Supplementary Materials for Optimized heteronuclear-ion quantum demodulator with "super-Heisenberg" and Heisenberg scalings

Jiawei Zhang, Wenqiang Ding, Haojie Du, Jintao Bu, Wenfei Yuan, Bin Wang, Wenjin Chen, Liang Chen, Jiachong Li, Geyi Ding, Fei Zhou, Qing-Shou Tan, and Mang Feng

Supplementary Note 1: Theoretical limit of double-parameter optimization

For the case of two-parameter estimation, it is essential to calculate the Fisher Information Matrix, e.g., the quantum Fisher information matrix (QFIM). For a unitary process U with a pure probe state $|\psi_0\rangle$, the entry of the QFIM is given by [1]

$$\mathcal{F}_{ab} = 4\,\mathrm{cov}_{|\psi_0\rangle}(\mathcal{H}_a, \mathcal{H}_b), \tag{S1}$$

where $\mathrm{cov}_{|\psi_0\rangle}(\mathcal{H}_a, \mathcal{H}_b)$ is the covariance. The QFI for the parameter x_a is given by $\mathcal{F}_{aa} = 4\,\mathrm{var}_{|\psi_0\rangle}(\mathcal{H}_a)$, where $\mathrm{var}_{|\psi_0\rangle}(\mathcal{H}_a) := \mathrm{cov}_{|\psi_0\rangle}(\mathcal{H}_a, \mathcal{H}_a)$ represents the variance of $\mathcal{H}_a$ with respect to the state $|\psi_0\rangle$. The operator $\mathcal{H}_a$ is defined as

$$\mathcal{H}_a := i\left(\partial_a U^\dagger\right) U = -iU^\dagger\left(\partial_a U\right), \tag{S2}$$

where $\mathcal{H}_a$ is a Hermitian operator for any parameter x_a as defined above. For unitary processes, the parameterized state remains pure for a pure probe state. In the context of frequency and amplitude demodulation, the general form of the QFIM simplifies to

$$\mathcal{F}_{ab} = (\partial_a \vec{S}) \cdot (\partial_b \vec{S}), \tag{S3}$$

where $a, b \in \{\omega, A\}$.

For optimal precision, the sum of variances in the estimated parameters ω and A is given by

$$(\Delta\omega^2 + \Delta A^2)_{\min} = \mathrm{Tr}[\mathcal{F}^{-1}] = \frac{\mathcal{F}_{aa} + \mathcal{F}_{bb}}{\mathcal{F}_{aa}\mathcal{F}_{bb} - (\mathcal{F}_{ab})^2} \geq \frac{1}{\mathcal{F}_{aa}} + \frac{1}{\mathcal{F}_{bb}}, \tag{S4}$$

where $\mathcal{F}^{-1}$ denotes the inverse of the QFIM. This relation sets a lower bound for the estimation precision of the two parameters based on the QFIM. From this relation, we find that the overall precision for simultaneous estimation of two parameters is lower than that for the single-parameter estimation due to introduction of additional uncertainty.

In the scenarios considered in our main text, the overall precision of frequency and amplitude is lower than Heisenberg scalings t^2. Therefore, we primarily focus on single-parameter estimation by using heteronuclear ions. When estimating the frequency ω_1 using $^{40}Ca^+$, we assume that a priori estimate of A_1^{est} has been obtained from measurements of $^{43}Ca^+$. Conversely, when estimating A_2, we assume that a priori estimate of ω_2^{est} has been derived from measurements of $^{40}Ca^+$. By making simultaneous measurements, we can achieve accurate detection of both the frequency and the amplitude.

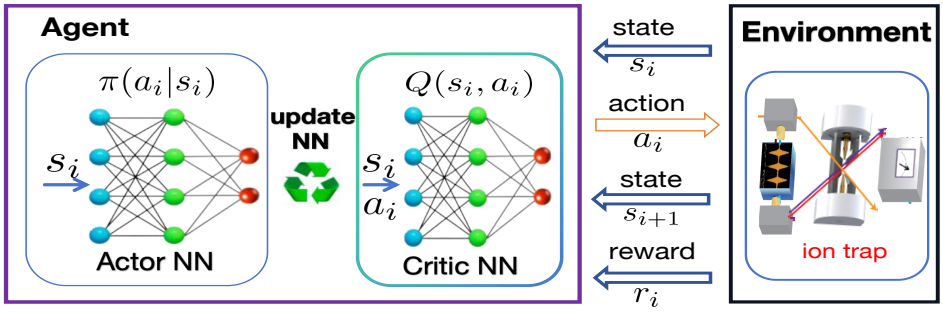
Supplementary Note 2: Introduction of proximal policy optimization algorithm

The proximal policy optimization (PPO) algorithm is a widely adopted reinforcement learning method, particularly effective in optimizing policies within complex environments. It improves upon traditional policy gradient methods by mitigating instability caused by large policy updates, imposing constraints on the step size during each update. Specifically, PPO introduces a "trust region" that limits deviations between the new and old policies through a carefully designed objective function. This function uses a clipping technique to prevent excessive updates, thereby balancing exploration and exploitation while enhancing learning stability and efficiency.

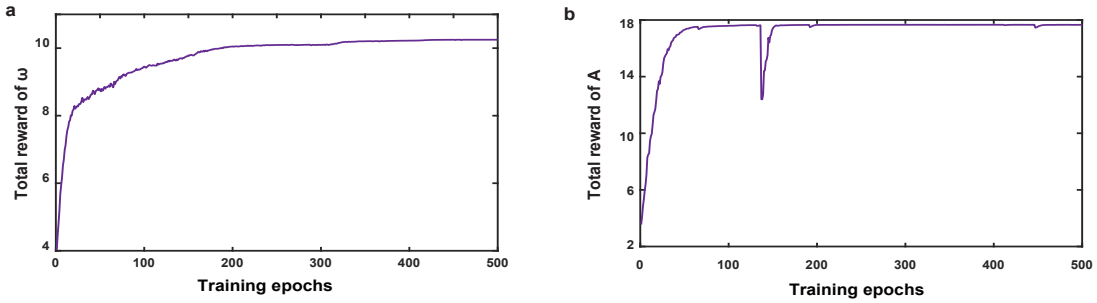
At the core of PPO is the objective function used for policy updates, which prevents large updates while optimizing expected rewards. The objective function is given by

$$L^{\mathrm{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t,\, \mathrm{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right], \tag{S5}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t|s_t)}$ is the probability ratio between the new policy π_θ and the old policy $\pi_{\theta_{\mathrm{old}}}$, $\hat{A}_t$ is the advantage estimate at time step t, ϵ is a hyperparameter controlling the range of allowable updates, $\mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)$ restricts the probability ratio to the range $[1-\epsilon, 1+\epsilon]$, ensuring that updates remain close to the old policy. This objective



Supplementary Figure 1. Schematic representation of the PPO learning process. The PPO agent consists of two neural networks (NNs): the actor NN and the critic NN. At each time step i, the actor NN selects an action a_i based on the current environment state s_i through the policy $\pi(a_i|s_i)$. The environment, modeled as a trapped ions system, then evolves to a new state s_{i+1} and provides a reward r_i to the agent. The agent uses this information (state, action, reward, and new state) to update both the actor and critic NNs, optimizing the policy $\pi(a|s)$ and value functions $Q(s, a)$ for improved decision-making.



Supplementary Figure 2. The total reward evolves with training epochs, where panel **a** (**b**) corresponds to frequency (amplitude) estimation.

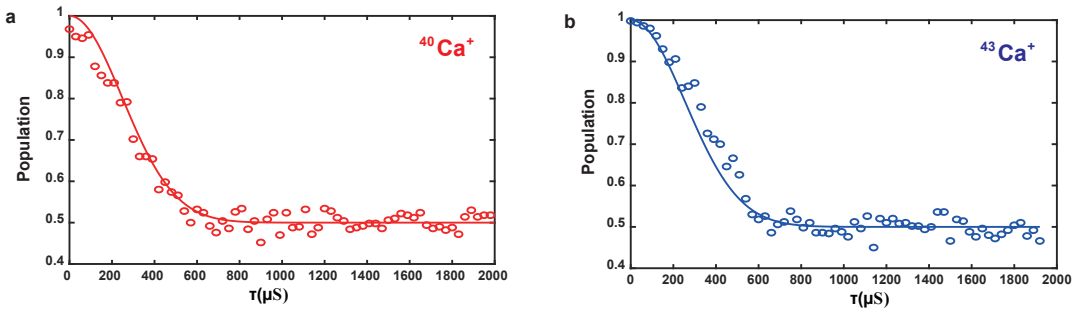
function minimizes the original probability ratio $r_t(\theta)\hat{A}_t$ and its clipped version, ensuring that updates remain within the trust region. This approach enhances stability while allowing effective learning, with the clipping mechanism being a key feature of PPO, preventing large and destabilizing policy changes. This encourages exploration by preventing premature convergence to suboptimal policies.

In our specific implementation, the RL process is illustrated in Supplementary Figure 1. To determine the optimal action sequence $f_i(t)$ over a given time interval, each episode is divided into a finite number of steps $i = 1, 2, 3, \ldots, N$. At each step i, the agent observes the current state s_i of the environment and selects an action a_i according to the policy $\pi(a_i|s_i)$. After executing the action, the environment transitions to a new state s_{i+1} and provides a reward r_i to the agent. The policy is iteratively updated based on the accumulated experience, with the objective of maximizing the cumulative reward R. The state s_i comprises three observables, $\mathrm{Tr}(\rho\sigma^j_{x,y,z})$ for $j = 1$ or $j = 2$, corresponding to either amplitude estimation or frequency estimation. These observables are computed using the QuTiP toolkit [2, 3].

We employed the Ray Python toolkit for training, with neural network parameters set to Ray's default values. For $N = 20$, the ideal total reward, based on the reward function, is 20. As shown in Supplementary Figure. 2, the mean reward converges to a constant after approximately 200 episodes for both frequency and amplitude estimation tasks. For frequency estimation, the final total reward is around 11, indicating that, although reinforcement learning is used, the model does not fully saturate the theoretical limit, but approaches the scaling of t^4. In contrast, for amplitude estimation, the final total reward reaches 18, demonstrating near saturation of the theoretical limit, achieving the optimal scaling of t^2.

Supplementary Note 3: Other experimental details

Experimentally, we use sympathetic cooling to simultaneously cool $^{40}Ca^+$ and $^{43}Ca^+$ ions. To examine the cooling effect, we simultaneously execute the Ramsey sequences to obtain the coherence time of $^{40}Ca^+$ and $^{43}Ca^+$ ions. The Ramsey sequence includes an initial $\pi/2$ carrier pulse exciting the lower state $|g\rangle$ of the qubits to the superposition state $\frac{\sqrt{2}}{2}(|g\rangle + |e\rangle)$, a waiting time τ and another $\pi/2$ carrier pulse. By varying the waiting time τ, the outcome is

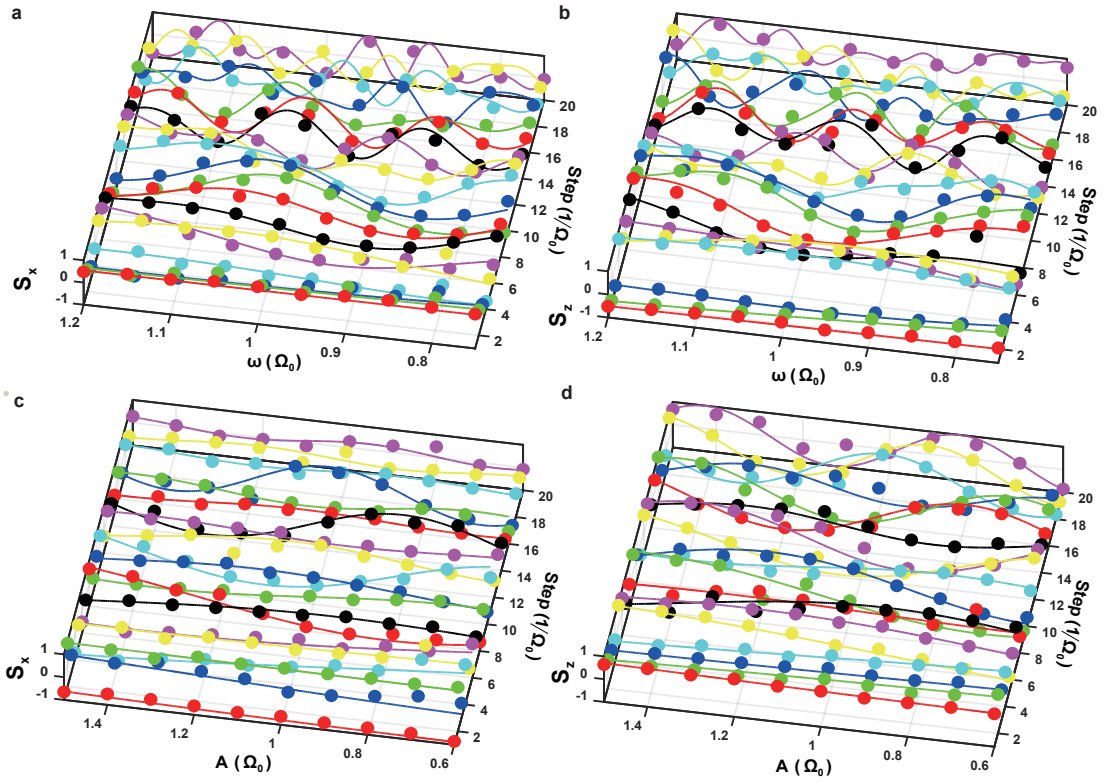


Supplementary Figure 3. Time evolution of population of $|e\rangle$ state in **a,** $^{40}Ca^{+}$ and **b,** $^{43}Ca^{+}$. Dots are experimental data obtained by 10 000 measurements.

governed by

$$P_{e_i} = [\exp(-\tau^2/T_2^2) + 1]/2, \tag{S6}$$

where P_{e_i} represents the population of $|e\rangle_i$, T_2 represents the coherence time and $i = 1$ ($i = 2$) represents $^{40}Ca^{+}$ ($^{43}Ca^{+}$) ion. Using the arbitrary waveform generator as well as the way mentioned in Methods, we execute the Ramsey pulses and simultaneously acquire the experimental data in Supplementary Figure. 3. Then, using Eq. (S6) to fit the data, we acquire the coherence time of $^{40}Ca^{+}$ and $^{43}Ca^{+}$ ions, respectively, 350 μS and 360 μS, which is sufficient for quantum demodulation.



Supplementary Figure 4. **a-b,** Experimental measurement of the Stokes parameter S_x (S_z) after each pulse (i.e., step) for global frequency estimation. **c-d,** Experimental measurement of the Stokes parameter S_x (S_z) after each pulse (i.e., step) for global amplitude estimation. Dots are experimental data obtained by 10 000 measurements for each data point and lines represent the theoretical simulations.

Next, we show the experimental results of $S_x \equiv \langle\sigma_x\rangle$ and $S_z \equiv \langle\sigma_z\rangle$ of every step for global frequency estimation and amplitude estimation shown in Supplementary Figure. 4. Cleanly, RL optimization improves the clarity of point

distinctions over time.

[1] Liu, J., et al J. Phys. A: Math. Theor. **53**, 023001 (2020).

[2] Johansson, J. R., Nation, P. D., & Nori, F. QuTiP: An open-source Python framework for the dynamics of open quantum systems. *Comput. Phys. Commun.* **183**, 1760 (2012).

[3] Johansson, J. R., Nation, P. D., & Nori, F. QuTiP 2: An open-source Python framework for the dynamics of open quantum systems. *Comput. Phys. Commun.* **184**, 1234 (2013).