^{(1)} = \frac{1}{2}\left((q_1+q_2)^2+(q_3+q_4)^2\right) =: \frac{1}{2}\left(x_1^2+(1-x_1)^2\right).$$

Following the results in the main text, we also expect that $p^{(1)}=p^{(1)}(x_1)$ in the noisy scenario, with the given set of gates.

2.2 Second distillation statistics

The relevant function here is `protocolExp2`. The second distillation protocol is described in Fig. 1 top right. The corresponding success probability $p^{(1)}$ is described by

$$p^{(2)} = \mathrm{Tr}\Big[M^{++}_{A_1B_1}(\eta_x^A,\eta_x^B)\otimes I_{A_2B_2}\Lambda^{y_A,y_B}_{\mathrm{CNOT}}\Big([(\Delta^{\zeta_A}_{\mathrm{dph},A_1}\circ\Delta^{\lambda_A}_{\mathrm{dpo},A_1})\otimes(\Delta^{\zeta_B}_{\mathrm{dph},B_1}\circ\Delta^{\lambda_B}_{\mathrm{dpo},B_1})](\rho_{A_1B_1})\otimes\rho_{A_2B_2}\Big)\Big].$$

In the noiseless case, and $\rho_{A_2B_2}=\rho_{A_1B_1}(\mathbf{q})$, we find that

$$p^{(2)} = \frac{1}{2}\left((q_1+q_3)^2+(q_2+q_4)^2\right) =: \frac{1}{2}\left(x_2^2+(1-x_2)^2\right).$$

Following the results in the main text, we also expect that $p^{(2)} = p^{(2)}(x_2)$ in the noisy scenario, with the given set of gates.

2.3 Third distillation statistics

The relevant function here is `protocolExp3`. The third distillation protocol is described in the bottom circuit in Fig. 1. The corresponding success probability $p^{(3)}$ is described by

$$\begin{aligned} p^{(3)} = \mathrm{Tr}\Big[& I_{A_1B_1}\otimes M^{00}_{A_2B_2}(\eta_z^A,\eta_z^B)\times \\ &\times\Lambda^{y_A,y_B}_{\mathrm{CNOT}}\circ(\Lambda^{m_A,m_B}_{R_x}\otimes\Lambda^{m_A,m_B}_{R_x})\Big([(\Delta^{\zeta_A}_{\mathrm{dph},A_1}\circ\Delta^{\lambda_A}_{\mathrm{dpo},A_1})\otimes(\Delta^{\zeta_B}_{\mathrm{dph},B_1}\circ\Delta^{\lambda_B}_{\mathrm{dpo},B_1})](\rho_{A_1B_1})\otimes\rho_{A_2B_2}\Big)\Big].\end{aligned}$$

In the noiseless case, and $\rho_{A_2B_2}=\rho_{A_1B_1}(\mathbf{q})$, we find that

$$p^{(3)} = \frac{1}{2}\left((q_1+q_4)^2+(q_2+q_3)^2\right) =: \frac{1}{2}\left(x_3^2+(1-x_3)^2\right).$$

Following the results in the main text, we also expect that $p^{(3)} = p^{(3)}(x_3)$ in the noisy scenario, with the given set of gates.

3 Inversion strategies

This section summarizes the code in the files `wernerEstimation.py` and `bellDiagEstimation.py`.

3.1 Werner state estimation I

As a special case, we consider a Werner state. Here,

$$q_1 = F, \qquad q_2=q_3=q_4=\frac{1-F}{3}.$$

where F is the fidelity with respect to $|\Phi^+\rangle$. Taking $F = 1-\frac{3w}{4}$, where w is the Werner parameter, we have

$$\rho_{AB}(w) = (1-w)|\Phi^+\rangle\langle\Phi^+| + w\frac{I_{AB}}{4}.$$

Our goal is to determine an estimation $\hat{w}$ following the first distillation protocol (Section 2.1), with some precision ϵ_w which we fix. Since a successful distillation happens when $F>1/2$, we only consider the range $0\le w<2/3$. We take both the source and target pairs to be the same right after generation. Also, the precision $\epsilon^{(1)}$ on $p^{(1)}$ is expected to be dependent on $\hat{w}$ and ϵ_w.

The inversion strategy that we adopt involves a bisection search algorithm. For the required ϵ_w, the number of iterations needed is bounded by

$$n_{1/2} \le \left\lceil \log_2\left(\frac{b_0-a_0}{\epsilon_w}\right)\right\rceil = \left\lceil \log_2\left(\frac{2}{3\epsilon_w}\right)\right\rceil,$$

where we have chosen $a_0 = 0$ and $b_0 = 2/3$ as the initial endpoints of the search.

The function for this inversion strategy is `invertWernerParamI` in `wernerEstimation.py`, with $\hat{w}$ as the output. For the associated left- and right-side precision in $\hat{p}^{(1)}$, we can evaluate them via the function `protocolExp1` in `distillationExperiment.py`, with the Bell vectors corresponding to $\hat{w} \pm \epsilon_w$ as inputs, and then determine the distance of the results from $\hat{p}^{(1)}$.

3.2 Werner state estimation II

For this inversion strategy, the precision $\epsilon^{(1)}$ for the empirical success probability is given instead of ϵ_w. We still use the same estimation method in the previous section to search for $\hat{w}$, as well as for $\hat{w}_p(\geq \hat{w})$ that corresponds to $\hat{p}^{(1)} - \epsilon^{(1)}$, and $\hat{w}_m(\leq \hat{w})$ that corresponds to $\hat{p}^{(1)} + \epsilon^{(1)}$ (since $p^{(1)}$ is monotonically decreasing with w). However, we have to take note that the precision of the bisection method should neither match $|\hat{w} - \hat{w}_p|$ nor $|\hat{w} - \hat{w}_m|$. We expect that the order of the precision of the estimation via distillation matches that of tomography. Thus, as a rule of thumb, we set the precision of the search as $\epsilon_w \leq 10^{-2}\epsilon^{(1)}$. After the search, we finally set $\epsilon_w \leq \max(|\hat{w} - \hat{w}_p|, |\hat{w} - \hat{w}_m|)$.

The function for this inversion strategy is `invertWernerParamII` in `wernerEstimation.py`, with $\hat{w}, \hat{w}_p, \hat{w}_m$ as the outputs.

3.3 Bell-diagonal state estimation I

We extend the inversion strategy to the general Bell-diagonal scenario. Given that $p^{(j)} = p^{(j)}(x_j)$, we can independently do a bisection search for the estimation $\hat{x}_j$ given some empirical $\hat{p}^{(j)}$. Then, we determine the estimated vector $\hat{\mathbf{q}} = [\hat{q}_1; \hat{q}_2; \hat{q}_3; \hat{q}_4]$ via

$$\begin{aligned}
q_1 &= \frac{1}{2}(-1 + x_1 + x_2 + x_3), \\
q_2 &= \frac{1}{2}(1 + x_1 - x_2 - x_3), \\
q_3 &= \frac{1}{2}(1 - x_1 + x_2 - x_3), \\
q_4 &= \frac{1}{2}(1 - x_1 - x_2 + x_3).
\end{aligned}$$

We set the precision of the bisection method to be ϵ_j, with $|x_j - \hat{x}_j| \leq \epsilon_j$, and with the number of iterations needed $n_{1/2,j}$ bounded by

$$n_{1/2,j} \leq \left\lceil \log_2 \left(\frac{b_0 - a_0}{\epsilon_j} \right) \right\rceil = \left\lceil \log_2 \left(\frac{1}{2\epsilon_j} \right) \right\rceil .$$

Here, we have chosen $a_0 = 1/2$ and $b_0 = 1$ as the initial endpoints of the search. Because the x_j's can be searched independently, we can parameterize the Bell vectors as $[x_1; 0; 1 - x_1; 0]$, $[x_2; 1 - x_2; 0; 0]$, and $[x_3; 1 - x_3; 0; 0]$ for the first, second, and third distillation protocols, respectively (since $x_j = q_1 + q_{j+1}$).

The function for this inversion strategy is `invertBellProtocolI` in `bellDiagEstimation.py`, with $\hat{\mathbf{x}} := [x_1; x_2; x_3]$ as the output. For the associated left- and right-side precision in $\hat{p}^{(j)}$, we can evaluate them via the functions `protocolExp1`, `protocolExp2`, and `protocolExp3` in `distillationExperiment.py`. Again, because of $p^{(j)} = p^{(j)}(x_j)$, we only need to consider $[x_1 \pm \varepsilon_1; 0; 1 - (x_1 \pm \varepsilon_1); 0]$, $[x_2 \pm \varepsilon_1; 1 - (x_2 \pm \varepsilon_1); 0; 0]$, and $[x_3 \pm \varepsilon_1; 1 - (x_3 \pm \varepsilon_1); 0; 0]$, respectively. Finally, we evaluate the Bell vector $\hat{\mathbf{q}}$ via $\hat{\mathbf{x}}$, which is facilitated by the function `convertToQ` in `bellDiagEstimation.py`.

3.4 Bell-diagonal state estimation II

For this inversion strategy, the precision $\epsilon^{(j)}$ for the empirical success probability is given instead of ϵ_j. We still use the same estimation method in the previous section to search for $\hat{x}_j$, as well as for $\hat{x}_{j,p}(\geq \hat{x}_j)$ that corresponds to $\hat{p}^{(j)} + \epsilon^{(j)}$, and $\hat{x}_{j,m}(\leq \hat{x}_j)$ that corresponds to $\hat{p}^{(j)} - \epsilon^{(j)}$ (since $p^{(j)}$ is monotonically increasing with x_j). Again, we have to take note that the precision of the bisection method should neither match $|\hat{x}_j - \hat{x}_{j,p}|$ nor $|\hat{x}_j - \hat{x}_{j,m}|$. We expect that the order of the precision of the estimation via distillation matches that of tomography. Thus, as a rule of thumb, we set the precision of the search as $\epsilon_j \leq 10^{-2}\epsilon^{(j)}$. After the search, we finally set $\epsilon_j \leq \max(|\hat{x}_j - \hat{x}_{j,p}|, |\hat{x}_j - \hat{x}_{j,m}|)$.

The function for this inversion strategy is `invertBellProtocolII` in `bellDiagEstimation.py`, with $\hat{\mathbf{x}} = [x_1; x_2; x_3]$, $\hat{\mathbf{x}}_p := [x_1 + \epsilon_1; x_2 + \epsilon_2; x_3 + \epsilon_3]$, $\hat{\mathbf{x}}_m := [x_1 - \epsilon_1; x_2 - \epsilon_2; x_3 - \epsilon_3]$ as the outputs.