

Supplementary Material: A Two-stage Optimization Method for Wide-range Single-electron Quantum Magnetic Sensing

Shiqian Guo¹, Jianqing Liu^{1*}, Thinh Le¹, Huaiyu Dai²

¹Department of Computer Science, North Carolina State University.

²Department of Electrical & Computer Engineering, North Carolina State University.
890 Oval Drive, Raleigh, 27659, NC, USA.

*Corresponding author(s). E-mail(s): jliu96@ncsu.edu;
Contributing authors: sguo26@ncsu.edu; tvle2@ncsu.edu; hdai@ncsu.edu;

1 Training of the BNN Estimator

The BNN estimator is implemented as a fully connected feed-forward neural network comprising two hidden layers, each with 32 neurons. The hidden layers employ the ReLU activation function, defined as $\max(x, 0)$. The input and output dimensions of the network are 1 and 100, respectively. We encode one training sample (or measurement outcome) from the dataset $\mathcal{D}_\omega$ as a one-hot vector and feed into the input layer of BNN. The output layer is normalized using the softmax function to produce a valid probability distribution over discrete magnetic field values. Specifically, the network outputs a posterior distribution $p(\omega_j | \mu_i)$, which represents the probability of the magnetic field taking the value ω_j conditioned on a single-shot measurement outcome μ_i .

The loss function is defined as the average negative log-likelihood across all training samples:

$$\mathcal{L} = -\frac{1}{b_\omega |\mathcal{D}_\omega|} \sum_{\ell=1}^{b_\omega} \sum_{i=1}^{|\mathcal{D}_\omega|} \log p(\omega_\ell | \mu_{i,\ell}), \quad (1)$$

where b_ω denotes the number of discrete values within the range $\omega_\ell \in (0, \omega_{\max})$, and $|\mathcal{D}_\omega|$ is the number of measurement outcomes in the dataset. To improve generalization and training stability, an L_2 regularization term is added to the loss. Optimization is performed using the Adam algorithm [1] for stochastic gradient descent. All neural network estimators are implemented and trained using the JAX framework [2].

Assuming conditional independence of measurements and a uniform prior over ω , the posterior distribution over ω_j given a sequence of M measurement outcomes $\vec{\mu}^{(M)} = \{\mu_1, \mu_2, \dots, \mu_M\}$ is given by the normalized product of individual posteriors:

$$p(\omega_j | \vec{\mu}) \propto \prod_{m=1}^M p(\omega_j | \mu_m). \quad (2)$$

equation (2) can be obtained via Bayesian updating. Specifically, the posterior distribution after incorporating the M -th measurement outcome μ_m is given by

$$p(\omega_j | \vec{\mu}^{(M)}) = \frac{p(\mu_M | \omega_j) p(\omega_j | \vec{\mu}^{(M-1)})}{\sum_{\ell=1}^{b_\omega} p(\mu_M | \omega_\ell) p(\omega_\ell | \vec{\mu}^{(M-1)})}. \quad (3)$$

where, according to Bayes' theorem, the conditional probability $p(\mu_m | \omega_j)$ can be expressed as

$$p(\mu_m | \omega_j) = \frac{p(\omega_j | \mu_m) \cdot p(\mu_m)}{p(\omega_j)}. \quad (4)$$

Substituting into the update rule yields

$$p(\omega_j | \vec{\mu}^{(M)}) \propto \frac{p(\omega_j | \mu_m) \cdot p(\mu_m)}{p(\omega_j)} \cdot p(\omega_j | \vec{\mu}^{(M-1)}). \quad (5)$$

By iterating this update over successive measurements, one arrives at Eq. (2). Here, $p(\omega_j)$ represents the prior distribution, and $p(\mu_m)$ serves as a normalization factor to ensure the posterior remains properly normalized. Thus, the posterior distribution shown in Figure 1 of the main document is given by

$$p(\omega_j | \vec{\mu}^{(M)}) \propto p(\omega_j)^{1-M} \prod_{m=1}^M p(\omega_j | \mu_m). \quad (6)$$

For the uniform prior used in this work, $p(\omega_j)^{1-M}$ is constant with respect to ω_j and is therefore absorbed into the normalization, yielding Eq. (2). And the final estimation value is then obtained via maximum a posteriori (MAP) inference:

$$\hat{\omega} = \arg \max_{\omega_j} [p(\omega_j | \vec{\mu})]. \quad (7)$$

A training dataset for the BNN estimator consists of measurement outcomes, i.e., classical bit-strings, denoted as $\mu_m \in \{0, 1\}$. These outcomes are collected at b_ω distinct true parameter values uniformly sampled across the range $(0, \omega_{\max})$. For every Ramsey measurement, we always choose $\tau = \pi/\omega_{\max}$ and $\varphi_0 = 0$ for quantum magnetic sensing. For each value of ω , a sequence of $|\mathcal{D}_\omega|$ Ramsey measurements are performed and their outcomes are recorded. Thus, the total number of bit-strings in the training dataset is $b_\omega \times |\mathcal{D}_\omega|$, forming a two-dimensional binary array of shape $(b_\omega, |\mathcal{D}_\omega|)$. The batch size is $b_\omega = 100$ and the dataset comprises $|\mathcal{D}_\omega| = 5,000$ samples. Training is performed over 8,000 iterations with a learning rate of 10^{-3} .

2 Training of the Federated RL Agent

The RL agent is implemented as a fully connected neural network comprising 5 hidden layers, each with 64 neurons. The input and output dimensions of the network are 5 and 2, respectively. A hyperbolic tangent (tanh) activation function is applied to the final layer to bound the output within the interval $[-1, 1]$.

The input (observation) to the agent after the k^{th} Ramsey experiment is denoted as

$$s_k = (\tilde{\omega}_k, \tilde{\sigma}_k, \Gamma_k, \tilde{k}, \tilde{R}_k), \quad (8)$$

where all components are scaled to lie within the range $[-1, 1]$. Here, $\tilde{\omega}_k$ represents the normalized mean of the estimation posterior, and

$$\tilde{\sigma}_k = -\frac{2}{10} \ln \sqrt{\Sigma_k} - 1, \quad (9)$$

with Σ_k denoting the covariance (or variance in the scalar case) around the posterior mean, defined as

$$\Sigma_k := \int (\omega - \hat{\omega}_k)(\omega - \hat{\omega}_k)^\top p(\omega | \vec{\mu}^{(k)}) d\omega, \quad (10)$$

The term Γ_k represents the correlation matrix of the posterior distribution. For the single-parameter estimation problem considered here, this simplifies to $\Gamma_k = 1$. The variables $\tilde{k}$ and $\tilde{R}_k$ denote the normalized experiment index and normalized consumed time resources, respectively.

The agent, parameterized by λ , outputs two actions: sensing time τ and control phase φ . This new sensing setup is then applied to the subsequent Ramsey measurement to refine the parameter estimation.

The local loss function, capturing the cumulative estimation error over a sequence of experiments, serves as the reward signal for training [3]. For agent n trained on dataset $\mathcal{D}_n$, the loss is defined as

$$\mathcal{L}_n(\boldsymbol{\lambda}) := \frac{1}{K|\mathcal{D}_n|} \sum_{k=1}^K \sum_{i=1}^{|\mathcal{D}_n|} \ell(\hat{\omega}_{i,k}, \omega_i), \quad (11)$$

where K is the maximum number of Ramsey experiments under the limited time resource $R_{\max}$, $\omega_i \in [n\omega_{\max}/N, (n+1)\omega_{\max}/N)$ denotes the true value and $\hat{\omega}_{i,k}$ is the estimate after the k^{th} Ramsey measurement. The estimate is computed as a weighted sum over basis points:

$$\hat{\omega}_{i,k} = \sum_{p=1}^P a_{i,p}^k \omega_{i,k,p}, \quad \sum_{p=1}^P a_{i,p}^k = 1, \quad (12)$$

where $\omega_{i,k,p}$ is the location of particle p for trajectory i after measurement k , and $a_{i,p}^k$ is its normalized posterior weight.

The global loss function is defined as the weighted average of all local losses:

$$\mathcal{L}(\boldsymbol{\lambda}) := \frac{1}{\sum_{n=1}^N |\mathcal{D}_n|} \sum_{n=1}^N |\mathcal{D}_n| \mathcal{L}_n(\boldsymbol{\lambda}). \quad (13)$$

The federated learning objective is to determine the optimal global parameters:

$$\boldsymbol{\lambda}^* = \arg \min_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda}). \quad (14)$$

During each training iteration t , local model parameters are updated using the Adam optimizer as

$$\boldsymbol{\lambda}_n^{t+1} = \boldsymbol{\lambda}_n^t - \eta \nabla \mathcal{L}_n(\boldsymbol{\lambda}_n^t), \quad (15)$$

where η is the learning rate. The global parameters are then updated by aggregating the local updates:

$$\boldsymbol{\lambda}^{t+1} = \sum_{n=1}^N \frac{|\mathcal{D}_n|}{\sum_{n=1}^N |\mathcal{D}_n|} \boldsymbol{\lambda}_n^{t+1}. \quad (16)$$

The RL agent is trained over 2,048 iterations with a batch size of 1,024 and a learning rate of $\eta = 10^{-3}$. The number of agents participating in the federated setting is set to 10. During training, the number of particles P used to represent the posterior distribution is chosen according to the maximum frequency range: $P = 240$ for $\omega_{\max} = 5$ MHz and 10 MHz, and $P = 240 \times 2$ for $\omega_{\max} = 20$ MHz. For performance evaluation in the online inference phase, 2,000 test episodes are conducted. The coherence time of the Ramsey experiment T_2^* is set as $96\mu\text{s}$ [4].

The number of local agents N in our federated RL framework was chosen based on a trade-off between learning performance and computational cost. Specifically, we selected $N = 10$ to divide the global estimation range $[0, \omega_{\max}]$ into 10 equal subranges, enabling each agent to focus on a simpler local task. This modularization has the potential to improve the model training convergence, compared to training a single global agent over the full range. From a learning accuracy perspective, training a single agent over the full ω -range leads to a significantly harder optimization problem due to the broader and more complex parameter space. This often results in slow convergence and poor generalization. In what follows, we provide a convergence analysis of the federated RL scheme.

Let $\mathbf{g}_t^{(n)}$ denote an unbiased stochastic gradient estimator of $\nabla \mathcal{L}_n(\boldsymbol{\lambda}_t)$ at iteration t . Define the per-agent variance as

$$\sigma_n^2 := \mathbb{E} \left\| \mathbf{g}_t^{(n)} - \nabla \mathcal{L}_n(\boldsymbol{\lambda}_t) \right\|^2. \quad (17)$$

Using the law of total variance, we decompose σ_n^2 into two contributions:

$$\sigma_n^2 = \underbrace{\mathbb{E}_{\omega_i \in \mathcal{D}_n} [\text{Var}(\mathbf{g}_t^{(n)} \mid \omega_i)]}_{\text{intrinsic gradient noise}} + \underbrace{\text{Var}_{\omega_i \in \mathcal{D}_n} (\nabla \mathcal{L}_n(\boldsymbol{\lambda}_t; \omega_i))}_{\text{variance across } \omega \text{ in subrange}}. \quad (18)$$

By assumption, the stochastic gradient has bounded per- ω variance:

$$\mathbb{E}_{\omega_i \in \mathcal{D}_n} [\text{Var}(\mathbf{g}_t^{(n)} \mid \omega_i)] = \mathbb{E} \|\mathbf{g}_t^{(n)} - \nabla \mathcal{L}_n(\boldsymbol{\lambda}_t; \omega_i)\|^2 \leq v_0. \quad (19)$$

Assuming the gradient of $\mathcal{L}_n$ with respect to $\boldsymbol{\lambda}$ is Lipschitz continuous in ω with constant H , and that ω is uniformly distributed, $\omega \sim \mathcal{U}[0, \omega_{\max}]$, we have $\text{Var}(\omega) = \omega_{\max}^2 / (12N^2)$, where the subrange length is $\omega_{\max}/N$. Consequently, the variance of the gradient over the subrange $\mathcal{D}_n$ is bounded by

$$\text{Var}_{\omega_i \in \mathcal{D}_n} (\nabla \mathcal{L}_n(\boldsymbol{\lambda}_t; \omega_i)) \leq H^2 \frac{\omega_{\max}^2}{12N^2}. \quad (20)$$

Thus, the per-agent variance bound is

$$\sigma_n^2 \leq v_0 + \frac{H^2 \omega_{\max}^2}{12N^2}. \quad (21)$$

After E local steps, different agents' gradients may differ because each sees a different subrange. Define the heterogeneity drift after a local step as

$$\zeta^2 := \frac{1}{N} \sum_{n=1}^N \left\| \nabla \mathcal{L}_n(\boldsymbol{\lambda}_t) - \nabla \mathcal{L}(\boldsymbol{\lambda}_t) \right\|^2. \quad (22)$$

Assuming an L_0 -smooth objective $\mathcal{L}(\boldsymbol{\lambda})$, after one global update we have

$$\mathcal{L}(\boldsymbol{\lambda}_{t+1}) \leq \mathcal{L}(\boldsymbol{\lambda}_t) + \langle \nabla \mathcal{L}(\boldsymbol{\lambda}_t), \boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_t \rangle + \frac{L_0}{2} \|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_t\|^2. \quad (23)$$

Using the standard FedAvg analysis (e.g., [5]) for $\eta \leq 1/(2L_0)$ over T global iterations, the stationary bound on the average squared gradient norm is

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\nabla \mathcal{L}(\boldsymbol{\lambda}_t)\|^2 \leq \frac{2(\mathcal{L}(\boldsymbol{\lambda}_0) - \mathcal{L}(\boldsymbol{\lambda}^*))}{\eta T} + C_1 \eta \bar{\sigma}^2 + C_2 \eta^2 E \zeta^2, \quad (24)$$

where $C_1 = 1$, $C_2 = 4L_0^2$, and $\bar{\sigma}^2 = \frac{1}{N} \sum_{n=1}^N \sigma_n^2$.

Substituting the per-agent variance bounds into the convergence result gives

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\nabla \mathcal{L}(\boldsymbol{\lambda}_t)\|^2 \leq \frac{2(\mathcal{L}(\boldsymbol{\lambda}_0) - \mathcal{L}(\boldsymbol{\lambda}^*))}{\eta T} + \eta v_0 + \frac{C_1 \eta H^2 \omega_{\max}^2}{12N^2} + C_2 \eta^2 E \zeta^2. \quad (25)$$

From this bound, we see that Federated RL can converge to a stationary point of the global loss function, but this stationary point may not be the global optimum. Next, we derive a stronger suboptimality bound under the additional idealized assumption that every local objective $\mathcal{L}_n(\boldsymbol{\lambda})$ is convex and L_0 -smooth with respect to the model parameters $\boldsymbol{\lambda}$. Although the squared-error loss is convex with respect to the estimated output, its composition with the implemented neural-network policy is generally non-convex with respect to $\boldsymbol{\lambda}$. Consequently, the following convex analysis should be interpreted as an analytical benchmark illustrating the effects of stochastic variance and federated heterogeneity, rather than as a formal convergence guarantee for the implemented deep-RL policy.

Under this additional convexity assumption, the convergence bound can be expressed in terms of the aggregated variance $\tilde{\sigma}^2$. Using the per-agent decomposition introduced above, one can define

$$\tilde{\sigma}^2 := \bar{\sigma}^2 + \zeta^2, \quad \bar{\sigma}^2 = \frac{1}{N} \sum_{n=1}^N \sigma_n^2 \leq v_0 + \frac{H^2 \omega_{\max}^2}{12N^2}. \quad (26)$$

The aggregated gradient estimator $\mathbf{g}_t$ used by the global agent satisfies

$$\mathbb{E}[\mathbf{g}_t \mid \boldsymbol{\lambda}_t] = \nabla \mathcal{L}(\boldsymbol{\lambda}_t), \quad \mathbb{E} \|\mathbf{g}_t - \nabla \mathcal{L}(\boldsymbol{\lambda}_t)\|^2 \leq \tilde{\sigma}^2, \quad (27)$$

where $\tilde{\sigma}^2$ is an aggregated variance bound encompassing the intrinsic per- ω variance v_0 , the per-agent heterogeneity $H^2\omega_{\max}^2/(12N^2)$, and the FedAvg drift contributions.

In addition to the assumptions introduced above (unbiased gradients, bounded per- ω variance v_0 , Lipschitz continuity in ω with constant H , etc.), we assume that each local loss function $\mathcal{L}_n(\boldsymbol{\lambda}; \omega)$ is convex. Consequently, the global loss $\mathcal{L}(\boldsymbol{\lambda})$ is also convex. Let $\boldsymbol{\lambda}^* := \arg \min_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda})$. We show that the weighted-average iterate $\tilde{\boldsymbol{\lambda}}_T := \arg \min_{\boldsymbol{\lambda}} \mathbb{E}[\mathcal{L}(\boldsymbol{\lambda}^t)]$ satisfies $\mathbb{E}[\mathcal{L}(\tilde{\boldsymbol{\lambda}}_T)] \rightarrow \mathcal{L}(\boldsymbol{\lambda}^*)$ under a standard diminishing step-size schedule.

The global update after aggregation is $\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t - \eta \mathbf{g}_t$. Expand the squared distance to the optimum and take conditional expectation:

$$\begin{aligned} \mathbb{E}[\|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}^*\|^2 \mid \boldsymbol{\lambda}_t] &= \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^*\|^2 - 2\eta \langle \nabla \mathcal{L}(\boldsymbol{\lambda}_t), \boldsymbol{\lambda}_t - \boldsymbol{\lambda}^* \rangle + \eta^2 \mathbb{E}[\|\mathbf{g}_t\|^2 \mid \boldsymbol{\lambda}_t] \\ &= \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^*\|^2 - 2\eta \langle \nabla \mathcal{L}(\boldsymbol{\lambda}_t), \boldsymbol{\lambda}_t - \boldsymbol{\lambda}^* \rangle \\ &\quad + \eta^2 (\|\nabla \mathcal{L}(\boldsymbol{\lambda}_t)\|^2 + \mathbb{E}\|\mathbf{g}_t - \nabla \mathcal{L}(\boldsymbol{\lambda}_t)\|^2) \\ &\leq \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^*\|^2 - 2\eta \langle \nabla \mathcal{L}(\boldsymbol{\lambda}_t), \boldsymbol{\lambda}_t - \boldsymbol{\lambda}^* \rangle + \eta^2 \|\nabla \mathcal{L}(\boldsymbol{\lambda}_t)\|^2 + \eta^2 \tilde{\sigma}^2. \end{aligned}$$

By convexity,

$$\mathcal{L}(\boldsymbol{\lambda}_t) - \mathcal{L}(\boldsymbol{\lambda}^*) \leq \langle \nabla \mathcal{L}(\boldsymbol{\lambda}_t), \boldsymbol{\lambda}_t - \boldsymbol{\lambda}^* \rangle. \quad (28)$$

Therefore

$$\mathbb{E}[\|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}^*\|^2 \mid \boldsymbol{\lambda}_t] \leq \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^*\|^2 - 2\eta (\mathcal{L}(\boldsymbol{\lambda}_t) - \mathcal{L}(\boldsymbol{\lambda}^*)) + \eta^2 \|\nabla \mathcal{L}(\boldsymbol{\lambda}_t)\|^2 + \eta^2 \tilde{\sigma}^2. \quad (29)$$

Using L_0 -smoothness and convexity one has the standard inequality $\|\nabla \mathcal{L}(\boldsymbol{\lambda}_t)\|^2 \leq 2L_0(\mathcal{L}(\boldsymbol{\lambda}_t) - \mathcal{L}(\boldsymbol{\lambda}^*))$. Substituting and assuming $\eta \leq 1/(2L_0)$ yields

$$\mathbb{E}[\|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}^*\|^2 \mid \boldsymbol{\lambda}_t] \leq \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^*\|^2 - \eta (\mathcal{L}(\boldsymbol{\lambda}_t) - \mathcal{L}(\boldsymbol{\lambda}^*)) + \eta^2 \tilde{\sigma}^2. \quad (30)$$

Taking full expectation and summing $t = 0, \dots, T-1$ gives

$$\mathbb{E}\|\boldsymbol{\lambda}_T - \boldsymbol{\lambda}^*\|^2 \leq \mathbb{E}\|\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}^*\|^2 - \sum_{t=0}^{T-1} \eta \mathbb{E}[\mathcal{L}(\boldsymbol{\lambda}_t) - \mathcal{L}(\boldsymbol{\lambda}^*)] + \tilde{\sigma}^2 T \eta^2. \quad (31)$$

Rearranging,

$$\sum_{t=0}^{T-1} \eta \mathbb{E}[\mathcal{L}(\boldsymbol{\lambda}_t) - \mathcal{L}(\boldsymbol{\lambda}^*)] \leq \|\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}^*\|^2 + \tilde{\sigma}^2 T \eta^2. \quad (32)$$

By convexity of $\mathcal{L}$ and the definition of $\tilde{\boldsymbol{\lambda}}_T$ and $\tilde{\sigma}^2$,

$$\mathbb{E}[\mathcal{L}(\tilde{\boldsymbol{\lambda}}_T)] - \mathcal{L}(\boldsymbol{\lambda}^*) \leq \frac{\|\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}^*\|^2}{\eta T} + \eta v_0 + \frac{C_1 \eta H^2 \omega_{\max}^2}{12 N^2} + C_2 \eta^2 E \zeta^2. \quad (33)$$

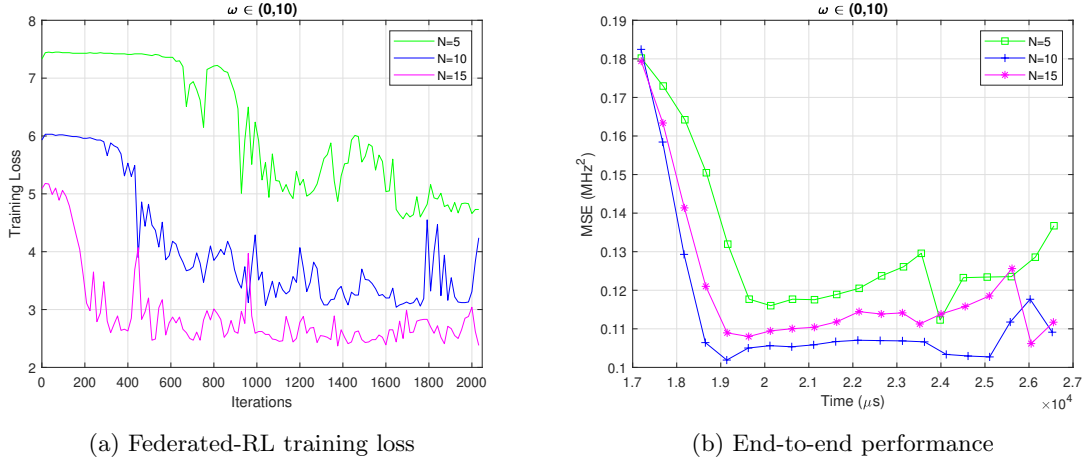
Now, let us examine the right hand side of the inequality. The first term decays with the total number of iterations T . The second term, ηv_0 , captures the irreducible stochastic noise. The third term scales as $H^2 \omega_{\max}^2 / N^2$ and decreases with the number of subranges N , representing the per-agent variance. The last term reflects the heterogeneity drift induced by model federation. As N increases, the data heterogeneity across agents grows, so ζ^2 itself increases with N .

Accordingly, by selecting an appropriate N and maintaining moderate η and E , Federated RL can converge to a lower expected loss than a global single-agent approach, which coincides with our numerical findings in Figure 5 of the main manuscript. This strategy reduces the variance due to a wide ω range while limiting the heterogeneity drift, i.e., $\frac{C_1 \eta H^2 \omega_{\max}^2}{12 N^2} + C_2 \eta^2 E \zeta^2 \leq \frac{C_1 \eta H^2 \omega_{\max}^2}{12}$.

To enable a controlled comparison between the proposed Federated-RL method and the single-agent RL baseline, we use the same per-agent network architecture, optimizer, learning rate, particle resolution, action space, and loss definition. From a learning efficiency perspective, the federated setup has implications for training cost. We compare it to the single-agent baseline in terms of memory and time complexity in parallel:

	Baseline RL	Federated RL
Memory Complexity	$\mathcal{O}(\mathcal{D}_n \cdot K \cdot P)$	$\mathcal{O}(N \cdot \mathcal{D}_n \cdot K \cdot P)$
Time Complexity	$\mathcal{O}(\mathcal{D}_n \cdot K \cdot P)$	$\mathcal{O}(\mathcal{D}_n \cdot K \cdot P)$

To assess the sensitivity of our method to the number of agents, we conducted an ablation study with $N \in \{5, 10, 15\}$. To ensure consistent particle resolution per agent, the corresponding number of particles was set as $P \in \{240 \times 2, 240, 240\}$, respectively. The results in Supplementary Figure 1 illustrate the impact of N on the training performance and the final estimation mean squared error (MSE). We observe that training performance improves with increasing N , indicating that too few agents $N = 5$ limit the benefits of task decomposition and make local learning less effective. Conversely, too many agents ($N = 15$) increase memory or time costs without significant gains in accuracy.



Supplementary Figure 1: Training and evaluation performance as affected by the number of agents.

3 Complexity Analysis of the Protocol

The computational overhead associated with both the BNN and the federated RL components is summarized in the insights below.

3.1 Offline Training Phase – BNN

Training Time Complexity. Assuming $|\mathcal{D}_\omega|$ weight samples per input and a batch size b_ω , the BNN training time per step scales as:

$$\text{Time}_{\text{BNN-Train}} = \mathcal{O}(|\mathcal{D}_\omega| \cdot b_\omega \cdot S_1),$$

where S_1 is the number of trainable parameters in the BNN, typically a small multilayer perceptron with 32 neurons per layer.

Memory Complexity. The memory overhead primarily comes from intermediate activations and gradient storage over all samples:

$$\text{Memory}_{\text{BNN-Train}} = \mathcal{O}(|\mathcal{D}_\omega| \cdot b_\omega \cdot 32 + S_1).$$

Inference Cost. During deployment, we use $|\mathcal{D}_\omega|_{\text{test}} \in [10, 100]$ Monte Carlo samples for posterior averaging:

$$\text{Time}_{\text{BNN-Infer}} = \mathcal{O}(|\mathcal{D}_\omega|_{\text{test}} \cdot b_\omega \cdot S_1), \quad \text{Memory}_{\text{BNN-Infer}} = \mathcal{O}(|\mathcal{D}_\omega|_{\text{test}} \cdot b_\omega \cdot 32 + S_1).$$

Inference is substantially cheaper than training and can be accelerated by using fewer samples or mean weights.

3.2 Offline Training Phase – Federated RL

Memory Complexity. For each RL agent trained over $|\mathcal{D}_n|$ samples and K measurements with P particles:

$$\text{Memory}_{\text{per agent}} = \mathcal{O}(|\mathcal{D}_n| \cdot K \cdot P).$$

The total memory scales as:

$$\text{Memory}_{\text{FedRL}} = \begin{cases} \mathcal{O}(|\mathcal{D}_n| \cdot K \cdot P), & (\text{sequential}) \\ \mathcal{O}(N \cdot |\mathcal{D}_n| \cdot K \cdot P), & (\text{parallel}) \end{cases}$$

Time Complexity. The time cost per update step per agent is:

$$\text{Time}_{\text{FedRL}} = \begin{cases} \mathcal{O}(|\mathcal{D}_n| \cdot K \cdot P), & (\text{parallel}) \\ \mathcal{O}(N \cdot |\mathcal{D}_n| \cdot K \cdot P), & (\text{sequential}) \end{cases}$$

FedAvg incurs only a minor overhead of $\mathcal{O}(S_2)$, where S_2 is the RL model size.

3.3 Inference Phase – BNN + Federated RL Agent

At runtime, only the global RL policy and BNN are used for adaptive control. The per-measurement inference cost is:

$$\text{Time}_{\text{Inference}} = \mathcal{O}(S_1 + K),$$

which matches the cost of the single-agent baseline. Local agents are not involved at test time.

In our simulation experimental setup, the total inference time per measurement (BNN + Federated RL) is approximately $70 \sim 140 \mu\text{s}$, comparable to the NV spin initialization time ($\sim 200 \mu\text{s}$) [4]. Moreover, the initialization and inference stages can be executed in parallel, enabling real-time sensing without additional latency. Although BNN training incurs extra cost due to sampling, this is efficiently handled by GPU parallelism. The federated RL module scales with agent count in memory, but can be parallelized for efficient training. Most importantly, inference cost is low and equivalent to baseline methods, ensuring practical real-time deployment. All simulations in this work were performed on an NVIDIA RTX 4090 GPU.

3.4 Bayesian Cramér–Rao bound in two-stage measurement

For estimating a single parameter ω , the Bayesian Cramér–Rao bound (BCRB) reads [3]:

$$\varepsilon_\omega^2 \geq \frac{1}{\mathbb{E}_\omega[I(\omega|\tau, \varphi)] + I(\pi)}, \quad (34)$$

where $I(\omega|\tau, \varphi)$ is the Fisher information (FI) for a measurement with control settings (τ, φ) , and $I(\pi)$ is the FI of the prior $\pi(\omega)$.

The FI per measurement takes the form

$$I(\omega|\tau, \varphi) = \frac{\left(\frac{\tau}{2} e^{-\tau/T_2^*} \sin(\omega\tau + \varphi)\right)^2}{\frac{1}{4} [1 - e^{-2\tau/T_2^*} \cos^2(\omega\tau + \varphi)]}. \quad (35)$$

Non-adaptive stage.

For M repeated measurements, the total FI is additive:

$$I(\omega|\boldsymbol{\tau}, \boldsymbol{\varphi}) = \sum_{m=1}^M I(\omega|\tau_m, \varphi_m) \leq M \sup_{\tau, \varphi} I(\omega|\tau, \varphi). \quad (36)$$

All measurements use a fixed evolution time $\tau_{\min}$, then $M = T_{\text{na}}/\tau_{\min}$. Approximating at the optimal phase ($\omega\tau_{\min} + \varphi_0 \approx \pi/2$) and including decoherence:

$$I_{\text{na}} \approx M \tau_{\min}^2 e^{-2\tau_{\min}/T_2^*} = T_{\text{na}} \tau_{\min} e^{-2\tau_{\min}/T_2^*}. \quad (37)$$

Considering per-shot overhead τ_0 for spin initialization and readout, $M' = T_{\text{na}}/(\tau_{\text{min}} + \tau_0)$, giving

$$I'_{\text{na}} \approx M' \tau_{\text{min}}^2 e^{-2\tau_{\text{min}}/T_2^*} = \frac{T_{\text{na}} \tau_{\text{min}}^2 e^{-2\tau_{\text{min}}/T_2^*}}{\tau_{\text{min}} + \tau_0}. \quad (38)$$

Adaptive stage.

In the second stage, with total sensing time T_{a} , the achievable FI is bounded by [3]:

$$I_{\text{a}} \leq T_{\text{a}} \sup_{\tau, \varphi} \frac{I(\omega|\tau, \varphi)}{\tau}. \quad (39)$$

Including per-shot overhead τ_0 :

$$I'_{\text{a}} \leq T_{\text{a}} \sup_{\tau, \varphi} \frac{I(\omega|\tau, \varphi)}{\tau + \tau_0}. \quad (40)$$

Optimizing over φ yields $\varphi_{\text{opt}} = \pi/2 - \omega\tau$, leading to

$$I(\omega|\tau, \varphi_{\text{opt}}) = \tau^2 e^{-2\tau/T_2^*}, \quad \frac{I(\omega|\tau, \varphi_{\text{opt}})}{\tau} = \tau e^{-2\tau/T_2^*}, \quad \frac{I(\omega|\tau, \varphi_{\text{opt}})}{\tau + \tau_0} = \frac{\tau^2 e^{-2\tau/T_2^*}}{\tau + \tau_0}. \quad (41)$$

Maximizing over τ gives

$$\tau_{\text{opt}} = \frac{T_2^*}{2}, \quad \sup_{\tau} \frac{I(\omega|\tau, \varphi_{\text{opt}})}{\tau} = \frac{T_2^*}{2e}, \quad (42)$$

$$\tau'_{\text{opt}} = \frac{T_2^*}{4} - \frac{\tau_0}{2} + \frac{1}{4} \sqrt{(T_2^*)^2 + 12T_2^* \tau_0 + 4\tau_0^2}, \quad \sup_{\tau} \frac{I(\omega|\tau, \varphi_{\text{opt}})}{\tau + \tau_0} = \frac{\tau_{\text{opt}}'^2 e^{-2\tau'_{\text{opt}}/T_2^*}}{\tau'_{\text{opt}} + \tau_0}. \quad (43)$$

Thus, the total adaptive FI is

$$I_{\text{a}} = T_{\text{a}} \frac{T_2^*}{2e} \approx 0.184 T_{\text{a}} T_2^*, \quad I'_{\text{a}} = T_{\text{a}} \frac{\tau_{\text{opt}}'^2 e^{-2\tau'_{\text{opt}}/T_2^*}}{\tau'_{\text{opt}} + \tau_0}. \quad (44)$$

Two-stage BCRB.

Combining the stage 1 posterior with the adaptive stage, the overall BCRB reads

$$\varepsilon_{\omega}^2 \geq \frac{1}{I_{\text{a}} + I(\pi_1)}, \quad (45)$$

where the posterior FI is

$$I(\pi_1) = I(\pi) + I_{\text{na}} = \frac{12}{\omega_{\text{max}}^2} + T_{\text{na}} \tau_{\text{min}} e^{-2\tau_{\text{min}}/T_2^*}. \quad (46)$$

Following Refs. [3, 6], assuming a uniform prior $\pi(\omega)$ over $[0, \omega_{\text{max}}]$ gives

$$I(\pi) = \frac{1}{\text{Var}(\omega)} = \frac{12}{\omega_{\text{max}}^2}. \quad (47)$$

Finally, the BCRB reads

$$\varepsilon_{\omega}^2 \geq \frac{1}{0.184 T_{\text{a}} T_2^* + \frac{12}{\omega_{\text{max}}^2} + T_{\text{na}} \tau_{\text{min}} e^{-2\tau_{\text{min}}/T_2^*}}, \quad (48)$$

and including per-shot overhead:

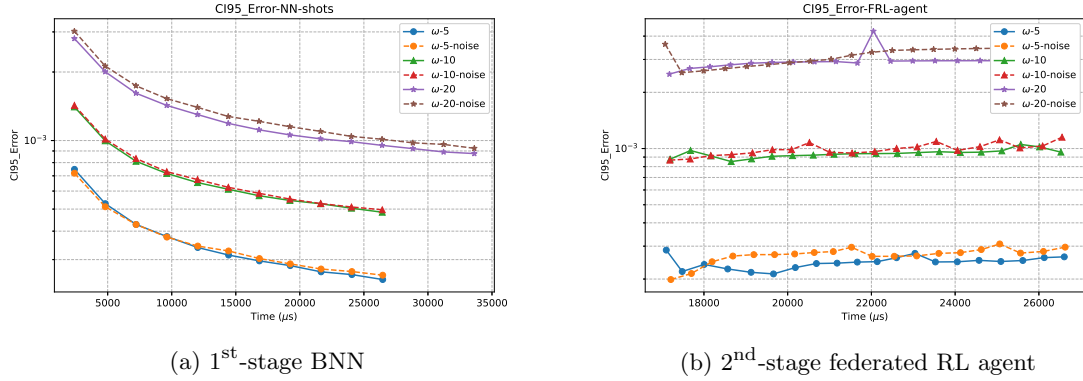
$$\varepsilon_{\omega}'^2 \geq \frac{1}{I'_{\text{a}} + \frac{12}{\omega_{\text{max}}^2} + I'_{\text{na}}}. \quad (49)$$

4 Robustness Analysis of the Protocol

For the realistic noisy scenario, we model asymmetric readout fidelities as $F_1 = 0.993$ and $F_0 = C + (1 - F_1)$, where the experimentally achievable contrast is set to $C = 0.99$. These values reflect effective readout performance attainable in room-temperature NV-center experiments with 50,000 repetitions, rather than single-shot readout fidelity, and therefore characterize an averaged-readout regime in our robustness analysis [4].

We simulate 1024 independent test trajectories for each configuration over 200 iterations, yielding 1024×200 samples per condition. The statistical error is quantified using the 95% confidence interval (CI95), computed as: $\text{CI95} = \bar{x} \pm t_{0.975, d-1} \cdot \frac{s}{\sqrt{d}}$, where $\bar{x}$ is the sample mean, s is the sample standard deviation, and $t_{0.975, d-1}$ is the critical value from the t-distribution with $d - 1$ degrees of freedom.

Below, we summarize the CI95 errors for both noise-free and noisy conditions under various $\omega_{\max}$ values, including the 1st-stage BNN and 2nd-stage adaptive estimation performance. Supplementary Figure 2 shows the statistical error of BNN and federated RL agent under testing by $1,024 \times 200$ samples. Supplementary Figure 2(a) shows the statistical error of BNN after 10–110 measurement shots under $\omega \in (0, 5)$ MHz and $\omega \in (0, 10)$ MHz, and after 10–140 shots under $\omega \in (0, 20)$ MHz. Supplementary Figure 2(b) shows the statistical error of the federated RL agent when the total time budget is 10 ms. To specify the statistical error of BNN after the 1st-stage (i.e., after 70 shots in this work), the table lists CI95 values. The CI95 for the 2nd-stage is the mean CI95 over the whole measurement time budget.

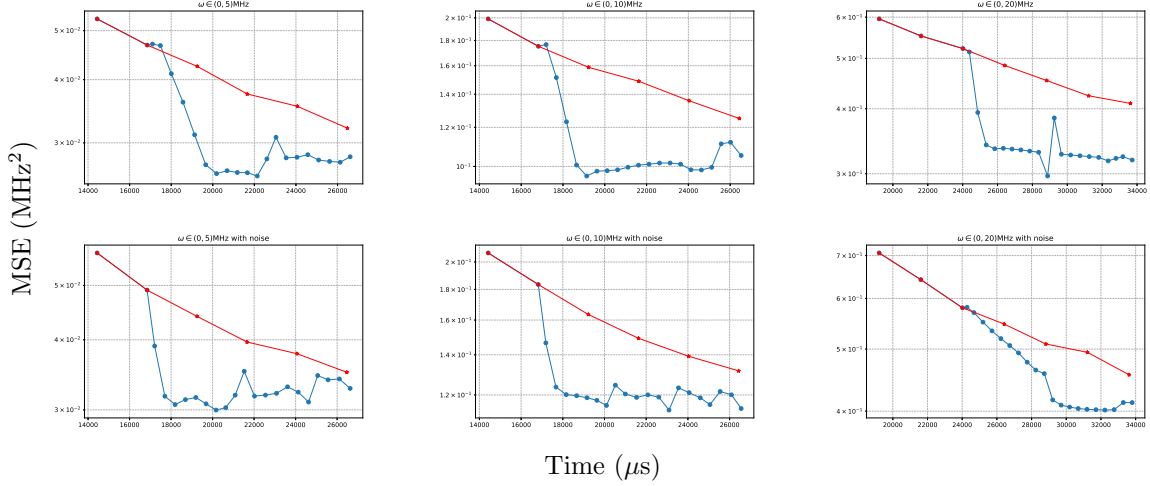


Supplementary Figure 2: Statistical error of results for the 1st-stage BNN and 2nd-stage federated RL agent.

Supplementary Table 1: CI95 results for different ranges of ω .

Ranges of ω (MHz)	CI ₉₅ (with noise / w/o noise)	bandwidth-normalized CI ₉₅
1st Stage: $\omega_{\max} = 5$	0.0296% / 0.0303%	$\sim 0.0059\%$ / 0.0061%
1st Stage: $\omega_{\max} = 10$	0.0572% / 0.0587%	$\sim 0.0057\%$ / 0.0059%
1st Stage: $\omega_{\max} = 20$	0.1115% / 0.1215%	$\sim 0.0056\%$ / 0.0061%
1st Stage (100 shots): $\omega_{\max} = 20$	0.0989% / 0.1043%	$\sim 0.0049\%$ / 0.0052%
2nd Stage: $\omega_{\max} = 5$	0.02458% / 0.02695%	0.00492% / 0.00539%
2nd Stage: $\omega_{\max} = 10$	0.09409% / 0.09934%	0.00941% / 0.00993%
2nd Stage: $\omega_{\max} = 20$	0.29998% / 0.31687%	0.01500% / 0.01584%

In Supplementary Figure 3, the red curve represents the NN-shots approach, while the blue curve corresponds to the performance of the proposed method. The 1st Stage concludes after 70 measurement shots for the cases $\omega \in (0, 5)$ MHz and $\omega \in (0, 10)$ MHz. For the case $\omega \in (0, 20)$ MHz, the 1st Stage ends after 100 measurement shots, and the time budget for the 2nd Stage is set to 10 ms. As shown in Supplementary Figure 3, under both noise-free and noisy conditions, the proposed scheme consistently outperforms the NN-shots strategy within the given resource constraints. Moreover, as shown in Supplementary Table 1, the CI₉₅ intervals are consistently tight, even in noisy settings and



Supplementary Figure 3: Comparison of the proposed scheme with NN-shots under noisy and noise-free conditions.

across a wide range of $\omega_{\max}$. Therefore, the results in Supplementary Table 1 and Supplementary Figure 3 collectively demonstrate the stability and reliability of the simulation results and support the practical feasibility of the proposed protocol in realistic experimental environments.

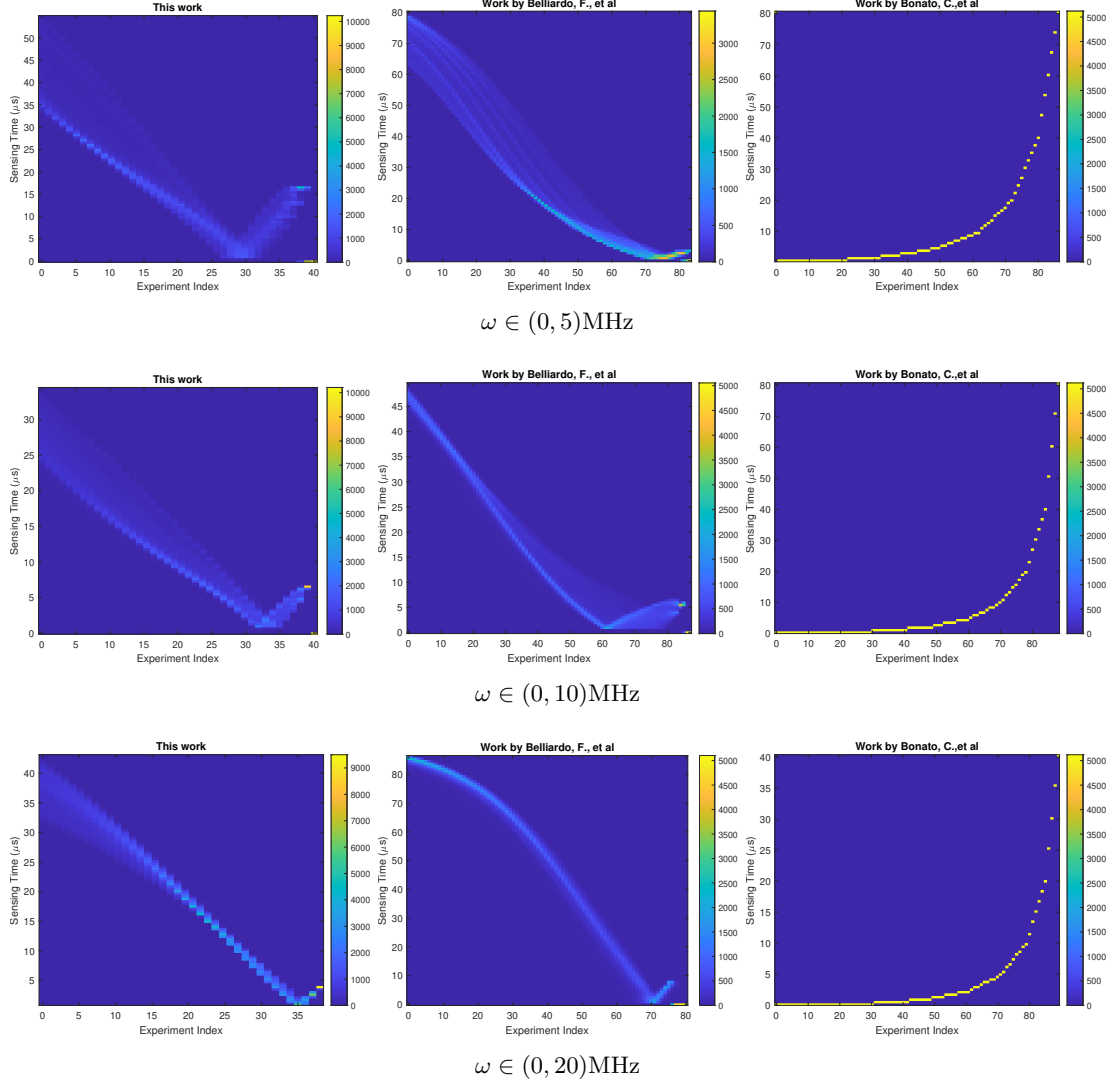
5 Runtime Evolution of Sensing Parameters

Supplementary Figure 4 and 5 show the frequency distributions of the optimized sensing parameters that correspond to the estimation results presented in Figure 2 and 3 of the main document. In Supplementary Figure 4, the optimized sensing time of this work’s protocol exhibits a trend similar to that reported by Belliardo et al. [6]. In both cases, the sensing time initially decreases as time resources are dissipated, and then slightly increases near the end of the estimation process as the remaining resources become limited. Nevertheless, the advantage of our protocol is the faster convergence speed as evidenced by the steeper slope and a smaller number of experiments (40 in ours while > 90 in others). This is because the search space of our algorithm is comparably smaller which is in the range of 0-40 μs , thanks to the narrowed range from the output of the protocol’s 1st Stage. It is also worth to mention that the Bonato baseline follows an exponential function to select the sensing time τ . Most of the experiments choose a very small τ while large values of τ ’s are selected very sparsely. This effect is very similar to the method by Nolan et al. [7] in which τ is fixed as $\tau_{\min}$. This unfortunately results in a significant loss of exploration in the search space. In addition to the sensing time optimization, Supplementary Figure 5 shows the distribution of control phase φ during the runs of the three algorithms. It can be observed that the evolution of φ has a stark difference among the three protocols, but they all exhibit convergence toward the final stages of the estimation sequence, as evidenced by the bright point on the far right of each figure. The results in Supplementary Figure 5 further confirm that our protocol converges faster.

To further compare the performance of the two adaptive baseline implementations based on the methods of Belliardo et al. [6] and Bonato et al. [4], we examine the performance of their models in terms of training loss measured by the estimation MSE. Note that the swarm optimization method by Bonato et al. [4] initially used Holevo variance to dictate iterative search of the sensing parameter φ . To make a fair comparison, we implement Bonato baseline using the same RL framework as the one in Belliardo et al. [6]. The changes include but are not limited to using particle filter to approximate the Bayesian posterior probability and using MSE as the loss function.

We use a modified implementation based on the protocol of Bonato et al. [4], reproducing its multi-time protocol, including the bit-by-bit variation of the sensing time. We set the repetition parameters to $N = 8$, $G = 5$, and $F = 2$, for which the complete protocol fits within the stated 22 ms experimental-time budget. We refer to this implementation below as the Bonato baseline.

Some exemplary simulation setups are as follows: the number of particles was fixed at $P = 960$, and the number of training iterations is set to 16,384. Supplementary Figure 6 plots the training loss curves of the two methods. This result can explain the performance of the two methods in Figure

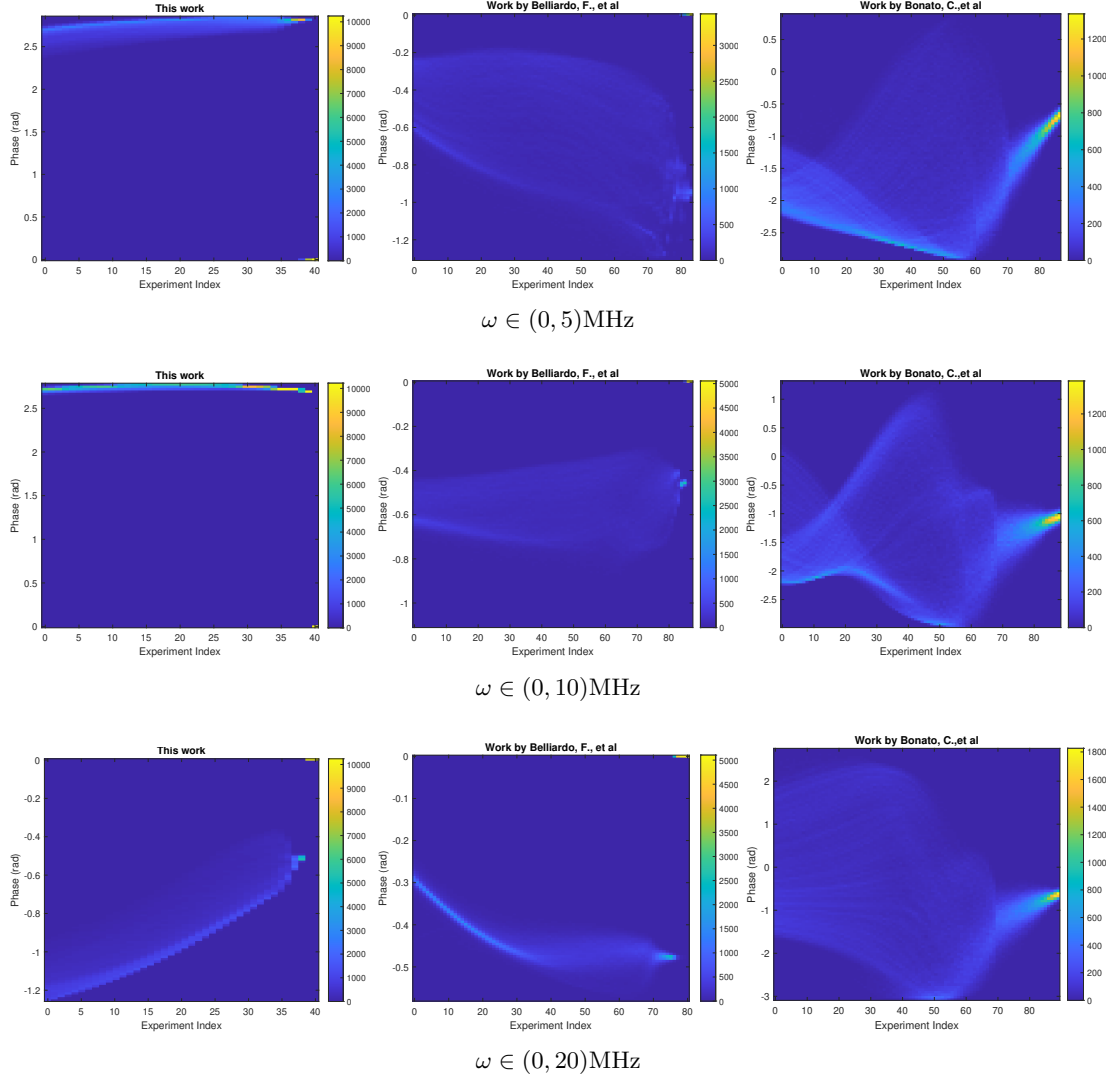


Supplementary Figure 4: Distribution of optimized sensing time for different protocols

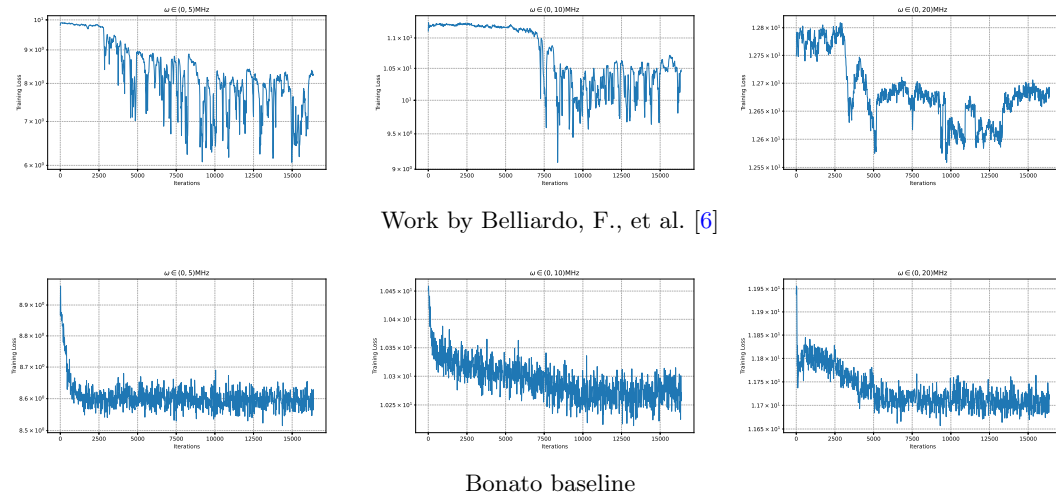
2 and 3 of the main document. Because Bonato baseline only optimizes φ , its model convergence is much better than Belliardo's method, particularly when $\omega \in (0, 20)\text{MHz}$ for wide-range sensing. Their training losses in three ω settings are rather close, with Belliardo's method having a minor edge for small ω 's.

6 Optimization of Δ

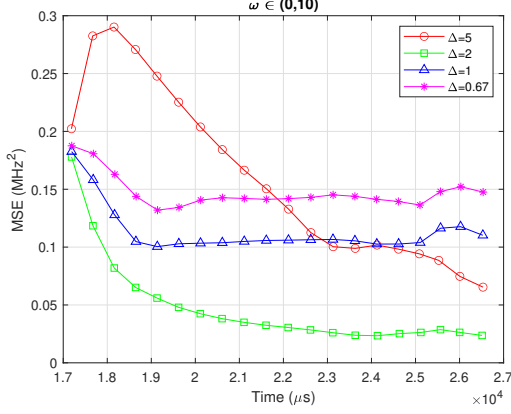
The following simulation results to illustrate how performance is affected by varying Δ .



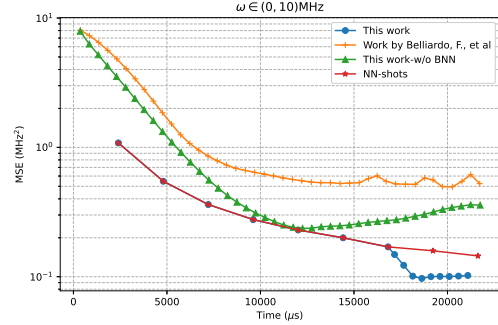
Supplementary Figure 5: Distribution of optimized control phase for different protocols



Supplementary Figure 6: Training loss of the RL agent in the two baseline protocols



(a) $\Delta < \omega_{\max}$



(b) $\Delta = \omega_{\max}$

Supplementary Figure 7: MSE performance with respect to different choices of Δ .

Supplementary Figure 7 illustrates the sensitivity of performance with respect to the choice of Δ . To isolate the effect of Δ and avoid confounding due to changes in particle resolution (Δ/P), we scale the number of particles P proportionally with Δ : for $\Delta = 5$, we set $P = 240 \times 4$; for $\Delta = 2$, $P = 240 \times 2$; and for $\Delta = 1$ and $\Delta = 0.67$, $P = 240$. This ensures a consistent particle density across the tested ranges, allowing a fair comparison of the impact of Δ on estimation performance.

Supplementary Figure 7(a) presents results for the field range $\omega \in (0, 10)$ MHz, with a total time budget of $22,000 \mu\text{s}$ and a non-adaptive first stage comprising 70 Ramsey measurements. The BNN estimation error is approximately $\varepsilon = 0.41$ MHz. The case with $\Delta = 0.67$ MHz (purple curve) violates the lower bound 2ε , and its performance degrades significantly. In contrast, $\Delta = 1$ and 2 MHz satisfy the criterion and lead to better results. A larger Δ increases the likelihood that the true ω lies within the adaptive range, improving the effectiveness of both Bayesian updates and RL exploration. Supplementary Figure 7(b) illustrates performance when $\Delta = \omega_{\max}$, effectively bypassing the first-stage BNN and relying solely on RL-based inference (“This work w/o BNN”). In this regime, large Δ values (e.g., 5 and 10 MHz) yield broad posterior distributions and inefficient exploration, resulting in slower convergence and reduced estimation precision.

This sensitivity analysis confirms the critical role of Δ : if too small (violating $2\varepsilon \leq \Delta$), it risks excluding the true parameter; if too large (approaching $\omega_{\max}$), it dilutes the efficiency of exploration. These findings validate our empirical choices and underscore the trade-off: Δ must be wide enough to include the true ω with high probability, yet narrow enough to guide focused and efficient learning.

References

- [1] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
- [2] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., et al.: Jax: composable transformations of python+ numpy programs (2018)
- [3] Fiderer, L.J., Schuff, J., Braun, D.: Neural-network heuristics for adaptive bayesian quantum estimation. *Prx Quantum* **2**(2), 020303 (2021)
- [4] Bonato, C., Blok, M.S., Dinani, H.T., Berry, D.W., Markham, M.L., Twitchen, D.J., Hanson, R.: Optimized quantum sensing with a single electron spin using real-time adaptive measurements. *Nature nanotechnology* **11**(3), 247–252 (2016)
- [5] Sun, Y., Lin, Z., Mao, Y., Jin, S., Zhang, J.: Channel and gradient-importance aware device scheduling for over-the-air federated learning. *IEEE Transactions on Wireless Communications* **23**(7), 6905–6920 (2023)
- [6] Belliardo, F., Zoratti, F., Marquardt, F., Giovannetti, V.: Model-aware reinforcement learning for high-performance bayesian experimental design in quantum metrology. *Quantum* **8**, 1555 (2024)
- [7] Nolan, S., Smerzi, A., Pezzè, L.: A machine learning approach to bayesian parameter estimation. *npj Quantum Information* **7**(1), 169 (2021)