ACTION-FACTORED MULTI-AGENT REINFORCEMENT LEARNING FOR SCALABLE QUANTUM DEVICE TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Cooperative multi-agent reinforcement learning is well suited to problems with large parameter spaces and exploitable local structure, such as the tuning of electrostatically-defined quantum-dot arrays. However, if parameter cross-talk is strong, a non-stationary environment from the perspective of any individual agent can destabilize learning – the same effect that plagues manual tuning of such systems. We propose using a factored representation of the action space, learned online, to decouple agents and minimize their interference. Our framework, QADAPT, uses this factorization to efficiently learn shared policies based on local measurements and rewards. With this modular strategy, we achieve zero-shot generalization to unseen quantum device sizes and maintain an approximately constant number of convergence steps to reach target regimes. This work provides a scalable route toward the rapid calibration of large-scale quantum processors.

1 INTRODUCTION

Reinforcement learning (RL) learns policies for sequential decision-making under uncertainty through environmental interaction (Sutton & Barto, 2018). Deep RL has achieved human-level performance on complex, high-dimensional tasks (Mnih et al., 2015) and efficient continuous control from sensory inputs (Lillicrap et al., 2020), supporting robotics, locomotion, and quantum computing applications: error-robust quantum gate design (Baum et al., 2021), analog pulse shaping (Semola et al., 2022), real-time qubit feedback (Reuer et al., 2023; Vaidhyanathan et al., 2026), and quantum-dot voltage tuning (Nguyen et al., 2021). As RL and machine learning become central to quantum control (Alexeev et al., 2025; Cao et al., 2026), efficient scaling to larger devices remains challenging (Ares, 2021; Zwolak et al., 2024; Kondo et al., 2025; Vaidhyanathan et al., 2025; Schorling et al., 2025; Bukov & Marquardt, 2026).

Semiconductor spin qubits in particular, while a compelling quantum computing platform (Chatterjee et al., 2021; Bartee et al., 2025; Steinacker et al., 2025; Dijkema et al., 2026), face enormous control complexity due to measurement noise, device non-uniformities, and cross-talk. The tuning procedure involves calibrating dc gate voltages to create confined islands of charge, i.e. quantum dots, as shown in Fig. 1a. There exist two main gate types; *plunger gates* predominantly control the electrochemical potential of quantum dots, hence their charge occupations, and *barrier gates* modulate the tunnel coupling between dots. Because of dense gate electrode arrangements, adjusting one gate voltage perturbs neighboring dots via capacitive cross-talk (Borsoi et al., 2024). Measuring the system's response as a function of two gate voltages produces images known as *charge stability diagrams* (CSDs). These diagrams contain relevant tuning and cross-talk information, which may be inferred from the position and curvature of the line features they contain. As shown in Fig. 1b, the objective is to reach a voltage configuration that brings all charge occupations and tunnel-couplings to a predefined target.

Even with existing autotuning approaches – including fully autonomous spin qubit tuning (Schuff et al., 2026) and a broad class of quantum dot tuning methods (Moon et al., 2020; Zwolak et al., 2020; Roux et al., 2025; Carlsson et al., 2025) – demonstrations beyond two-dot (sub)systems are lacking. Sequential tuning of smaller sub-systems offers no guarantee of global optimality, and designing meaningful score functions for supervised or Bayesian algorithms becomes increasingly non-trivial as parameter spaces grow and tuning landscapes become highly non-convex. Since data acquisition remains the primary time bottleneck in practice, stricter sample efficiency is also demanded.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We present QADAPT (Quantum Arrays of Dots in Automated Parallel Tuning), a multi-agent reinforcement learning (MARL) framework for scalable quantum device tuning. By learning a factored voltage representation online, QADAPT enables decoupled control over each dot's electrochemical potential. This supports *centralized training with decentralized execution* (CTDE), where agents act independently on local observations and share policies according to physical roles. The resulting modular architecture mitigates the combinatorial action-space growth, enables parallelization, and allows for zero-shot generalization through policy inheritance by new agents.

Our core contributions are:

- **Adaptive Action-Space Factorization:** A Kalman-guided pipeline estimates the local gate-to-dot capacitance matrix online to construct a virtual action basis. This structure-aware reparameterization reduces cross-agent interference and improves sample efficiency.
- **Role-Shared Decentralized MARL:** We formulate tuning as a cooperative continuous Decentralized Partially Observable Markov Decision Process (Dec-POMDP), introducing modular actor-critic networks with gate-type-specific parameter sharing (Gupta et al., 2017). This decomposes the high-dimensional global search while keeping the number of learned policies independent of array size.
- **Zero-Shot Scalability and Empirical Evaluation:** QADAPT generalizes without retraining across system sizes, unseen device configurations, and various non-idealized physical effects, outperforming state-of-the-art baselines across all evaluated benchmarks.

2 Related Work

Automatic tuning of semiconductor quantum dots spans simple physics-inspired heuristics (Baart et al., 2016; Yon et al., 2025) to machine-learning-assisted pipelines (Schuff et al., 2026; Carlsson et al., 2025). Recent works have introduced various classifiers to enable the autonomous tuning of quantum dots (Kalantre et al., 2019; Ziegler et al., 2023; Van Straaten et al., 2025), reach target charge occupations (Durrer et al., 2020; Czischek et al., 2021), and calibrate tunnel couplings (van Esbroeck et al., 2020). In conjunction with Bayesian optimization, classifier methods have also proven successful across multiple device architectures (Severin et al., 2024), yet, their validation is restricted beyond double-dot systems.

The standard strategy for larger systems relies on sequential tuning: capacitive crosstalk is probed iteratively and quantum dots are tuned using virtual voltage controls via an "n+1" procedure (Volk et al., 2019). Using CSDs as input, the underlying feature-tracking task for gate virtualization has been offloaded to vision models, including Convolutional Neural Networks (CNNs) (Tsuzuki et al., 2026), U-Net-like architectures (Rao et al., 2025; Muto et al., 2025) and Transformers (Hader et al., 2025; Marchand et al., 2025), but requires additional pre-processing for larger systems (Oakes et al., 2020). In sequential tuning, devices are also prone to biasing into locally compatible rather than globally optimal voltage configurations. Furthermore, post-hoc and fixed cross-capacitance estimates greatly limit practical voltage tuning windows.

Decentralized policies mitigate combinatorial joint-action-space growth, but ambiguous credit assignment and conflicting local objectives complicate coordination when cross-talk is present (Oroojlooy & Hajinezhad, 2023). Independent Q-learning (IQL) (Tan et al., 1993) treats other agents as part of the environment and generally lacks convergence guarantees because simultaneous policy updates induce non-stationarity. We group approaches addressing these challenges into five categories:

Value function factorization decomposes joint action-values into per-agent utilities for decentralized execution with shared rewards. VDN uses additive decomposition, QMIX monotonic mixing, and QTRAN unconstrained surrogate factorization (Sunehag et al., 2017; Rashid et al., 2020; Son et al., 2019). COMIX and FacMADDPG extend factorization to continuous control using cross-entropy optimization and factored critics, respectively (de Witt et al., 2020).

Centralized critics use global information in per-agent value estimates. COMA (Foerster et al., 2018) addresses credit assignment with counterfactual baselines yielding agent-specific advantages; MADDPG (Lowe et al., 2017) conditions each critic on all agents' observations and actions. FACMAC (Peng et al., 2021) combines QMIX's factored critics with MADDPG's centralized policy gradients.

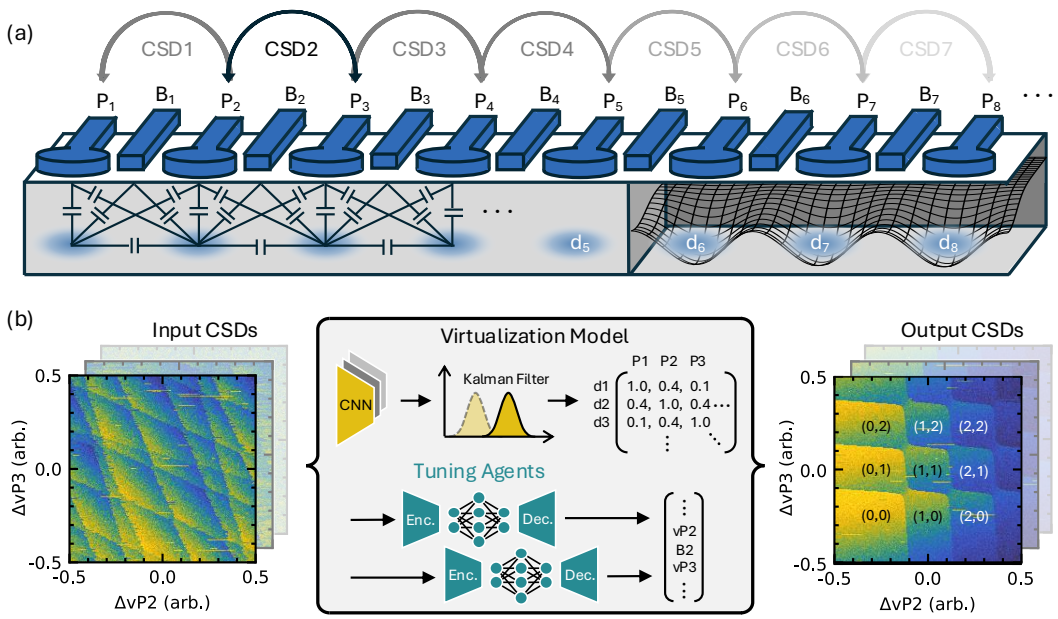


Figure 1: **Algorithmic Flow of QADAPT (a)** Barrier gates ($\mathrm{B_x}$) interdigitate plunger gates ($\mathrm{P_x}$), and primarily control dot-dot tunnel couplings and charge occupations, respectively. All gates contribute in defining the confinement potential necessary to form quantum dots ($\mathrm{d_1 ... d_D}$). Gate cross-talk can be approximated by a capacitive network, which in general extends beyond nearest neighbors, greatly complicating array tuning. Charge stability diagrams (CSDs) are acquired by sweeping the voltage on neighboring pairs of plunger gates and bear information on the array's current tuning state. **(b)** Each CSD is passed through a virtualization model and tuning agent model. The virtualization model returns relative cross-talk estimates between $\mathrm{P_x}$ and dots $\mathrm{d_{x\pm1}}, \mathrm{d_{x\pm2}}$, refined by a Kalman filter at each tuning step, resulting in a cross-capacitance matrix that enables decoupled plunger control. Tuning agents operate in the decoupled plunger gate basis, outputting decentralized actions in the form of new virtual voltage setpoints denoted by $\{\mathrm{vP_x}, \mathrm{B_x}\}$. Through collective optimization of local objectives, a global target configuration is reached with a desired level of tunnel-coupling between dots, and charge occupation across all dots in the array (right).

Learning to communicate enables agents to exchange information to improve coordination, either through graph-structured policies or learned message-passing protocols (Sukhbaatar et al., 2016). Notably, GPG (Khan et al., 2020) introduced parametrized swarm policies over a proximity graph, InforMARL (Nayak et al., 2023) proposed graph attention over local neighborhoods, and MADDPG-M (Kilinc & Montana, 2018) learns explicit communication channels alongside task policies.

Consensus-based methods coordinate agents through shared latent representations formed from joint observations, as in CMAT's auto-regressive Transformer decoder (Zhao et al., 2026; Wen et al., 2022), or local observations, for example through contrastive learning (Feng et al., 2024).

Parameter sharing and specialization shares policies across similar agent roles, reducing independently trained networks and improving sample efficiency (Christianos et al., 2021). IPPO and MAPPO (De Witt et al., 2020; Yu et al., 2022) demonstrate strong cooperative performance with shared PPO policies; HAPPO (Kuba et al., 2021; Zhong et al., 2024) extends this to heterogeneous agents. HyperMARL (Tessera et al., 2024) mitigates gradient interference through agent-identity-conditioned hypernetworks.

QADAPT instead learns a *factored action space representation* to decouple agents, transforming joint control into an approximately separable sum of local objectives from which cooperative outcomes follow (Appendix A).

3 METHODS

3.1 PROBLEM FORMULATION AND SIMULATION SETTING

We simulate the quantum dot tuning environment using QArray (van Straaten et al., 2024b;a), acquiring CSDs by sweeping the voltage on pairs of nearest neighbor plunger gates. The array is considered optimally tuned when all inter-dot tunnel couplings meet a pre-defined target value, and each dot contains exactly one charge, as in Fig. 1b (right). These conditions are satisfied over a voltage manifold that constitutes a small (<1%) fraction (Schuff et al., 2026) of the total voltage hyper-volume, the center of which we denote as the target voltage configuration $\mathbf{v}^*$. We note that orthogonal pairs of transition lines in each CSD are an indicator of accurate cross-talk compensation, however, they are not a pre-requisite for convergence.

We formulate this tuning process as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) utilizing $2N-1$ agents (Bernstein et al., 2002). Each agent represents one device gate across all plungers and barriers in the device.

- **State & Observations:** The system state $s^t = (\mathbf{v}_p^t, \mathbf{v}_b^t) \in \mathbb{R}^{2N-1}$ is the vector of all plunger and barrier gate voltages. At time step t, agent i receives a local observation o_i^t consisting of its current normalized voltage $v_i^t \in [-1, 1]$ concatenated with a CNN representation of its local CSD scans.
- **Actions:** Each agent outputs a normalized action $\hat{a}_i \in [-1, 1]$, corresponding to an absolute voltage assignment $a_i^t = v_i^{t+1}$. The transition dynamics over the joint action space $\mathbf{a^t}$ are fully deterministic, given by $P(s^{t+1}|s^t, \mathbf{a}_t) = \delta(s^t - \mathbf{a^t})$.
- **Rewards:** The *lever arm* is a device-specific factor that relates a change in gate voltage to the resulting shift in a dot's electrochemical potential. Identical voltage changes can therefore induce different physical responses. We apply reward shaping (Hu et al., 2020) to define a normalized per-agent reward $r_i^t \in [0, 1]$, rescaled according to the agent's hidden lever arm α_i and the voltage tuning range. The agents' goal is to maximize the cumulative return $J = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T}\sum_{i=1}^{N} r_i^t\right]$, which in our cooperative setting is equivalent to optimizing a common global objective (Rashid et al., 2020).

3.2 ADAPTIVE GATE VIRTUALIZATION

To mitigate action interference caused by capacitive cross-talk, QADAPT learns a factored representation of the plunger action space online. At each time step, CSDs are passed through a lightweight CNN, $f_{\text{virt}}(\cdot)$, to estimate the capacitances between each plunger gate and its surrounding dots (up to two neighbors away).

A Kalman filter incrementally updates these estimates, treating the unknown couplings as latent state variables, and uses new measurements as observations for Bayesian updates (Chen et al., 2003; Fang et al., 2018). See Fig. 2 for an example. This process yields a time-dependent $D \times D$ cross-capacitance matrix Φ_t, which maps physical plunger voltages $\mathbf{v}_p$ to virtual voltages $\mathbf{u}_p = \Phi_t \mathbf{v}_p$.

In this space, the new control variables u_i act approximately independently by diagonalizing the local coupling between gates and dots. Let $\mathbf{e}_t$ denote the local device-state error and let C_t be the local response Jacobian. Under the virtual action update $\Delta\mathbf{v}_p^t = \Phi_t^{-1}\Delta\mathbf{u}_p^t$, the one-step quadratic model becomes:

$$\frac{1}{2}\|\mathbf{e}_t + C_t\Phi_t^{-1}\Delta\mathbf{u}_p^t\|_2^2. \tag{1}$$

When $C_t\Phi_t^{-1} \approx I$, this objective has small mixed action terms and is close to the separable surrogate $\frac{1}{2}\sum_i (e_{t,i} + \Delta u_{p,i}^t)^2$. This transforms the joint control problem into an approximately separable sum of local objectives, preconditioning the local quadratic control landscape (Appendix A).

3.3 MODULAR ACTOR-CRITIC ARCHITECTURE

To process observations and output actions, we employ a modular advantage actor-critic architecture (Konda & Tsitsiklis, 1999). To reduce model complexity and enable zero-shot scalability to new

arrays (Neumann & Gros, 2022), parameters $\{\theta, \phi\}$ are shared across all agents of the same physical type (Christianos et al., 2021) (i.e., one parameter set for all plunger gates, and another for all barrier gates).

The network relies on a convolutional encoder $f_{\mathrm{CNN}}(\cdot)$ to extract spatial hierarchies from the CSDs in which gate i participates, denoted as $\mathbf{X}_i^t$. This feature representation is concatenated with a learned linear projection $p(\cdot)$ of the gate's current voltage, such that $o_i^t = f_{\mathrm{CNN}}(\mathbf{X}_i^t)||p(v_i^t)$. For plungers at the edge of the array that participate in only one CSD, the single available scan is duplicated across channels. For inner plungers, the two participating CSDs are stacked along the channel dimension. Barrier gates participate only in CSDs acquired by sweeping their two immediate neighboring plungers, because barrier-induced cross-talk on next-nearest dots is near zero (Rao et al., 2025). The network then branches:

- **Actor Head:** Outputs a probability distribution over continuous voltage adjustments via a Gaussian policy $\pi_\theta(a_i^t \mid o_i^t)$.
- **Critic Head:** Produces an estimate of the value function $V_\phi(o_i^t)$ based strictly on local observations, preserving decentralized execution.

3.4 Training Pipeline

We train QADAPT on a 4-dot system using a modified Proximal Policy Optimization (PPO) method for stable and sample-efficient learning in continuous control tasks (Schulman et al., 2017). At each training iteration, we roll out multiple parallel episodes using the current policy π_θ. Because of our parameter-sharing scheme, experiences from all agents of the same type are aggregated into a single batch. This effectively multiplies our batch size by the number of agents, significantly stabilizing updates.

Because the optimal policy at any given step is always to move the system closer to the target configuration, future rewards offer little additional guidance (Bouneffouf et al., 2020). Therefore, we model the task as a contextual bandit, eliminating temporal credit assignment (discount factor $\gamma = 0$). Each step is treated independently with dense, immediate feedback, which simplifies the learning problem and handles any residual non-stationarity of the multi-agent environment (supported by our ablation in Table 1).

With $\gamma = 0$, there is no bootstrapping or reward-to-go: generalized advantage estimation reduces to the immediate-reward residual below, and PPO's clipped surrogate serves update stability rather than temporal credit assignment. Nevertheless, training contexts remain on-policy, because sequential rollouts visit states induced by the policy's own actions rather than drawing contexts independently.

After collecting a batch of trajectories, per-agent advantage estimates $\hat{A}_i^t$ are computed, which, without temporal credit assignment, reduces to immediate reward residuals:

$$\hat{A}_i^t = r_i^t - V_\phi(o_i^t) \tag{2}$$

To update the shared policy parameters θ, local advantages are aggregated and the joint PPO clipped surrogate objective is maximized across all agents i and batch time steps t:

$$\mathcal{L}^{\mathrm{CLIP}}(\theta) = \hat{\mathbb{E}}_{t,i}\left[\min\left(\kappa_i^t(\theta)\hat{A}_i^t, \mathrm{clip}(\kappa_i^t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i^t\right)\right] \tag{3}$$

where $\kappa_i^t(\theta) = \frac{\pi_\theta(a_i^t|o_i^t)}{\pi_{\theta_{\mathrm{old}}}(a_i^t|o_i^t)}$ is the probability ratio.

Concurrently, the shared critic parameters ϕ are updated by minimizing the aggregated mean squared error between the predicted values and the actual returns (which here equal the immediate rewards r_i^t):

$$\mathcal{L}^{\mathrm{VF}}(\phi) = \hat{\mathbb{E}}_{t,i}\left[\left(V_\phi(o_i^t) - r_i^t\right)^2\right] \tag{4}$$

By computing these expectations over the joint experience buffer of all homogeneous agents, gradients become averaged across the array. This decentralized PPO algorithm is thus an instance of centralized training with decentralized execution (Xu et al., 2018).

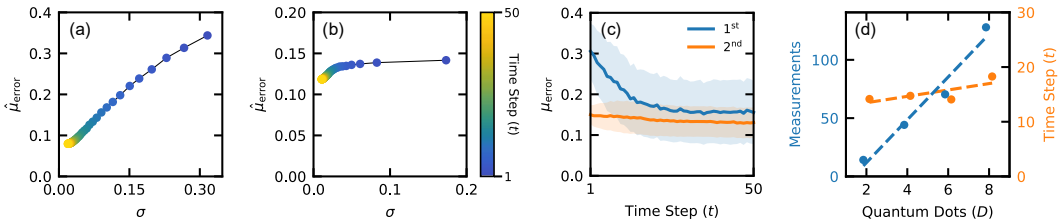


Figure 2: **Gate Virtualization and Scaling Performance of QADAPT. a)** Kalman posterior uncertainty (standard deviation) σ versus absolute estimation error $\hat{\mu}_{\text{error}}$ for nearest-neighbor plunger gate-to-dot cross-capacitances (ground truth: 0.7). Each marker represents one time step in a 4-dot system (3 CSD measurements), averaged over 1000 episodes. The descending trajectory shows improving estimates tracked by decreasing uncertainty. **b)** As in a), for second-nearest neighbors (ground truth: 0.3); their weaker cross-talk produces a weaker but stable trend. **c)** Raw prediction errors of $f_{\text{virt}}(\cdot)$ over time for first- and second-nearest couplings. **d)** Zero-shot scaling of CSD acquisitions and tuning steps to convergence in 2, 4, 6, and 8 dot systems.

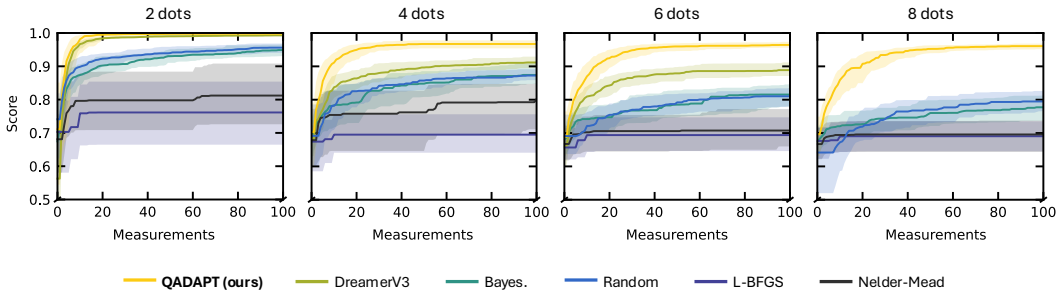


Figure 3: **Domain-specific Benchmarks with Array Size.** From left to right, arrays increase from 2 to 8 dots. QADAPT maintains rapid convergence against DreamerV3, Gaussian-process Bayesian optimization, Nelder-Mead, L-BFGS, and random search. Scores use the cumulative minimum normalized distance to target voltages across all gates at each scan step. Lines average 100 runs; envelopes show half a standard deviation. Initial and target voltages are randomized per run. DreamerV3 is omitted for 8 dots because of compute constraints.

4 Experiments

To highlight the benefits of our deep multi-agent approach, we evaluate QADAPT against several strong baseline methods (Fig. 3). To represent approaches commonly found in the literature of low-dimensional quantum dot tuning, we first consider several gradient-free optimizers: random search, Nelder–Mead (Nelder & Mead, 1965), Bayesian optimization with Gaussian processes (Rasmussen & Williams, 2006; Shahriari et al., 2016), and L-BFGS (Nocedal, 1980) using finite-difference gradient estimates. To represent model-based reinforcement learning, we employ DreamerV3 (Hafner et al., 2024), adapting its latent world model to plan actions for the tuning task. This comparison probes whether explicit dynamics modeling improves sample efficiency in our measurement-constrained regime.

To identify components responsible for QADAPT's performance, we conduct several further baseline comparisons and ablations (Table 1). Firstly, we compare $f_{\text{CNN}}(\cdot)$ of our actor-critic architecture with three alternative backbones: (i) a heavier ResNet-style encoder ("Nature CNN") (He et al., 2016; Tsuzuki et al., 2026); (ii) an LSTM-based recurrent policy (Hochreiter & Schmidhuber, 1997); and (iii) a vanilla Transformer (Vaswani et al., 2017). To isolate the contribution of our action space factorization, we ablate the virtualization model in addition to a variant where only the Kalman filtering is disabled. Without virtualization, QADAPT becomes identical to IPPO (De Witt et al.,

2020). We compare this against MAPPO (Yu et al., 2022) (the critic receives a concatenation of all agents' observations), allowing us to discern whether giving the critic access to global information during training can overcome the absence of virtualization to coordinate agents. We also include MADDPG (Lowe et al., 2017) and FACMAC (Peng et al., 2021) as off-policy baselines, probing the benefits of past experiences (in the form of replay buffers) toward agent coordination. Finally, we evaluate the contribution of QADAPT's multi-agent framework using single-agent PPO policies that concatenate all agents' observations and jointly control all gates. We compare a policy acting in the raw gate-voltage basis with one using QADAPT's online virtualization, isolating whether action factorization alone accounts for the benefits of decentralized control. In all comparisons, we report both convergence and steps-to-convergence to disentangle reliability from efficiency.

To separate the immediate-reward objective from sequential data collection, we also train a strict contextual-bandit baseline with the same architecture, reward, $\gamma = 0$ objective, and experience budget as QADAPT. This variant uses single-step episodes, taking one action from each independently sampled initial device state, so its training contexts are exogenous. It is evaluated using the same sequential tuning protocol as the other entries in Table 1.

Table 1: **Ablation Study and MARL-specific Baseline Comparisons.** Each row summarizes 100 tuning trials on a 4-dot array. Convergence requires all agents to lie within 2%, 5%, or 10% of their total tuning range from target voltages (Appendix B). We report the percentage converging within 100 time steps and mean steps μ_{steps} over successful trials only. Training uses early stopping; epochs are reported per variant.

Variant	Convergence Rate (%) ↑	μ_{steps} ↓	Epochs
QADAPT (ours)	**93 / 95 / 95**	**20.5 / 16.7 / 13.2**	150
w/ Nature CNN (He et al., 2016; Tsuzuki et al., 2026)	91 / 98 / 100	20.1 / 17.8 / 14.6	250
w/ LSTM (Hochreiter & Schmidhuber, 1997)	73 / 85 / 94	22.1 / 17.3 / 12.7	275
w/ Transformer (Vaswani et al., 2017)	3 / 26 / 61	33.7 / 29.2 / 27.1	150
w/o Kalman filter	10 / 55 / 89	26.9 / 24.2 / 18.1	100
$\gamma = 0.9$	89 / 94 / 97	19.5 / 16.3 / 13.3	150
w/ single-step episodes	0 / 0 / 0	N/A / N/A / N/A	150
IPPO (De Witt et al., 2020)	63 / 84 / 98	28.0 / 22.5 / 15.2	275
MAPPO (Yu et al., 2022)	78 / 90 / 95	26.1 / 19.2 / 14.2	400
MADDPG (Lowe et al., 2017)	1 / 2 / 5	19.0 / 20.5 / 18.8	500k*
FACMAC (Peng et al., 2021)	0 / 0 / 1	N/A / N/A / 1	500k*
Single Agent PPO w/o virt.	0 / 1 / 6	N/A / 43.0 / 26.2	150
Single Agent PPO w/ virt.	2 / 18 / 71	33.5 / 50.8 / 11.8	150

*For off-policy methods, the number of environment steps is reported.

5 Results and Discussion

Across evaluated array sizes and randomized devices, QADAPT achieves strong convergence and measurement efficiency relative to the learning-based and domain-specific baselines in Section 4.

In Fig. 2(a-b), we first illustrate the performance of our virtualization model in the baseline 4-dot system. As tuning progresses, a trend emerges in which the mean error in predicted gate-to-dot cross capacitance decreases, along with the variance in those predictions. This correlation is expected, because the confidence in Φ – derived from the Kalman filter's uncertainty signal – should increase as prediction errors decrease (proof provided in Appendix A). A high density of predictions with small errors, occurring beyond $t \approx 20$, is comparable to QADAPT's mean steps to convergence (see Fig. 3a), indicating a constructive feedback loop between Φ and the MDP. Overall, this supports our claim that a factored action space improves agent coordination and therefore accelerates quantum dot array tuning.

Removing virtualization (IPPO, Table 1) reduces convergence and increases tuning steps, consistent with cross-talk turning local raw-voltage actions into global perturbations. Removing only the Kalman filter also harms performance: filtering regularizes abrupt capacitance changes that could destabilize tuning, particularly near targets where transition lines in CSDs are sparse. Negative predictions

indicate overcompensation by neighboring gates when gate i changes. We include such examples when training $f_{\text{virt}}(\cdot)$ so that Φ can self-correct online. Appendix G (Fig. 7) further quantifies how injected cross-capacitance measurement error and Kalman-filter process noise affect convergence rates and steps to convergence. The heat-map experiment shows that temporal filtering preserves high convergence rates, even across a broad range of observation errors.

In Fig. 3, we present the evolution of the tuning score with the number of simulated measurements (CSD acquisitions). Because data acquisition is typically the dominant wall-clock cost in experiments, we report performance as a function of the number of measured CSDs, rather than gradient steps or environment transitions. The score is defined as $1 - \sum_{i=1}^{N} \tilde{v}_i / N$, where $\tilde{v}_i$ is the 'best-so-far' distance to the target voltage v_i^*, normalized by the voltage bounds of the tuning environment. We observe the largest performance margins between QADAPT and traditional L-BFGS and Nelder-Mead optimizers, likely due to the non-convex optimization landscape, which causes the algorithms to become trapped in local minima. For this reason, we are unsurprised to find a random search to be more efficient, and showing higher mean scores. Using RL-based approaches avoids spending many evaluations exploring uninformative directions in the high-dimensional voltage space.

The policy trained on 4 dots transfers to arrays of 2–8 dots with only shallow degradation in convergence. Sharing parameters within barrier and plunger roles mitigates the joint action-space growth faced by centralized baselines such as DreamerV3. Figure 2d shows near-linear $\mathcal{O}(N)$ growth in measurements needed for a mean tuning score of 0.95, corresponding to nearly constant tuning steps. Bayesian optimization slows sharply beyond 4 dots (Fig. 3); its Gaussian-process surrogate adds cubic complexity in the number of observations, increasing overhead relative to other learning-based and gradient-free methods.

MADDPG and FACMAC achieve very low convergence rates under our measurement budget (Table 1). We observed actor gradient collapse in MADDPG and critic gradient explosion in FACMAC, suggesting that off-policy replay and centralized value estimation are not in themselves sufficient to stabilize learning when agents act in a strongly coupled action basis.

Without virtualization, single-agent PPO converges in $0/1/6\%$ of trials at the $2/5/10\%$ radii; adding online virtualization improves these rates to $2/18/71\%$, but they remain below QADAPT's $93/95/95\%$ (Table 1). Thus, virtualization benefits joint control, while the shared decentralized policies provide additional gains in convergence reliability. Unlike the fixed-size concatenated observation and action representation, QADAPT's local policies can be reused when the array size changes.

The strict contextual-bandit baseline trained on independently sampled single-step episodes achieves 0% convergence at all three radii (Table 1), despite its training returns plateauing within budget. It tends to move toward a fixed point near the centre of the voltage tuning range and remain there. This result supports the importance of sequential on-policy data collection in this setting: removing temporal credit assignment does not remove the need to learn from states reached through the policy's own actions.

Encoder ablations mainly trade computation for modest changes in steps-to-convergence. Once essential CSD features are captured, our lightweight encoder remains effective against larger attention-based or recurrent alternatives.

5.1 Scalability

QADAPT's modular policies permit agents to be added at test time, avoiding growing joint optimization problems and error accumulation from long-horizon imagined roll-outs. A policy trained on N dots deploys directly to N' dots (using $2N' - 1$ agents) without architectural changes or retraining. Local agents act concurrently during training and execution, while action factorization reduces cross-agent interference. Single-agent models instead face rapid growth in joint state-action complexity as control variables are added. For example, training DreamerV3 on 2/4/6 dot-arrays in Fig. 2 required 2/1/0.5 million rollouts.

Other MARL approaches also face transfer constraints. FACMAC's policy gradients and value-function semantics remain tied to the training agent count despite its factored critic and value mixing. MAPPO's centralized critic concatenates all agent observations, likewise tying its input to the training configuration. Even scalable graph-based methods such as InforMARL control individual entities:

here, each agent would control one voltage parameter Δv_p^t and would need to learn virtualized adjustments Δu_p^t implicitly. QADAPT offloads this transformation to the lightweight $f_{\text{virt}}(\cdot)$, freeing actor-critic capacity for action selection and reducing training overhead.

5.2 Generalizability

We consider how QADAPT would generalize to two- and three-dimensional arrays, where quantum dots may have more than two nearest neighbors. In this case, QADAPT scales trivially: each gate agent will simply participate in more CSDs and the cross-capacitance matrix Φ will become more dense, owing to an increased number of nearest- and second-nearest dot neighbors. While this would require re-training $f_{\text{virt}}(\cdot)$, all components of the QADAPT framework would remain unchanged and the algorithm becomes backward compatible. Compute memory can be easily sized according to resource availability and the resolution of CSD images can be adjusted to accommodate faster measurement cycles.

Here, we redirect the reader Appendix H where we apply the QADAPT framework to a completely different setting – the tuning of superconducting qubits in the presence of drive cross-talk. This shares many similarities to the quantum dot array tuning task: each qubit has configurable parameters and the adjustment of any one parameter influences the performance of nearby qubits. With minor modifications to input and output dimensions, we show that an action space factorization and paramater sharing strategy is also highly suitable for this quantum control problem.

5.3 Limitations

While QADAPT generalizes well to realistic simulated data (see Appendix B), the sim-2-real gap still remains unexplored. Real hardware can experience parameter drift, hysteresis, and time-varying voltage offsets. While the bandit reduction avoids long-horizon planning that would be destabilized by such effects, they may nonetheless hamper the online learning of Φ if cross-capacitance values vary significantly between tuning steps. This can be mitigated by increasing the Kalman filter's process noise, which relaxes posterior confidence and permits larger matrix As shown in Appendix G, this need not necessarily increase the steps to convergence.

A further limitation of QADAPT may stem from the strict assignment of agent types among barrier and plunger categories. In simulation, these model distinctly different behaviors, however, fabrication imperfections and gate dimensions may blur their respective roles on real hardware. Whether our parameter sharing strategy degrades gracefully when physical symmetries break remains to be validated. This could be mitigated, for instance, using parameter sharing masks (Li et al., 2024), which could be selected dynamically based on cross-capacitance behaviors.

Finally, we note that simulated devices are assumed to operate in an 'open' regime, whereby each quantum dot is capable of acquiring charges from a nearby reservoir. While such device architectures are standard, future devices may operate in an 'isolated' regime (Bertrand et al., 2015). This introduces non-steady state solutions and modifies charge stability diagram (CSD) features, which may reduce the effectiveness of our non-temporal credit assignment

6 Conclusion

QADAPT combines parameter sharing across gate-local agents with online action-space factorization refined by Kalman-guided cross-capacitance estimation. This enables scalable, coordinated, decentralized tuning of quantum dot arrays and zero-shot transfer across device sizes. Simulated benchmarks demonstrate strong convergence and measurement efficiency against RL and black-box baselines under fixed measurement budgets. Action-factored MARL therefore offers a route toward scalable quantum device calibration, addressing a bottleneck in large-scale quantum processors. Future work will test robustness to experimental non-idealities and validate the approach with hardware in the loop.

AI USE STATEMENT

In this work, we used generative AI tools to assist in the writing of proofs and implementation of methods. The use of AI for generating synthetic datasets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, assisting with translation, cleaning and reformating datasets, supporting qualitative and thematic data analysis, and interpreting results are not applicable to this work. Additionally, we used generative AI tools for editing software code and editing the research paper to improve readability. All LLM-generated code was verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work was conducted in accordance with the ICLR Code of Ethics. The authors have considered the broader impacts of their research and believe it contributes positively to scientific progress while posing minimal risk of harm. The research does not involve human subjects, personal data, or personally identifiable information. All data used in this study were generated by the authors. Any accompanying resources have been used in accordance with their respective licences, and their provenance is documented accordingly. All applicable licences, copyrights, and intellectual-property agreements have been respected. The authors do not foresee that this work disproportionately disadvantages any group, enables discrimination, or presents a direct risk to health, safety, security, or privacy. The authors declare no conflicts of interest that would undermine the independence of this work.

REPRODUCIBILITY STATEMENT

All code for the QADAPT algorithm and generated results are provided through an anonymous URL referenced in Appendix N.

REFERENCES

Yuri Alexeev, Marwa H Farag, Taylor L Patti, Mark E Wolf, Natalia Ares, Alán Aspuru-Guzik, Simon C Benjamin, Zhenyu Cai, Shuxiang Cao, Christopher Chamberland, et al. Artificial intelligence for quantum computing. *Nature Communications*, 16(1):10829, 2025.

Natalia Ares. Machine learning as an enabler of qubit scalability. *Nature Reviews Materials*, 6(10): 870–871, 2021.

Timothy A Baart, Pieter T Eendebak, Christian Reichl, Werner Wegscheider, and Lieven MK Vandersypen. Computer-automated tuning of semiconductor double quantum dots into the single-electron regime. *Applied Physics Letters*, 108(21), 2016.

Samuel K Bartee, Will Gilbert, Kun Zuo, Kushal Das, Tuomo Tanttu, Chih Hwan Yang, Nard Dumoulin Stuyck, Sebastian J Pauka, Rocky Y Su, Wee Han Lim, et al. Spin-qubit control with a milli-kelvin cmos chip. *Nature*, pp. 1–6, 2025.

Yuval Baum, Mirko Amico, Sean Howell, Michael Hush, Maggie Liuzzi, Pranav Mundada, Thomas Merkh, Andre RR Carvalho, and Michael J Biercuk. Experimental deep reinforcement learning for error-robust gate-set design on a superconducting quantum computer. *PRX quantum*, 2(4):040324, 2021.

Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4): 819–840, 2002.

Benoit Bertrand, Hanno Flentje, Shintaro Takada, Michihisa Yamamoto, Seigo Tarucha, Arne Ludwig, Andreas D Wieck, Christopher Bäuerle, and Tristan Meunier. Quantum manipulation of two-electron spin states in isolated double quantum dots. *Physical review letters*, 115(9):096801, 2015.

Francesco Borsoi, Nico W Hendrickx, Valentin John, Marcel Meyer, Sayr Motz, Floor Van Riggelen, Amir Sammak, Sander L De Snoo, Giordano Scappucci, and Menno Veldhorst. Shared control of a 16 semiconductor quantum dot crossbar array. *Nature Nanotechnology*, 19(1):21–27, 2024.

Djallel Bouneffouf, Irina Rish, and Charu Aggarwal. Survey on applications of multi-armed and contextual bandits. In *2020 IEEE congress on evolutionary computation (CEC)*, pp. 1–8. IEEE, 2020.

Marin Bukov and Florian Marquardt. Reinforcement learning for quantum technology. *arXiv preprint arXiv:2601.18953*, 2026.

Shuxiang Cao, Zijian Zhang, Abhishek Agarwal, Grace Bratrud, Niyaz R Beysengulov, Daniel C Cole, Alejandro Gómez Frieiro, Elena O Glen, Hao Hsu, Gang Huang, et al. Qcaleval: Benchmarking vision-language models for quantum calibration plot understanding. *arXiv preprint arXiv:2604.25884*, 2026.

Cornelius Carlsson, Jaime Saez-Mollejo, Federico Fedele, Stefano Calcaterra, Daniel Chrastina, Giovanni Isella, Georgios Katsaros, and Natalia Ares. Automated all-rf tuning for spin qubit readout and control. *arXiv preprint arXiv:2506.10834*, 2025.

Anasua Chatterjee, Paul Stevenson, Silvano De Franceschi, Andrea Morello, Nathalie P de Leon, and Ferdinand Kuemmeth. Semiconductor qubits in practice. *Nature Reviews Physics*, 3(3):157–177, 2021.

Zhe Chen et al. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182 (1):1–69, 2003.

Filippos Christianos, Georgios Papoudakis, Muhammad A Rahman, and Stefano V Albrecht. Scaling multi-agent reinforcement learning with selective parameter sharing. In *International Conference on Machine Learning*, pp. 1989–1998. PMLR, 2021.

Stefanie Czischek, Victor Yon, Marc-Antoine Genest, Marc-Antoine Roux, Sophie Rochette, Julien Camirand Lemyre, Mathieu Moras, Michel Pioro-Ladrière, Dominique Drouin, Yann Beilliard, et al. Miniaturizing neural networks for charge state autotuning in quantum dots. *Machine Learning: Science and Technology*, 3(1):015001, 2021.

Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

Christian Schroeder de Witt, Bei Peng, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. Deep multi-agent reinforcement learning for decentralized continuous cooperative control. *arXiv preprint arXiv:2003.06709*, 19, 2020.

Jurgen J Dijkema, Xin Zhang, Achilleas Bardakas, Daniel Bouman, Alice Cuzzocrea, David van Driel, Davide Girardi, Lucas EA Stehouwer, Giordano Scappucci, Anne-Marije J Zwerver, et al. Simultaneous operation of an 18-qubit modular array in germanium. *arXiv preprint arXiv:2604.01063*, 2026.

Renato Durrer, Benedikt Kratochwil, Jonne V Koski, Andreas J Landig, Christian Reichl, Werner Wegscheider, Thomas Ihn, and Eliska Greplova. Automated tuning of double quantum dots into specific charge states using neural networks. *Physical Review Applied*, 13(5):054019, 2020.

Huazhen Fang, Ning Tian, Yebin Wang, MengChu Zhou, and Mulugeta A Haile. Nonlinear bayesian estimation: From kalman filtering to a broader horizon. *IEEE/CAA Journal of Automatica Sinica*, 5(2):401–417, 2018.

Pu Feng, Junkang Liang, Size Wang, Xin Yu, Xin Ji, Yiting Chen, Kui Zhang, Rongye Shi, and Wenjun Wu. Hierarchical consensus-based multi-agent reinforcement learning for multi-robot cooperation tasks. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 642–649. IEEE, 2024.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Yvonne Y Gao, M Adriaan Rol, Steven Touzard, and Chen Wang. Practical guide for building superconducting quantum devices. *PRX quantum*, 2(4):040202, 2021.

Pierre Guilmin, Adrien Bocquet, Élie Genois, Daniel Weiss, and Ronan Gautier. Dynamiqs: an open-source python library for gpu-accelerated and differentiable simulation of quantum systems. 2025. URL https://github.com/dynamiqs/dynamiqs.

Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *International conference on autonomous agents and multiagent systems*, pp. 66–83. Springer, 2017.

Fabian Hader, Fabian Fuchs, Sarah Fleitmann, Karin Havemann, Benedikt Scherer, Jan Vogelbruch, Lotte Geck, and Stefan Van Waasen. Automated charge transition detection in quantum dot charge stability diagrams. *IEEE Transactions on Quantum Engineering*, 2025.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024. URL https://arxiv.org/abs/2301.04104.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Sepp Hochreiter and J"urgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1314–1324, 2019.

Yujing Hu, Weixun Wang, Hangtian Jia, Yixiang Wang, Yingfeng Chen, Jianye Hao, Feng Wu, and Changjie Fan. Learning to utilize shaping rewards: A new approach of reward shaping. *Advances in Neural Information Processing Systems*, 33:15931–15941, 2020.

Sandesh S Kalantre, Justyna P Zwolak, Stephen Ragole, Xingyao Wu, Neil M Zimmerman, MD Stewart Jr, and Jacob M Taylor. Machine learning techniques for state recognition and auto-tuning in quantum dots. *npj Quantum Information*, 5(1):6, 2019.

Arbaaz Khan, Ekaterina Tolstaya, Alejandro Ribeiro, and Vijay Kumar. Graph policy gradients for large scale robot control. In *Conference on robot learning*, pp. 823–834. PMLR, 2020.

Ozsel Kilinc and Giovanni Montana. Multi-agent deep reinforcement learning with extremely noisy observations. *arXiv preprint arXiv:1812.00922*, 2018.

Jens Koch, Terri M Yu, Jay Gambetta, Andrew A Houck, David I Schuster, Johannes Majer, Alexandre Blais, Michel H Devoret, Steven M Girvin, and Robert J Schoelkopf. Charge-insensitive qubit design derived from the cooper pair box. *Physical Review A—Atomic, Molecular, and Optical Physics*, 76(4):042319, 2007.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

Chihiro Kondo, Raisei Mizokuchi, Jun Yoneda, and Tetsuo Kodera. Environment model construction toward auto-tuning of quantum dot devices based on model-based reinforcement learning. *APL Machine Learning*, 3(1), 2025.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. *arXiv preprint arXiv:2109.11251*, 2021.

Xinran Li, Ling Pan, and Jun Zhang. Kaleidoscope: Learnable masks for heterogeneous multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 37:22081–22106, 2024.

Timothy Paul Lillicrap, Jonathan James Hunt, Alexander Pritzel, Nicolas Manfred Otto Heess, Tom Erez, Yuval Tassa, David Silver, and Daniel Pieter Wierstra. Continuous control with deep reinforcement learning, September 15 2020. US Patent 10,776,692.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Rahul Marchand, Lucas Schorling, Cornelius Carlsson, Jonas Schuff, Barnaby van Straaten, Taylor L Patti, Federico Fedele, Joshua Ziegler, Parth Girdhar, Pranav Vaidhyanathan, et al. End-to-end analysis of charge stability diagrams with transformers. *arXiv preprint arXiv:2508.15710*, 2025.

Pascal Massart. *Concentration inequalities and model selection: Ecole d'Eté de Probabilités de Saint-Flour XXXIII-2003*. Springer, 2007.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Hyungil Moon, Dominic T Lennon, James Kirkpatrick, Nina M van Esbroeck, Leon C Camenzind, Liuqi Yu, Florian Vigneau, Dominik M Zumbühl, G Andrew D Briggs, Michael A Osborne, et al. Machine learning enables completely automatic tuning of a quantum device faster than human experts. *Nature communications*, 11(1):4161, 2020.

Yui Muto, Michael R Zielewski, Motoya Shinozaki, Kosuke Noro, and Tomohiro Otsuka. Automatic detection of single-electron regime of quantum dots and definition of virtual gates using u-net and clustering. *arXiv preprint arXiv:2501.05878*, 2025.

Siddharth Nayak, Kenneth Choi, Wenqi Ding, Sydney Dolan, Karthik Gopalakrishnan, and Hamsa Balakrishnan. Scalable multi-agent reinforcement learning through intelligent information aggregation. In *International conference on machine learning*, pp. 25817–25833. PMLR, 2023.

John A Nelder and Roger Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

Oren Neumann and Claudius Gros. Scaling laws for a multi-agent reinforcement learning model. *arXiv preprint arXiv:2210.00849*, 2022.

V Nguyen, SB Orbell, Dominic T Lennon, Hyungil Moon, Florian Vigneau, Leon C Camenzind, Liuqi Yu, Dominik M Zumbühl, G Andrew D Briggs, Michael A Osborne, et al. Deep reinforcement learning for efficient measurement of quantum devices. *npj Quantum Information*, 7(1):100, 2021.

Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

Giovanni A Oakes, Jingyu Duan, John JL Morton, Alpha Lee, Charles G Smith, and M Zalba. Automatic virtual voltage extraction of a 2x2 array of quantum dots with machine learning. *arXiv preprint arXiv:2012.03685*, 2020.

Afshin Oroojlooy and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, 53(11):13677–13722, 2023.

Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. Facmac: Factored multi-agent centralised policy gradients. *Advances in neural information processing systems*, 34:12208–12221, 2021.

Anantha S Rao, Donovan Buterakos, Barnaby van Straaten, Valentin John, Cécile X Yu, Stefan D Oosterhout, Lucas Stehouwer, Giordano Scappucci, Menno Veldhorst, Francesco Borsoi, et al. Modular autonomous virtualization system for two-dimensional semiconductor quantum dot arrays. *Physical Review X*, 15(2):021034, 2025.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

Carl Edward Rasmussen and Christopher K I Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Matthew Reed. *Entanglement and quantum error correction with superconducting qubits*. Lulu. com, 2013.

Kevin Reuer, Jonas Landgraf, Thomas Fösel, James O'Sullivan, Liberto Beltrán, Abdulkadir Akin, Graham J Norris, Ants Remm, Michael Kerschbaum, Jean-Claude Besse, et al. Realizing a deep reinforcement learning agent for real-time quantum feedback. *Nature Communications*, 14(1): 7138, 2023.

Marc-Antoine Roux, Joffrey Rivard, Victor Yon, Alexis Morel, Dominic Leclerc, Claude Rohrbacher, El Bachir Ndiaye, Felice Francesco Tafuri, Brendan Bono, Stefan Kubicek, et al. Rapid autotuning of a sige quantum dot into the single-electron regime with machine learning and rf-reflectometry fpga-based measurements. *arXiv preprint arXiv:2509.19537*, 2025.

Lucas Schorling, Pranav Vaidhyanathan, Jonas Schuff, Miguel J Carballido, Dominik Zumbühl, Gerard Milburn, Florian Marquardt, Jakob Foerster, Michael A Osborne, and Natalia Ares. Meta-learning characteristics and dynamics of quantum systems. *arXiv preprint arXiv:2503.10492*, 2025.

Jonas Schuff, Miguel J Carballido, Madeleine Kotzagiannidis, Juan Carlos Calvo, Marco Caselli, Jacob Rawling, David L Craig, Barnaby van Straaten, Brandon Severin, Federico Fedele, et al. Fully autonomous tuning of a spin qubit. *Nature Electronics*, pp. 1–10, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Rudy Semola, Lorenzo Moro, Davide Bacciu, and Enrico Prati. Deep reinforcement learning quantum control on ibmq platforms and qiskit pulse. In *2022 ieee international conference on quantum computing and engineering (qce)*, pp. 759–762. IEEE, 2022.

Brandon Severin, Dominic T Lennon, Leon C Camenzind, Florian Vigneau, Federico Fedele, Daniel Jirovec, Andrea Ballabio, Daniel Chrastina, Giovanni Isella, Mathieu de Kruijf, et al. Cross-architecture tuning of silicon and sige-based quantum devices using machine learning. *Scientific Reports*, 14(1):17281, 2024.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2016.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, pp. 5887–5896. PMLR, 2019.

Paul Steinacker, Nard Dumoulin Stuyck, Wee Han Lim, Tuomo Tanttu, MengKe Feng, Santiago Serrano, Andreas Nickl, Marco Candido, Jesus D Cifuentes, Ensar Vahapoglu, et al. Industry-compatible silicon spin-qubit unit cells exceeding 99% fidelity. *Nature*, pp. 1–7, 2025.

Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29, 2016.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2nd edition, 2018.

Ming Tan et al. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pp. 330–337, 1993.

Kale-ab Abebe Tessera, Arrasy Rahman, Amos Storkey, and Stefano V Albrecht. Hypermarl: Adaptive hypernetworks for multi-agent rl. *arXiv preprint arXiv:2412.04233*, 2024.

Tatsuo Tsuzuki, Hideaki Yuta, Yui Muto, Arne Ludwig, Andreas Dirk Wieck, Akira Oiwa, and Takafumi Fujita. Cnn-assisted automatic cross-capacitance matrix update for virtual-gate control of quantum dot arrays. *Japanese Journal of Applied Physics*, 65(1):01SP03, 2026.

Pranav Vaidhyanathan, Aristotelis Papatheodorou, Mark T Mitchison, Natalia Ares, and Ioannis Havoutis. Metasym: A symplectic meta-learning framework for physical intelligence. *arXiv preprint arXiv:2502.16667*, 2025.

Pranav Vaidhyanathan, Florian Marquardt, Mark T Mitchison, and Natalia Ares. Quantum feedback control with a transformer neural network architecture. *Physical Review Research*, 8(1):L012043, 2026.

Nina M van Esbroeck, Dominic T Lennon, Hyungil Moon, Vu Nguyen, Florian Vigneau, Leon C Camenzind, Liuqi Yu, Dominik M Zumbühl, G Andrew D Briggs, Dino Sejdinovic, et al. Quantum device fine-tuning using unsupervised embedding learning. *New Journal of Physics*, 22(9):095003, 2020.

Barnaby van Straaten, Joseph Hickie, Lucas Schorling, Jonas Schuff, Federico Fedele, and Natalia Ares. Codebase release 1.3 for qarray. *SciPost Physics Codebases*, pp. 035, 2024a.

Barnaby van Straaten, Joseph Hickie, Lucas Schorling, Jonas Schuff, Federico Fedele, and Natalia Ares. Qarray: A gpu-accelerated constant capacitance model simulator for large quantum dot arrays. *SciPost Physics Codebases*, pp. 035, 2024b.

Barnaby Van Straaten, Federico Fedele, Florian Vigneau, Joseph Hickie, Daniel Jirovec, Andrea Ballabio, Daniel Chrastina, Giovanni Isella, Georgios Katsaros, and Natalia Ares. All-rf-based coarse-tuning algorithm for quantum devices using machine learning. *Physical Review Applied*, 24 (5):054030, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Christian Volk, Anne-Marije J Zwerver, Uditendu Mukhopadhyay, Pieter T Eendebak, Cornelis Jacobus van Diepen, Juan Pablo Dehollain, Toivo Hensgens, Takafumi Fujita, Christian Reichl, Werner Wegscheider, et al. Loading a quantum-dot based "qubyte" register. *npj Quantum Information*, 5(1):29, 2019.

Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. *Advances in Neural Information Processing Systems*, 35:16509–16521, 2022.

Jaap J Wesdorp, Eric Hyyppä, Joona Andersson, Janos Adam, Rohit Beriwal, Ville Bergholm, Saga Dahl, Simone Diego Fasciati, Alejandro Gomez Friero, Zheming Gao, et al. Mitigating crosstalk errors for simultaneous single-qubit gates on a superconducting quantum processor. *arXiv preprint arXiv:2603.11018*, 2026.

Zhiyuan Xu, Jian Tang, Jingsong Meng, Weiyi Zhang, Yanzhi Wang, Chi Harold Liu, and Dejun Yang. Experience-driven networking: A deep reinforcement learning based approach. In *IEEE INFOCOM 2018-IEEE conference on computer communications*, pp. 1871–1879. IEEE, 2018.

Victor Yon, Bastien Galaup, Claude Rohrbacher, Joffrey Rivard, Alexis Morel, Dominic Leclerc, Clement Godfrin, Ruoyu Li, Stefan Kubicek, Kristiaan De Greve, et al. Experimental online quantum dots charge autotuning using neural networks. *Nano Letters*, 25(10):3717–3725, 2025.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.

Zijian Zhao, Jing Gao, and Sen Li. Bridging marl to sarl: An order-independent multi-agent transformer via latent consensus. *arXiv preprint arXiv:2604.13472*, 2026.

Yifan Zhong, Jakub Grudzien Kuba, Xidong Feng, Siyi Hu, Jiaming Ji, and Yaodong Yang. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25(32): 1–67, 2024.

Joshua Ziegler, Florian Luthi, Mick Ramsey, Felix Borjans, Guoji Zheng, and Justyna P Zwolak. Tuning arrays with rays: Physics-informed tuning of quantum dot charge states. *Physical Review Applied*, 20(3):034067, 2023.

Justyna P Zwolak, Thomas McJunkin, Sandesh S Kalantre, JP Dodson, ER MacQuarrie, DE Savage, MG Lagally, SN Coppersmith, Mark A Eriksson, and Jacob M Taylor. Autotuning of double-dot devices in situ with machine learning. *Physical review applied*, 13(3):034075, 2020.

Justyna P Zwolak, Jacob M Taylor, Reed W Andrews, Jared Benson, Garnett W Bryant, Donovan Buterakos, Anasua Chatterjee, Sankar Das Sarma, Mark A Eriksson, Eliška Greplová, et al. Data needs and challenges for quantum dot devices automation. *npj Quantum Information*, 10(1):105, 2024.

A PROOFS ON VIRTUALIZATION AND FILTERS

Notation. For $z \in \mathbb{R}^d$, define $\|z\|_2 := \sqrt{z^\top z}$. For a matrix $A \in \mathbb{R}^{m\times n}$, define the spectral norm

$$\|A\|_2 := \sup_{\|x\|_2=1} \|Ax\|_2.$$

Recall that $\|A\|_2$ equals the largest singular value $\sigma_{\max}(A)$, and for any x,

$$\|Ax\|_2 \le \|A\|_2 \, \|x\|_2 \qquad \text{(by the definition of the supremum).}$$

We also use the elementary implication: if $0 \prec B \preceq A$ (PSD order), then $A^{-1} \preceq B^{-1}$.

A.0.1 LOCAL LINEARIZATION MODEL

We start by fixing an operating point $u_0 \in \mathbb{R}^D$ (plunger voltages). Let $x(u) \in \mathbb{R}^D$ denote the local device state. Let $J(u) := \nabla x(u) \in \mathbb{R}^{D\times D}$.

Assumption A.1 (Locally Lipschitz Jacobian)**.** There exists a convex neighborhood $mathcalU \subset \mathbb{R}^D$ and a constant $L \ge 0$ such that for all $u, v \in \mathcal{U}$,

$$\|J(u) - J(v)\|_2 \le L\|u - v\|_2.$$

Lemma A.2 (First-order expansion with explicit remainder bound)**.** *Fix $u_0 \in \mathcal{U}$ and define $C := J(u_0)$. For any $u \in \mathcal{U}$, there exists a remainder vector $r(u) \in \mathbb{R}^D$ such that*

$$x(u) = x(u_0) + C(u - u_0) + r(u),$$

and $r(u)$ satisfies the explicit bound

$$\|r(u)\|_2 \le \frac{L}{2}\|u - u_0\|_2^2.$$

Proof. Define the path $\gamma : [0, 1] \to \mathcal{U}$ by $\gamma(t) = u_0 + t(u - u_0)$ (convexity ensures $\gamma(t) \in \mathcal{U}$). By the fundamental theorem of calculus,

$$x(u) - x(u_0) = \int_0^1 \frac{d}{dt} x(\gamma(t))\, dt = \int_0^1 J(\gamma(t))(u - u_0)\, dt.$$

Add and subtract $C(u - u_0)$ inside the integral and define

$$r(u) := \int_0^1 \big(J(\gamma(t)) - C\big)(u - u_0)\, dt.$$

Then the expansion holds. For the bound, use the definition of $\|\cdot\|_2$ and Assumption A.1: for each $t \in [0, 1]$,

$$\|\big(J(\gamma(t))-C\big)(u-u_0)\|_2 \le \|J(\gamma(t))-C\|_2 \, \|u-u_0\|_2 \le L\|\gamma(t)-u_0\|_2 \, \|u-u_0\|_2 = Lt\|u-u_0\|_2^2.$$

Integrating t from 0 to 1 gives

$$\|r(u)\|_2 \le \int_0^1 Lt\|u - u_0\|_2^2\, dt = \frac{L}{2}\|u - u_0\|_2^2.$$

□

A.0.2 ONE-STEP QUADRATIC OBJECTIVE AND GREEDY OPTIMUM

Let $x^\star \in \mathbb{R}^D$ be a target and define $e(u) := x(u) - x^\star$. At iteration t, the algorithm chooses Δu_t and applies $u_{t+1} = u_t + \Delta u_t$. In the linearized model (dropping $r(\cdot)$ for the moment), the one-step surrogate cost is

$$J_t(\Delta u) := \frac{1}{2}\|e_t + C\Delta u\|_2^2, \qquad e_t := e(u_t).$$

Proposition A.3 (Greedy optimum in raw coordinates)**.** *If C is invertible, the unique minimizer of $J_t(\Delta u)$ is*

$$\Delta u_t^\star = -C^{-1}e_t,$$

and under the linear model the next error satisfies $e_{t+1} = 0$.

Proof. Write $J_t(\Delta u) = \frac{1}{2}(e_t + C\Delta u)^\top (e_t + C\Delta u)$. Differentiating w.r.t. Δu yields

$$\nabla_{\Delta u} J_t(\Delta u) = C^\top (e_t + C\Delta u).$$

At a minimizer, $\nabla_{\Delta u} J_t(\Delta u) = 0$, i.e.

$$C^\top C\, \Delta u = -C^\top e_t.$$

Since C is invertible, $C^\top C$ is positive definite and invertible, hence the solution is unique:

$$\Delta u = -(C^\top C)^{-1} C^\top e_t.$$

Because C is invertible, one checks directly that $(C^\top C)^{-1} C^\top = C^{-1}$ (multiply by C on the right to obtain the identity), so $\Delta u_t^\star = -C^{-1} e_t$. Substituting gives $e_{t+1} = e_t + C\Delta u_t^\star = 0$. □

A.0.3 VIRTUALIZATION AS A PRECONDITIONED COORDINATE SYSTEM

QADAPT constructs an estimate $\widehat{C}_t$ of the cross-capacitance and uses it to define a virtualized coordinate Δv_t through the linear change of variables

$$\Delta u_t = \widehat{C}_t^{-1} \Delta v_t.$$

Under the linear model, the error evolves as

$$e_{t+1} = e_t + C\widehat{C}_t^{-1} \Delta v_t.$$

Theorem A.4 (Exact cancellation in ideal virtual coordinates)**.** *Assume C and $\widehat{C}_t$ are invertible and choose $\Delta v_t = -e_t$. Define the mismatch matrix*

$$E_t := I - C\widehat{C}_t^{-1}.$$

Then

$$e_{t+1} = E_t e_t.$$

Moreover, if there exists $\rho \in [0, 1)$ such that $\|E_t\|_2 \le \rho$ for all t, then

$$\|e_t\|_2 \le \rho^t \|e_0\|_2 \qquad \text{for all } t \ge 0.$$

Proof. Substitute $\Delta v_t = -e_t$ into the update:

$$e_{t+1} = e_t - C\widehat{C}_t^{-1} e_t = (I - C\widehat{C}_t^{-1}) e_t = E_t e_t.$$

To bound $\|e_{t+1}\|_2$, use the definition of $\|E_t\|_2$:

$$\|e_{t+1}\|_2 = \|E_t e_t\|_2 \le \|E_t\|_2 \|e_t\|_2 \le \rho \|e_t\|_2.$$

Iterating this inequality yields $\|e_t\|_2 \le \rho^t \|e_0\|_2$. □

Mismatch in terms of estimation error. Let $\Delta_t := \widehat{C}_t - C$.

Lemma A.5 (Mismatch identity and explicit norm bound)**.** *If $\widehat{C}_t$ is invertible, then*

$$E_t = \Delta_t \widehat{C}_t^{-1}.$$

Consequently,

$$\|E_t\|_2 \le \|\Delta_t\|_2 \, \|\widehat{C}_t^{-1}\|_2.$$

Proof. Compute

$$C\widehat{C}_t^{-1} = (\widehat{C}_t - \Delta_t)\widehat{C}_t^{-1} = \widehat{C}_t \widehat{C}_t^{-1} - \Delta_t \widehat{C}_t^{-1} = I - \Delta_t \widehat{C}_t^{-1}.$$

Hence $E_t = I - C\widehat{C}_t^{-1} = \Delta_t \widehat{C}_t^{-1}$.

For the norm bound, start from the definition of $\|E_t\|_2$:

$$\|E_t\|_2 = \sup_{\|x\|_2 = 1} \|E_t x\|_2 = \sup_{\|x\|_2 = 1} \|\Delta_t \widehat{C}_t^{-1} x\|_2.$$

For each such x, let $y := \widehat{C}_t^{-1} x$. Then $\|\Delta_t y\|_2 \le \|\Delta_t\|_2 \|y\|_2$. Therefore

$$\|\Delta_t \widehat{C}_t^{-1} x\|_2 \le \|\Delta_t\|_2 \, \|\widehat{C}_t^{-1} x\|_2 \le \|\Delta_t\|_2 \, \|\widehat{C}_t^{-1}\|_2 \, \|x\|_2 = \|\Delta_t\|_2 \, \|\widehat{C}_t^{-1}\|_2.$$

Taking the supremum over $\|x\|_2 = 1$ yields the claimed inequality. □

Invertibility and stability of the inverse.

Lemma A.6 (Neumann-series inverse bound)**.** *Let C be invertible and $\widehat{C} = C + \Delta$. If $\|C^{-1}\|_2\|\Delta\|_2 < 1$, then $\widehat{C}$ is invertible and*

$$\|\widehat{C}^{-1}\|_2 \le \frac{\|C^{-1}\|_2}{1 - \|C^{-1}\|_2\|\Delta\|_2}.$$

Proof. Factor $\widehat{C}$ as

$$\widehat{C} = C(I + C^{-1}\Delta).$$

Let $X := C^{-1}\Delta$. The assumption implies $\|X\|_2 < 1$. Consider the series $S := \sum_{k=0}^{\infty}(-X)^k$. For each partial sum $S_n = \sum_{k=0}^{n}(-X)^k$,

$$(I + X)S_n = \sum_{k=0}^{n}(-X)^k + \sum_{k=0}^{n}X(-X)^k = I + \sum_{k=1}^{n}(-X)^k + \sum_{k=1}^{n+1}(-X)^k = I - (-X)^{n+1}.$$

Since $\|X\|_2 < 1$, we have $\|(-X)^{n+1}\|_2 \le \|X\|_2^{n+1} \to 0$, hence $S_n \to S$ and $(I + X)S = I$. Thus $(I + X)^{-1} = S$ exists and

$$\|(I + X)^{-1}\|_2 = \|S\|_2 \le \sum_{k=0}^{\infty}\|X\|_2^k = \frac{1}{1 - \|X\|_2}.$$

Therefore

$$\widehat{C}^{-1} = (I + C^{-1}\Delta)^{-1}C^{-1}, \qquad \|\widehat{C}^{-1}\|_2 \le \|(I + X)^{-1}\|_2\,\|C^{-1}\|_2 \le \frac{\|C^{-1}\|_2}{1 - \|C^{-1}\|_2\|\Delta\|_2}.$$

□

Corollary A.7 (Explicit sufficient condition for contraction)**.** *Assume C is invertible and $\widehat{C}_t = C + \Delta_t$. If $\|C^{-1}\|_2\|\Delta_t\|_2 < \frac{1}{2}$, then $\|E_t\|_2 < 1$. In fact,*

$$\|E_t\|_2 \le \frac{\|C^{-1}\|_2\|\Delta_t\|_2}{1 - \|C^{-1}\|_2\|\Delta_t\|_2}.$$

Proof. By Lemma A.5,

$$\|E_t\|_2 \le \|\Delta_t\|_2\,\|\widehat{C}_t^{-1}\|_2.$$

Apply Lemma A.6 with $\Delta = \Delta_t$:

$$\|\widehat{C}_t^{-1}\|_2 \le \frac{\|C^{-1}\|_2}{1 - \|C^{-1}\|_2\|\Delta_t\|_2}.$$

Multiplying yields the claimed bound. Let $a := \|C^{-1}\|_2\|\Delta_t\|_2$. If $a < 1/2$, then

$$\frac{a}{1 - a} < \frac{1/2}{1 - 1/2} = 1,$$

hence $\|E_t\|_2 < 1$. □

A.1 Virtualization Improves Conditioning (Preconditioning)

Consider minimizing the one-step quadratic surrogate

$$J_t(\Delta u) := \frac{1}{2}\|e_t + C\,\Delta u\|_2^2.$$

In voltage coordinates, the Hessian is

$$\nabla^2_{\Delta u}J_t(\Delta u) = C^\top C,$$

so the conditioning of the local quadratic model is governed by

$$\kappa(C^\top C) = \frac{\lambda_{\max}(C^\top C)}{\lambda_{\min}(C^\top C)} = \kappa(C)^2,$$

which can be large when gate cross-talk makes C ill-conditioned.

Under virtualization, we change variables via

$$\Delta u = \widehat{C}_t^{-1} \Delta v.$$

Substituting into J_t yields the virtual-coordinate objective

$$J_t(\widehat{C}_t^{-1}\Delta v) = \frac{1}{2}\left\| e_t + C\,\widehat{C}_t^{-1}\Delta v \right\|_2^2 = \frac{1}{2}\left\| e_t + M_t \Delta v \right\|_2^2, \qquad M_t := C\,\widehat{C}_t^{-1}.$$

The Hessian with respect to Δv is therefore

$$\nabla^2_{\Delta v} J_t(\widehat{C}_t^{-1}\Delta v) = M_t^\top M_t.$$

Hence the conditioning of the virtual-coordinate quadratic problem is

$$\kappa(M_t^\top M_t) = \frac{\lambda_{\max}(M_t^\top M_t)}{\lambda_{\min}(M_t^\top M_t)} = \frac{\sigma_{\max}(M_t)^2}{\sigma_{\min}(M_t)^2} = \kappa(M_t)^2.$$

Thus, virtualization improves conditioning exactly when M_t is close to the identity.

Recall from Theorem B.4 that the mismatch matrix is defined as

$$E_t := I - C\,\widehat{C}_t^{-1}.$$

Equivalently,

$$M_t = C\,\widehat{C}_t^{-1} = I - E_t.$$

The next lemma shows that small mismatch implies $\kappa(M_t)$ is close to 1.

Lemma A.8 (Condition number bound for the preconditioned system)**.** *Let*

$$M_t := C\,\widehat{C}_t^{-1} \qquad \textit{and} \qquad E_t := I - M_t.$$

If

$$\|E_t\|_2 < 1,$$

then M_t is nonsingular and

$$\kappa(M_t) \le \frac{1 + \|E_t\|_2}{1 - \|E_t\|_2}.$$

Consequently,

$$\kappa(M_t^\top M_t) \le \left(\frac{1 + \|E_t\|_2}{1 - \|E_t\|_2} \right)^2.$$

Proof. We use the singular value characterizations

$$\sigma_{\max}(M_t) = \sup_{\|x\|_2 = 1} \|M_t x\|_2, \qquad \sigma_{\min}(M_t) = \inf_{\|x\|_2 = 1} \|M_t x\|_2.$$

Since $M_t = I - E_t$, for any x with $\|x\|_2 = 1$,

$$\|M_t x\|_2 = \|(I - E_t)x\|_2 = \|x - E_t x\|_2.$$

Upper bound. By the triangle inequality,

$$\|x - E_t x\|_2 \le \|x\|_2 + \|E_t x\|_2 = 1 + \|E_t x\|_2.$$

By the definition of $\|E_t\|_2$,

$$\|E_t x\|_2 \le \|E_t\|_2 \|x\|_2 = \|E_t\|_2.$$

Therefore $\|M_t x\|_2 \le 1 + \|E_t\|_2$ for all unit vectors x, and taking the supremum gives

$$\sigma_{\max}(M_t) \le 1 + \|E_t\|_2.$$

Lower bound. By the reverse triangle inequality $\|a-b\|_2 \geq \big|\|a\|_2 - \|b\|_2\big|$, we have

$$\|x - E_t x\|_2 \geq \big|\|x\|_2 - \|E_t x\|_2\big| = 1 - \|E_t x\|_2.$$

Using again $\|E_t x\|_2 \leq \|E_t\|_2 \|x\|_2 = \|E_t\|_2$, we obtain

$$\|M_t x\|_2 \geq 1 - \|E_t\|_2 \qquad \text{for all } \|x\|_2 = 1.$$

Taking the infimum over unit vectors yields

$$\sigma_{\min}(M_t) \geq 1 - \|E_t\|_2.$$

Because $\|E_t\|_2 < 1$, the right-hand side is strictly positive, hence $\sigma_{\min}(M_t) > 0$ and M_t is nonsingular.

Condition number bound. Finally,

$$\kappa(M_t) = \frac{\sigma_{\max}(M_t)}{\sigma_{\min}(M_t)} \leq \frac{1 + \|E_t\|_2}{1 - \|E_t\|_2}.$$

Squaring both sides and using $\kappa(M_t^\top M_t) = \kappa(M_t)^2$ gives the last claim. □

A.1.1 KALMAN FILTERING SHRINKS THE MISMATCH BOUND

QADAPT treats the unknown couplings as latent variables and uses a Kalman filter to refine them from CSD-derived observations. We formalize the contraction improvement in the standard linear-Gaussian setting.

Let $c := \mathrm{vec}(C) \in \mathbb{R}^{D^2}$ be the vectorized coupling parameters and suppose that at each step t we obtain an observation $y_t \in \mathbb{R}^m$ of the form

$$y_t = H_t c + \nu_t, \qquad \nu_t \sim \mathcal{N}(0, R_t), \qquad R_t \succ 0.$$

Let $(\hat{c}_t, P_t)$ be the Kalman posterior mean and covariance after assimilating $y_{1:t}$. In information form, the covariance satisfies

$$P_t^{-1} = P_{t-1}^{-1} + H_t^\top R_t^{-1} H_t.$$

Lemma A.9 (Posterior covariance is monotone decreasing)**.** *For all t, $P_t \preceq P_{t-1}$.*

Proof. Since $R_t \succ 0$, the matrix $H_t^\top R_t^{-1} H_t$ is positive semidefinite, so

$$P_t^{-1} - P_{t-1}^{-1} = H_t^\top R_t^{-1} H_t \succeq 0, \qquad \text{i.e.} \qquad P_t^{-1} \succeq P_{t-1}^{-1}.$$

We now use the implication $A \succeq B \succ 0 \Rightarrow A^{-1} \preceq B^{-1}$. To prove it: let $B^{1/2}$ be the PSD square root. Then $A \succeq B$ implies

$$B^{-1/2} A B^{-1/2} \succeq I.$$

Inverting both sides (order reverses under inversion for PSD matrices) yields

$$(B^{-1/2} A B^{-1/2})^{-1} \preceq I,$$

and multiplying by $B^{-1/2}$ on left and right gives $A^{-1} \preceq B^{-1}$. Applying this with $A = P_t^{-1}$ and $B = P_{t-1}^{-1}$ yields $P_t \preceq P_{t-1}$. □

Write $\widehat{C}_t := \mathrm{unvec}(\hat{c}_t)$ and $\Delta_t := \widehat{C}_t - C$. Conditioned on $y_{1:t}$, we have $c - \hat{c}_t \sim \mathcal{N}(0, P_t)$. Thus $\mathrm{vec}(\Delta_t) \sim \mathcal{N}(0, P_t)$ and

$$\mathbb{E}\|\Delta_t\|_F^2 = \mathbb{E}\|\mathrm{vec}(\Delta_t)\|_2^2 = \mathrm{tr}(P_t).$$

High-probability bound Let $n := D^2$. Let $g \sim \mathcal{N}(0, I_n)$ and write $\mathrm{vec}(\Delta_t) = P_t^{1/2} g$. A standard chi-square concentration inequality (Laurent–Massart) (Massart, 2007) gives that for any $s > 0$,

$$\mathbb{P}\big(\|g\|_2^2 \geq n + 2\sqrt{ns} + 2s\big) \leq e^{-s}.$$

Setting $s = \log(1/\delta)$ yields: with probability at least $1 - \delta$,

$$\|g\|_2 \leq \sqrt{n + 2\sqrt{n\log(1/\delta)} + 2\log(1/\delta)}.$$

Moreover,

$$\|\Delta_t\|_F = \|\mathrm{vec}(\Delta_t)\|_2 = \|P_t^{1/2} g\|_2 \leq \|P_t^{1/2}\|_2 \, \|g\|_2 = \sqrt{\|P_t\|_2} \, \|g\|_2,$$

where the inequality follows from the definition of $\|P_t^{1/2}\|_2$. Therefore, with probability at least $1 - \delta$,

$$\|\Delta_t\|_2 \leq \|\Delta_t\|_F \leq \sqrt{\|P_t\|_2} \sqrt{n + 2\sqrt{n\log(1/\delta)} + 2\log(1/\delta)}.$$

Combining this with Corollary A.7 yields an explicit sufficient condition on the Kalman covariance for strict contraction (with high probability): if

$$\|C^{-1}\|_2 \sqrt{\|P_t\|_2} \sqrt{n + 2\sqrt{n\log(1/\delta)} + 2\log(1/\delta)} < \frac{1}{2},$$

then $\|E_t\|_2 < 1$ and Theorem A.4 gives geometric decay $\|e_t\|_2 \leq \rho^t \|e_0\|_2$ on the linear model.

A.1.2 Robustness: policy error and nonlinear remainder

We now re-introduce (i) imperfect virtual actions and (ii) the Taylor remainder from Lemma A.2. Assume we pick

$$\Delta v_t = -e_t + \eta_t$$

for some action error η_t, and keep the first-order model around u_t. Let $C_t := J(u_t)$. By Lemma A.2 applied at u_t, there exists r_t such that

$$e_{t+1} = e_t + C_t \Delta u_t + r_t, \qquad \|r_t\|_2 \leq \frac{L}{2} \|\Delta u_t\|_2^2.$$

With $\Delta u_t = \widehat{C}_t^{-1} \Delta v_t$, we obtain

$$e_{t+1} = \big(I - C_t \widehat{C}_t^{-1}\big) e_t + C_t \widehat{C}_t^{-1} \eta_t + r_t.$$

Taking norms and using the definition of $\|\cdot\|_2$ as in Lemma A.5 yields the explicit inequality

$$\|e_{t+1}\|_2 \leq \|I - C_t \widehat{C}_t^{-1}\|_2 \, \|e_t\|_2 + \|C_t \widehat{C}_t^{-1}\|_2 \, \|\eta_t\|_2 + \frac{L}{2} \|\widehat{C}_t^{-1}\|_2^2 \, \|\Delta v_t\|_2^2.$$

In particular, if $\|I - C_t \widehat{C}_t^{-1}\|_2 \leq \rho < 1$ uniformly and $\eta_t \equiv 0$, then since $\|\Delta v_t\|_2 = \|e_t\|_2$,

$$\|e_{t+1}\|_2 \leq \rho \|e_t\|_2 + \frac{L}{2} \|\widehat{C}_t^{-1}\|_2^2 \|e_t\|_2^2,$$

which implies strict contraction for all $\|e_t\|_2$ in a sufficiently small neighborhood (standard local-stability argument).

Proposition A.10 (Objective-level factorization induced by virtualization)**.** *Let*

$$J_t(\Delta u) = \frac{1}{2} \|e_t + C_t \Delta u\|_2^2, \qquad e_t := x(u_t) - x^\star, \qquad C_t := \nabla x(u_t).$$

Then:

(i) J_t is separable across agents, i.e.

$$J_t(\Delta u) = c_t + \sum_{i=1}^{D} f_{t,i}(\Delta u_i),$$

if and only if $C_t^\top C_t$ is diagonal.

(ii) If C_t is invertible and we use exact virtual coordinates

$$\Delta u = C_t^{-1}\Delta v,$$

then

$$J_t(\Delta v) = \frac{1}{2}\sum_{i=1}^{D}(e_{t,i} + \Delta v_i)^2,$$

so the one-step control objective is exactly separable.

(iii) With an estimate $\hat{C}_t$, define

$$E_t := I - C_t\hat{C}_t^{-1}.$$

Then

$$J_t^{\mathrm{virt}}(\Delta v) = \frac{1}{2}\|e_t + (I - E_t)\Delta v\|_2^2.$$

If

$$\widetilde{J}_t(\Delta v) := \frac{1}{2}\|e_t + \Delta v\|_2^2,$$

then

$$|J_t^{\mathrm{virt}}(\Delta v) - \widetilde{J}_t(\Delta v)| \le \|E_t\|_2\|e_t + \Delta v\|_2\|\Delta v\|_2 + \frac{1}{2}\|E_t\|_2^2\|\Delta v\|_2^2.$$

Proof. Expand

$$J_t(\Delta u) = \frac{1}{2}e_t^\top e_t + (C_t^\top e_t)^\top \Delta u + \frac{1}{2}\Delta u^\top C_t^\top C_t \Delta u.$$

Hence for $i \neq j$,

$$\frac{\partial^2 J_t}{\partial \Delta u_i \partial \Delta u_j} = (C_t^\top C_t)_{ij}.$$

So all mixed partials vanish if and only if $C_t^\top C_t$ is diagonal, which is equivalent to separability for this quadratic objective.

Now substitute $\Delta u = C_t^{-1}\Delta v$:

$$J_t(\Delta v) = \frac{1}{2}\|e_t + \Delta v\|_2^2 = \frac{1}{2}\sum_i (e_{t,i} + \Delta v_i)^2.$$

For the approximate case, write $M_t = C_t\hat{C}_t^{-1} = I - E_t$, so

$$J_t^{\mathrm{virt}}(\Delta v) = \frac{1}{2}\|e_t + M_t\Delta v\|_2^2.$$

Then

$$J_t^{\mathrm{virt}}(\Delta v) - \widetilde{J}_t(\Delta v) = -\langle e_t + \Delta v,\ E_t\Delta v\rangle + \frac{1}{2}\|E_t\Delta v\|_2^2,$$

and the bound follows from Cauchy–Schwarz and

$$\|E_t\Delta v\|_2 \le \|E_t\|_2\|\Delta v\|_2.$$

□

This shows that other MARL techniques such as MADDPG, QMIX, and FACMAC learn coordination in a fixed action space through centralized critics or value mixing. QADAPT instead tries to make the control problem itself additive by learning a physically grounded change of coordinates.

B Details on the Simulated Quantum Dot Environment

In our simulator, quantum dot array configurations are randomized to ensure a broad and realistic training environment, including noise strengths, electrostatic couplings, and lever arms. We also reduce the contrast of charge stability diagram features toward the edges of our measurement environment, mimicking real devices whereby highly occupied quantum dots crossover into a classical transport regime and transition lines vanish. By accounting for such physical behaviors, we aim to bridge the sim-2-real gap of device tuning.

Table 2: Details on the base QArray quantum dot simulator. When a min and a max value are specified, we sample uniformly at random between those limits at each reset of the environment. Cxy denotes the cross-capacitance matrix between potentials of type x and of type y, where d = dot potentials, g = plunger gates, b = barrier gates.

Parameter	Min	Max	Value
Cdd nearest coupling	0	0.2	-
Cdd second nearest coupling	0	0.1	-
Cgd lever arm	0.95	1.0	-
Cgd nearest coupling	0.3	0.7	-
Cgd second nearest coupling	0.01	0.3	-
Cgd third nearest coupling	0	0.01	-
Dot to sensor coupling	0.035	0.05	-
Cbd nearest coupling	0.04	0.08	-
Cbd second nearest coupling	0.01	0.03	-
Cbd third nearest coupling	0.005	0.015	-
Cbg nearest coupling	0.08	0.15	-
Cbg second nearest coupling	0.03	0.18	-
Cbg third nearest coupling	0.01	0.03	-
Cbb nearest coupling	0.03	0.08	-
Cbb second nearest neighbour	0.01	0.03	-
Cbb third nearest neighbour	0.005	0.015	-
Barrier to sensor coupling	0.0003	0.001	-
Telegraph noise transition probability	0	0.01	-
Telegraph noise amplitude	0	0.012	
Latching noise lead coupling probability	0.2	1.0	-
Latching noise inter-dot coupling probability	0.2	1.0	-
Temperature (mK)	50	200	-
Coulomb peak width	0	0.4	-
Base tunnel coupling	0.5	3.0	-
Tunnel coupling exponential factor (α)	0.0001	0.0008	-
Max charge carriers	-	-	4
Optimal plunger dot charge occupancy	-	-	1
Optimal sensor charge occupancy (fractional)	-	-	0.53

Table 3: Details on the gym environment. When a min and a max value are specified, we sample uniformly at random between those limits at each reset of the environment. Distances are in units of simulator volts, which has dimensions of voltage but scaled up to equal the transition line spacing.

Parameter	Min	Max	Value
Total plunger voltage range	80.0	100.0	-
Scan size	3.0	4.0	-
Total barrier voltage range	20.0	30.0	-
Lower radius for zero additive white noise	20.0	30.0	-
Distance to maximum additive white noise	5.0	10.0	-
Distance to total white noise	30.0	40.0	-
Maximum additive white noise amplitude	-	-	0.05
Distance to zero reward	-	-	40.0
Distance to half reward	-	-	1.0
Distance to zero barrier reward	-	-	6.0

C BASELINE HYPERPARAMETER SELECTION

All baselines received comparable tuning effort. The classical baselines (Nelder–Mead, L-BFGS-B and basin hopping) were each tuned by grid search over 27 to 36 configurations, with 100 trials per configuration under the same scan budget; the selected values are reported alongside each grid. Bayesian optimization uses the standard modern stack (BoTorch defaults: GP surrogate, LogEI acquisition) with the same sliding-window scaffold as the other classical baselines, in which the optimizer acts either on all gates jointly or pairwise, on windows of two adjacent plungers and the barrier between them. For DreamerV3, we follow the authors' intended protocol of a single fixed configuration (the 12M preset), which has been validated across a wide range of domains and does not require per-task tuning. All methods were evaluated under the same protocol: 100 trials per array size and identical initialization distributions.

Table 4: Hyperparameter selection for each baseline. All grid searches used the same environment, seed (42), trial count (100) and scan budget (500) per configuration.

Baseline	Selection method	Configurations evaluated
Nelder–Mead	Grid search	36 ($4 \times 3 \times 3$ grid)
L-BFGS-B	Grid search	27 ($3 \times 3 \times 3$ grid)
Basin hopping (L-BFGS-B local)	Grid search	27 planned (19 completed)
Bayesian optimization (GP)	BoTorch defaults	–
DreamerV3	Fixed default hyperparameters	1
Random search	No hyperparameters	–

Table 5: Nelder–Mead hyperparameters. SciPy's default settings (`xatol` $=$ `fatol` $= 10^{-4}$, automatic initial simplex) were not used in the pairwise benchmark; instead, a custom initial simplex (vertices clipped to the voltage bounds) was used, with the simplex scale and tolerances selected by the 36-point grid search.

Parameter	Values swept	Final value
Initial simplex step, plungers (V)	20, 35, 50, 70	35
Initial simplex step, barriers (V)	2, 4, 8	4
`xatol` $=$ `fatol`	0.05, 0.1, 0.5	0.1
Bounds	–	Enabled (full voltage ranges)
Per-window iteration cap	–	100

Table 6: L-BFGS-B hyperparameters. All 27 swept configurations achieved 0% success at the 500-scan budget, so SciPy defaults were retained for `gtol` and `maxcor`, with `ftol` $= 10^{-6}$. A global-search variant (SciPy `basinhopping` with L-BFGS-B local minimization) was additionally swept over temperature $T \in \{0.5, 1.0, 2.0\}$, perturbation step $\in \{2, 5, 10\}$ V and iterations $\in \{50, 100, 200\}$; it also achieved 0% success in all completed configurations.

Parameter	Values swept	Final value	SciPy default?
`ftol`	10^{-7}, 10^{-5}, 10^{-3}	10^{-6}	No (default 2.2×10^{-9})
`gtol`	10^{-6}, 10^{-4}, 10^{-2}	10^{-5}	Yes
`maxcor`	5, 10, 20	10	Yes
Gradient estimation	–	Finite differences	Yes
Bounds	–	Enabled (full ranges)	–

We include DreamerV3 (Hafner et al., 2024) as a representative modern model-based reinforcement learning baseline. DreamerV3 learns a latent "world model" from experience and optimizes a policy by imagining trajectories in that latent space. In our setting, each environment step corresponds to a full measurement cycle (Section 3), i.e., acquiring all neighbor-pair CSDs, updating the virtualization estimate, then applying gate updates. DreamerV3 models the predictive distribution over next-step observations and rewards through a recurrent state-space model (RSSM) operating on a learned latent state z_t. In our case, the transition captures the effect of gate updates on subsequent measurement outcomes (i.e., changes in CSD geometry and reward).

Table 7: Bayesian optimization configuration. Kernel, priors and fitting procedure follow BoTorch defaults; no kernel-family search was performed. The sliding-window scaffold (window structure and convergence thresholds) is identical to that used for Nelder–Mead and L-BFGS-B.

Component	Setting
Surrogate	BoTorch `SingleTaskGP` (exact GP)
Kernel	BoTorch default with ARD (incl. priors)
Transforms	Inputs normalized to the unit cube; outcomes standardized
Hyperparameter fitting	Exact marginal log-likelihood, refit after every acquisition
Acquisition function	LogEI (qLogEI for batch size > 1)
Acquisition optimization	Multi-start L-BFGS: 10 restarts, 512 raw samples (joint) Multi-start L-BFGS: 5 restarts, 256 raw samples (pairwise)
Per-window budget	20 evaluations per window visit: a 19-point Latin hypercube, then 1 GP-guided acquisition
Hardware	GPU (CUDA)

Table 8: DreamerV3 configuration.

Parameter	Value
Model size preset	12M parameters (`size12m`)
Actor / critic	3-layer MLPs, 256 units; symexp two-hot critic
Optimizer	Learning rate 4×10^{-5}, gradient clipping 0.3, 1000 warm-up steps (defaults)
Batch	16 sequences $\times$ length 32
Replay buffer	5×10^5, uniform sampling, online
Train ratio	32
Imagination	Horizon 15; $\lambda = 0.95$; entropy 3×10^{-4} (defaults)
Environment	Same simulator, observation and action space; 50-step episodes
Training	Separate agent per array size
Tuning	None beyond size preset (defaults, as intended)

D ARRAY TUNING AGENT REWARDS

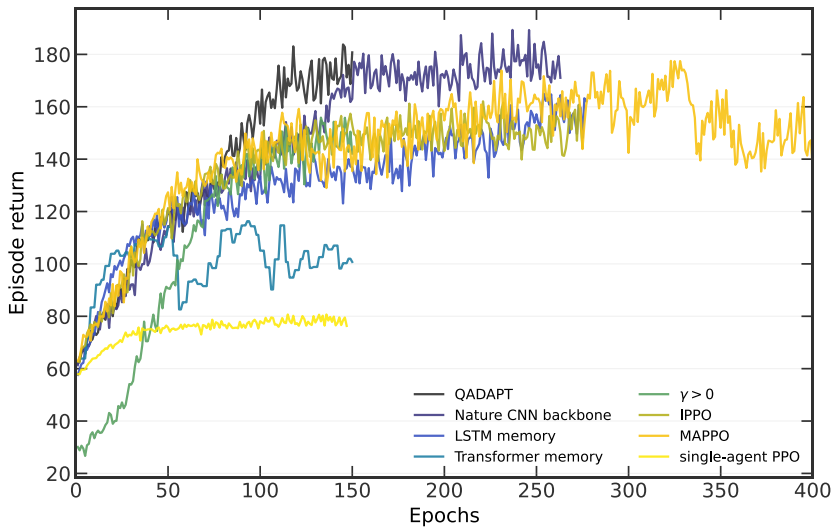


Figure 4: Average per-agent reward for QADAPT trained on a 4-dot array, in addition to each version of the ablated base model, detailed in Section 4. In all benchmarks, trained parameters are always taken from the epoch in which models performed best.

E ARRAY TUNING CONVERGENCE OVER TIME

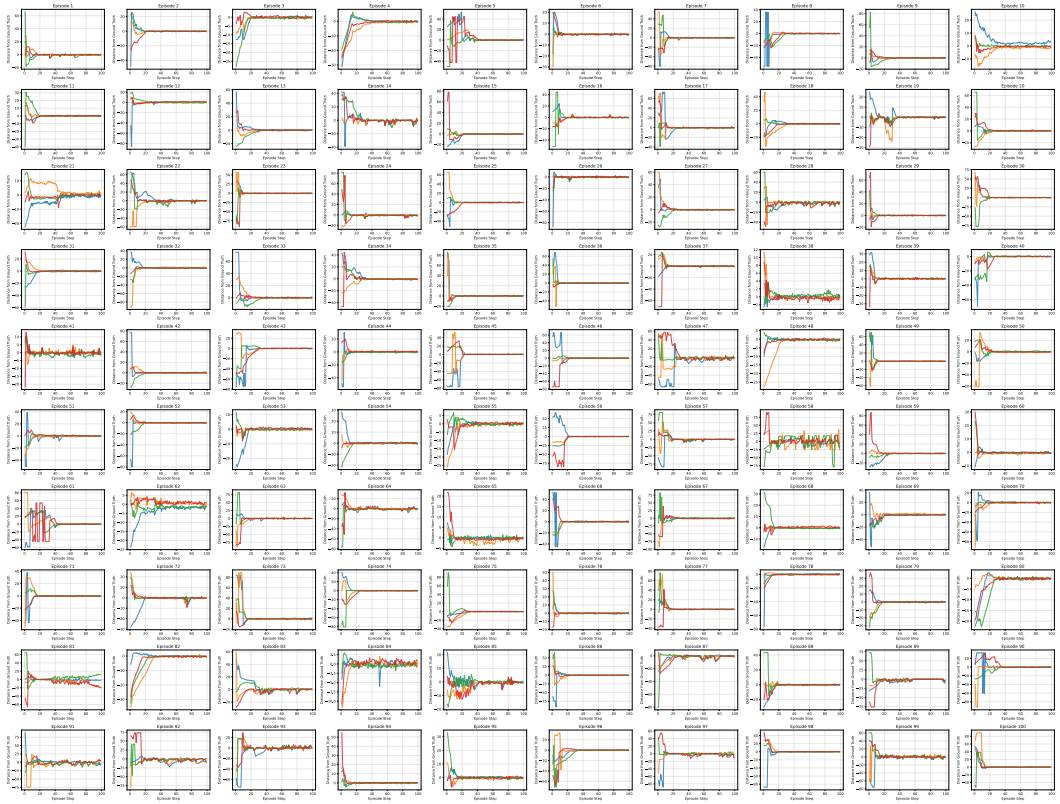


Figure 5: Trained agent behavior sampled over 100 episodes for a 4-dot system. The four colors in each graph represent the signed distance to the ground truth position in voltage space for each of the plunger gate agents.

F TRAINING HYPERPARAMETERS AND COMPUTE RESOURCES

Table 9: Hyperparameters for PPO.

Hyperparameter	Value
Batch size	16384
Grad clip	40
Minibatch size	2048
Epochs	10
Optimizer	Adam
Learning rate	3e-5
γ	0
λ	0
Clip	0.2
Entropy coefficient	0.01
Value function loss coeff	0.5
KL target	0.01
Predicted log std clip bounds	(-5, 2)

G CAPACITANCE EXTRACTION CNN

$f_{\text{virt}}(\cdot)$ is a MobileNetV3-Small (Howard et al., 2019) with two small MLP heads outputting three local cross-capacitances and their log-variances. Consider a four-dot system in which a CSD is acquired by sweeping the voltage on the middle two plungers (P_2 vs. P_3). The cross-capacitance predictions are between: $\mathrm{P}_2 \leftrightarrow \mathrm{d}_3 = \mathrm{P}_3 \leftrightarrow \mathrm{d}_2$, $\mathrm{P}_3 \leftrightarrow \mathrm{d}_1$ and $\mathrm{P}_2 \leftrightarrow \mathrm{d}_4$. Acquiring such predictions for any sized array enables the block-diagonal $D \times D$ matrix to be populated, with elements at all $(\mathrm{d}_\mathrm{x}, \mathrm{d}_{\mathrm{x}\pm1})$ and $(\mathrm{d}_\mathrm{x}, \mathrm{d}_{\mathrm{x}\pm2})$ indices. All on-diagonal elements are assumed to be one. Training scans are generated with the same simulator and parameter ranges as the tuning environment (Appendix B), except that cross-capacitances are sampled symmetrically about zero, so the model also learns to identify negative (overcompensated) couplings, enabling Φ to self-correct online. Training curves are shown in Fig. 6; we deploy the epoch-8 checkpoint, after which validation loss does not improve.

Table 10: Hyperparameters for the capacitance extraction CNN.

Hyperparameter	Value
Input CSD size	100×100
Parameters	1.3M
Dataset size (CSDs)	750k
Train/validation split	90/10
Optimizer	Adam
Learning rate	5e-4
Batch size	64
MSE loss weight	10.0
NLL loss weight	0.1

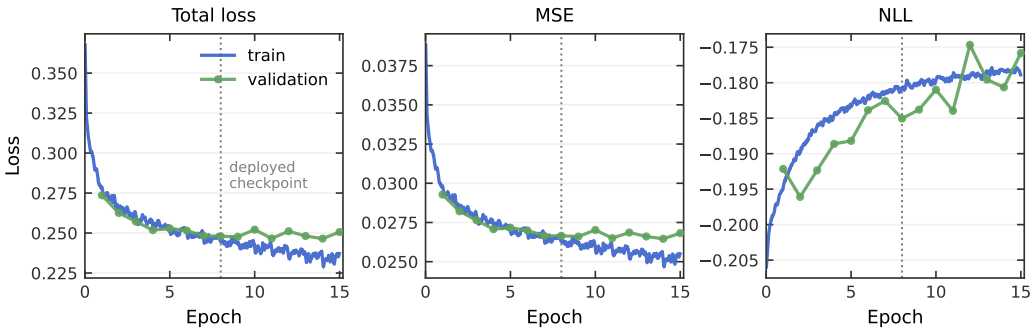


Figure 6: **Training Curves of the Capacitance Extraction CNN**. Total loss (left) is a weighted sum of MSE on capacitance predictions (center) and negative log-likelihood on variance estimates (right), shown unweighted. The dotted line marks the deployed epoch-8 checkpoint.

To quantify how virtualization accuracy affects tuning, we replace the Kalman filter's observations with ground-truth cross-capacitances plus an injected error, and sweep its magnitude together with the filter's process noise (Fig. 7). Convergence stays at or above 84% for errors up to half the training C_{gd} range at every process-noise setting, and at or above 86% for any error when the process noise is at most 10^{-3}. Performance degrades only when both the injected error and filter's process noise are large, indicating performance robustness. By comparison, disabling the filter altogether reduces convergence at the same radius to 10% (Table 1), even though the error of $f_{\text{virt}}(\cdot)$ is only ≈ 0.09. This concludes that temporal filtering applied to virtualization works constructively to achieve the QADATP's high performance. In cases where capacitances are expected to vary over a tuning run, this experiment also shown that the process noise can be increased with minimal effect on overall performance.

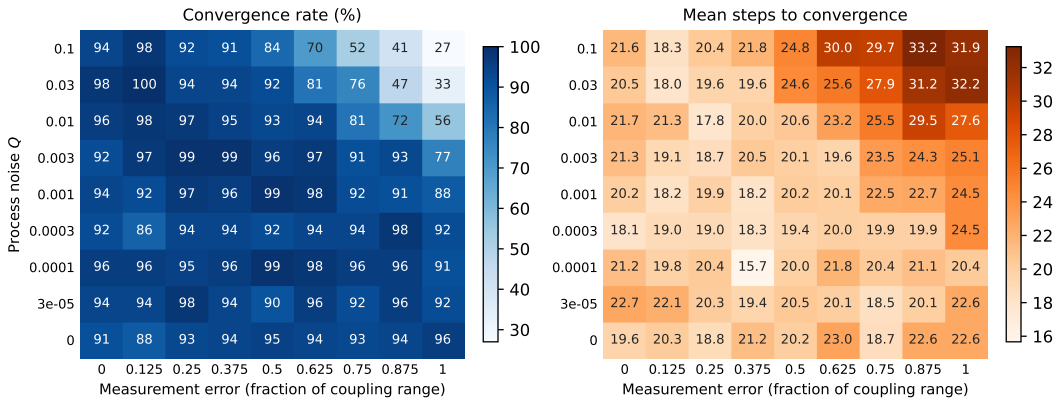


Figure 7: **Sensitivity to Virtualization Accuracy.** Convergence rate (left) and mean steps to convergence (right) of the trained 4-dot policy when the Kalman filter's observations are replaced by ground-truth cross-capacitances plus an injected error (a fixed offset equal to +- this value), given as a fraction of the training C_{gd} range (Appendix B), for a range of filter process-noise settings. Each cell averages 100 tuning trials, with convergence defined by the 2% radius of Table 1.

H EXTENSION TO SUPERCONDUCTING QUBITS

In this appendix, we apply QADAPT to the tuning of superconduting qubits in the presence of simultaneous drives. We largely follow the formulation of the control problem for flux-tunable transmon qubits, as laid out in Wesdorp et al. (2026); however, unlike in that work, we do not rely on analytical formulae and also extend the number of variables that are available for tuning.

Conventionally, the tune-up of superconducting qubits is based on a series of experiments, each of which targets one or more tunable parameters to varying levels of precision. A good overview is provided in Gao et al. (2021) and its references. Here, we consider five key parameters: bare qubit frequency, drive frequency, drive amplitude, phase, and a corrective envelope parameter (see Appendix I for full details). We probe the quality of qubits in response to these parameters using just one type of simulated experiment, known as an all-XY calibration sequence (Reed, 2013). In this protocol, 21 unique combinations of single-qubit gates are applied, and the resulting qubit state is measured. The result across all 21 operations ideally forms a "staircase" pattern where the qubit remains either entirely in the ground state, becomes fully excited, or enters into a perfect superposition of ground and excited states. The suboptimal calibration of any one parameter can lead to a deviation from the ideal staircase pattern.

The sequential optimization of transmon qubit parameters does not guarantee good performance in a quantum processor due to the presence of drive-crosstalk. Furthermore, while individual parameter miscalibrations can lead to qualitatively different error syndromes, the all-XY output can produce degenerate signatures when multiple parameters are far from their optimal configuration. These two effects together make transmon qubit tuning a challenge control problem, yet whose physical structure lends itself naturally to QADAPT.

We apply QADAPT to the transmon qubit tuning problem with minor modifications. Inputs, rather than being charge stability diagrams (CSDs) are now vectors of length 21. Actions are qubit parameter updates, which are split among two agent types, realized via shared policies – one type controls $\{\omega_{01}, \omega_d, \phi\}$ and the other controls $\{\Omega, \beta\}$. Refer to Appendix I for parameter descriptions and Appendix J for their ranges. This parameter split is motivated by the physical similarities in the errors these parameters produce. For simplicity, we factor the action space directly based on the coupling between parameter adjustments and all-XY outputs (see Appendix L). The resulting cross-talk matrix is refined online at each time step. The tuning results are shown Figs. 8 & 9.

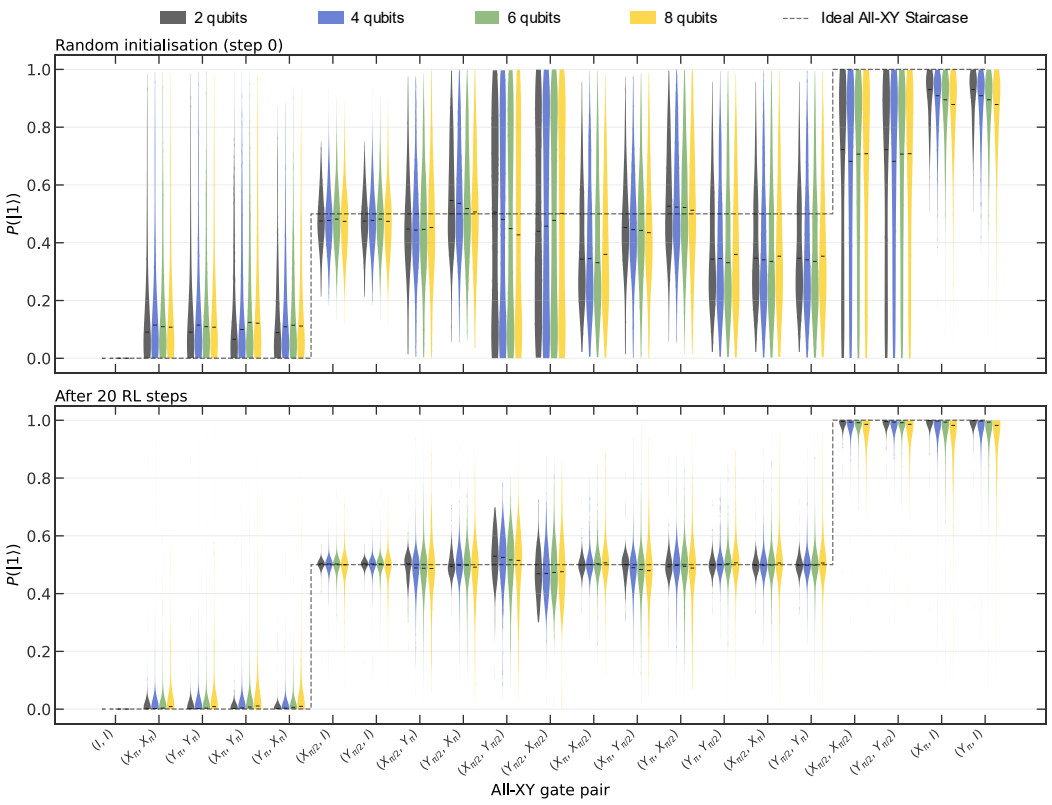


Figure 8: **All-XY Sequence Calibrated using QADAPT.** After random initialization (top), QADAPT reliably tunes superconducting qubit parameters (bare qubit frequency, drive frequency, amplitude, phase, and DRAG coefficient) to near-optimal values in 20 steps (bottom). All violin plots represent distributions over 100 tuning runs, and colors correspond to various qubit counts. The same policy, trained on a four-qubit system, is used across all shown qubit counts.

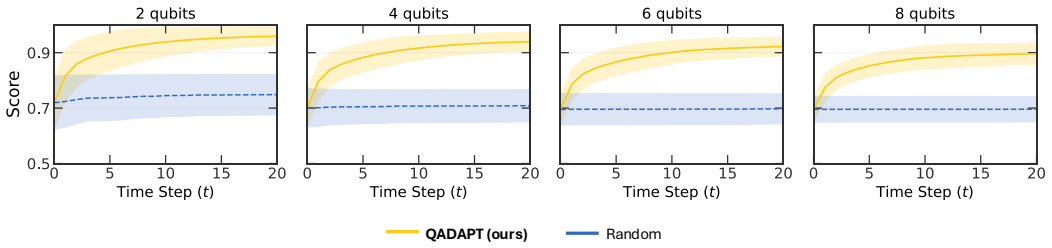


Figure 9: **Convergence of Superconducting Qubit Tuning**. QADAPT generalizes well to the superconducting setting, also scaling zero-shot to qubit numbers of 2, 6 and 8 using a policy trained on 4 qubits. The score is defined as the normalized overlap with the target excited state probability, averaged over qubits and gate-pairs in the all-XY calibration sequence. Lines represent averages over 100 runs, with an envelope of half a standard deviation. Initial parameter configurations are randomized in each run.

I TRANSMON QUBIT HAMILTONIAN

The Hamiltonian of a transmon qubit i in the presence of drive cross-talk from n other, simultaneously driven, qubits, is given by:

$$H = \omega_{01}^i a^\dagger a + \frac{\alpha^i}{2} a^\dagger a^\dagger a a + \mathrm{i} \sum_{j=1}^{n+1} \lambda_{ij} v^j(t)(a^\dagger - a), \tag{5}$$

where $a^\dagger$ and a are the creation and annihilation operators in the subspace of qubit i, ω_{01}^i is the qubit's bare transition frequency, $\alpha^i = \omega_{12}^i - \omega_{01}^i$ is the qubit anharmonicity, λ_{ij} is the relative cross-talk magnitude from qubit j onto qubit i, and $v^j(t)$ is the applied driving field, modulated by complex driving envelopes according to

$$v^j(t) = s_I^j(t)\cos\left(\omega_d^j t + \phi^j\right) + s_Q^j(t)\sin\left(\omega_d^j t + \phi^j\right) \qquad \forall \quad j \in \{0, ..., n\}. \tag{6}$$

Here, ω_d^j and ϕ^j denote the angular drive frequency and phase on qubit j, and $s_I^j(t)$ and $s_Q^j(t)$ are time-dependent in-phase and quadrature drive envelopes. We employ raised-cosine pulse envelopes and standard Derivative Reduction by Adiabatic Gate (DRAG) correction, which relate s_I^j and s_Q^j in the following way,

$$s_I^j(t) = \frac{\Omega^j}{2}\left[1 - \cos\left(\frac{2\pi t}{t_g}\right)\right], \tag{7}$$

$$s_Q^j(t) = -\left(\frac{\beta^j}{\alpha^j}\right)\frac{\mathrm{d}s_I^j}{\mathrm{d}t} = -\left(\frac{\beta^j}{\alpha^j}\right)\cdot\frac{\Omega^j \pi}{t_g}\sin\left(\frac{2\pi t}{t_g}\right), \tag{8}$$

where Ω^j is the nominal Rabi rate in $\mathrm{rad/ns}$, t_g is the gate time, and β^j is the DRAG coefficient.

I.1 HARDWARE EFFECTS

Experimentally, the drive amplitude in volts V is related to the Rabi rate via $\Omega^j = \eta^j \cdot V$, where η^j is the capacitive coupling strength between the transmon and its drive line. This coupling is predominantly determined by device geometry, which is fixed at fabrication, and pre-calibrated in earlier tuning stages. However, we account for imperfect calibration, or changes in line attenuation between qubits, by introducing a prefactor to Ω^j. Adjustments to the raw Ω^j by our algorithm should therefore be interpreted as corrective updates to the voltage amplitude of the qubit drive's carrier envelope.

Similarly, we introduce group and phase delays to mimic the behavior of pulses propagating through real cables, which in general are also unique for each qubit. In summary, this transforms t, ϕ^j and Ω^j parameters appearing in Eqs. (6–8) as follows,

$$t \to t + t_\delta^j, \tag{9}$$

$$\phi^j \to \phi^j + \phi_\delta^j, \tag{10}$$

$$\Omega^j \to A^j \Omega^j, \tag{11}$$

and prevents agent policies from collapsing toward trivial parameter settings. All parameter ranges used for training are summarized in Table. 11

We further note that, in practice, the transmon qubit frequency is tuned via an applied DC current I_{bias}^j through an on-chip bias line. This current produces a flux equal to $\Phi^j = M \cdot I_{\mathrm{bias}}^j$, which scales ω_{01}^j according to Koch et al. (2007),

$$\omega_{01}^j(\Phi^j) \propto \left|\cos\left(\frac{\pi\Phi^j}{\Phi_0}\right)\right|^{\frac{1}{2}}, \tag{12}$$

where $\Phi_0 = h/2e$ is the flux quantum. By applying parameter updates onto ω_{01}^j directly, phase accumulation during the simulated gate becomes linearized, which better conditions the action space for agent learning. The characterization of ω_{01}^j with respect to I_{bias}^j is routinely done in experimental

settings, allowing a direct mapping to be made between the two variables at deployment. For fixed-frequency qubits, parameter updates to ω_{01} would drop out from the tuning algorithm.

The effect of flux noise and flux cross-talk between qubits, as well as associated changes in the anharmonicity α^j caused by second order effects upon varying Φ^j, are outside the scope of this work. We further neglect the weak dependence of the Rabi rate Ω^j on the flux operating point. A more complete treatment would explicit parametrize both α^j and A^j as functions of ω_{01}^j, which could be pre-calibrated via Rabi experiments at multiple flux operating points. However, since our algorithm treats each round of all-XY measurements independently, with no temporal credit assignment, the new effective Rabi rate would become implicitly absorbed into A^j in a real experiment for each new operating point. The inability to track flux-dependent dynamics is therefore not a limitation of the model, but rather a consequence of the immediate-feedback reward structure.

J Transmon Qubit Simulation Parameters

Table 11: Parameter ranges used in transmon qubit simulations. When a min and a max value are specified, we sample uniformly at random between those limits at each reset of the environment. For values sampled from a normal distribution, the mean and standard deviation are indicated.

Parameter	Min	Max	Value
Number of qubits	-	-	4
Fock space truncation	-	-	3
$\omega_{01}/2\pi$ (GHz)	4	6	-
$\alpha/2\pi$ (MHz)	-300	-200	-
$\lambda_{ii'}$	0.0	0.5	-
t_g (ns)	16	24	-
$\omega_d/2\pi$ (GHz)	-	-	$\sim \mathcal{N}(\omega_{01}/2\pi, 0.01)$
ϕ (rad)	$-\pi$	π	-
Ω (rad/ns)	-	-	$2\pi/t_g(1+\delta), \quad \delta \sim \mathcal{N}(0, 0.1)$
β	0.0	1.5	-
t_δ (ns)	-2	2	-
ϕ_δ (rad)	$-\pi$	π	-
A	0.85	1.15	-

Table 12: Parameter bounds for transmon qubit simulations. Min and max values are stated relative to their initialization at the start of each training episode.

Parameter	Min	Max
$\omega_{01}/2\pi$ (MHz)	-150	0
$\omega_d/2\pi$ (MHz)	$\omega_{01} - 20$	$\omega_{01} + 20$
ϕ (rad)	$-\pi$	π
Ω (rad/ns)	$0.8 \times 2\pi/t_g$	$1.2 \times 2\pi/t_g$
β	0.0	1.5

K ALL-XY CALIBRATION SEQUENCE

The 21 gate pairs that comprise the all-XY calibration sequence are provided in Table 13 together with their ideal excited state probabilities. The drive amplitude Ω^j enters the Hamiltonian linearly, such that the resulting rotation angle on qubit j scales proportionally with Ω^j. Likewise, the drive phase ϕ^j enters the Hamiltonian as a rigid rotation of the drive vector in the IQ-plane, and so the axis of rotation is determined directly by ϕ^j, independently of the rotation angle. As a result, all gates in the calibration sequence can be expressed using amplitude scalings and/or phase offsets relative to a X_π base pulse. In real hardware, phase offsets can implemented at no extra cost using programmable 'virtual' phases.

Table 13: Gate pairs participating in the all-XY calibration sequence. The qubit is assumed to be in the ground state $|0\rangle$ at the start of each gate-pair. Amplitude scalings and phase offsets are stated relative to a perfectly calibrated X_π gate pulse.

Index	Gate pair	Amplitude scaling	Phase offset	$P(\|1\rangle)$
1	I, I	0.0, 0.0	0, 0	0.0
2	X_π, X_π	1.0, 1.0	0, 0	0.0
3	Y_π, Y_π	1.0, 1.0	$\frac{\pi}{2}$, $\frac{\pi}{2}$	0.0
4	X_π, Y_π	1.0, 1.0	0, $\frac{\pi}{2}$	0.0
5	Y_π, X_π	1.0, 1.0	$\frac{\pi}{2}$, 0	0.0
6	$X_{\frac{\pi}{2}}, I$	0.5, 0.0	0, 0	0.5
7	$Y_{\frac{\pi}{2}}, I$	0.5, 0.0	$\frac{\pi}{2}$, 0	0.5
8	$X_{\frac{\pi}{2}}, Y_\pi$	0.5, 1.0	0, $\frac{\pi}{2}$	0.5
9	$Y_{\frac{\pi}{2}}, X_\pi$	0.5, 1.0	$\frac{\pi}{2}$, 0	0.5
10	$X_{\frac{\pi}{2}}, Y_{\frac{\pi}{2}}$	0.5, 0.5	0, $\frac{\pi}{2}$	0.5
11	$Y_{\frac{\pi}{2}}, X_{\frac{\pi}{2}}$	0.5, 0.5	$\frac{\pi}{2}$, 0	0.5
12	$X_\pi, X_{\frac{\pi}{2}}$	1.0, 0.5	0, 0	0.5
13	$X_\pi, Y_{\frac{\pi}{2}}$	1.0, 0.5	0, $\frac{\pi}{2}$	0.5
14	$Y_\pi, X_{\frac{\pi}{2}}$	1.0, 0.5	$\frac{\pi}{2}$, 0	0.5
15	$Y_\pi, Y_{\frac{\pi}{2}}$	1.0, 0.5	$\frac{\pi}{2}$, $\frac{\pi}{2}$	0.5
16	$X_{\frac{\pi}{2}}, X_\pi,$	0.5, 1.0	0, 0	0.5
17	$Y_{\frac{\pi}{2}}, Y_\pi,$	0.5, 1.0	$\frac{\pi}{2}$, $\frac{\pi}{2}$	0.5
18	$X_{\frac{\pi}{2}}, X_{\frac{\pi}{2}}$	0.5, 0.5	0, 0	1.0
19	$Y_{\frac{\pi}{2}}, Y_{\frac{\pi}{2}}$	0.5, 0.5	$\frac{\pi}{2}$, $\frac{\pi}{2}$	1.0
20	X_π, I	1.0, 0.0	0, 0	1.0
21	Y_π, I	1.0, 0.0	$\frac{\pi}{2}$, 0	1.0

L Implementation Details of Qubit Dynamics

All transmon qubit simulations are performed by solving the time-dependent Schrödinger equation under the Hamiltonian in 5 using `dynamiqs` (Guilmin et al., 2025), a JAX-based quantum dynamics library. We rely heavily on the library's batching functionality to reduce compute time: all $N_{\text{QUBITS}} \times N_{\text{AllXY}} = 4 \times 21$ independent quantum systems are evolved simultaneously within a single `sesolve` call by constructing a batched Hamiltonian of shape $(N_{\text{QUBITS}} \times N_{\text{AllXY}} \times 3 \times 3)$, avoiding any Python-level loop over sequences or qubits.

Each time-dependent drive is implemented as a scalar coefficient $f(t)$ that modulates the Hamiltonian operator $\mathrm{i}(a^\dagger - a)$. Cross-talk between drive lines is incorporated via a linear transformation of the per-qubit coefficients at each time step. f is passed as a Python callable to `dq.modulated`, which `dynamiqs` evaluates on demand at each step taken by the adaptive ODE solver. To account for per-qubit timing offsets t_δ^j, the integration window is expanded beyond $[0, 2t_g]$ to capture both earliest and latest arriving pulses, while each qubit's envelope is still evaluated in its own shifted time coordinate $\tau^j = t - t_\delta^j$, ensuring no pulse is truncated regardless of the sign of the delay.

The full Jacobian of All-XY staircase outcomes with respect to all qubit control parameters is of shape $(N_{\text{QUBITS}} \times N_{\text{AllXY}} \times N_{\text{QUBITS}} \times N_{\text{params}})$, and is also computed in a single forward pass using `jax.jacfwd`, which leverages `dynamiqs`' forward sensitivity mode `dq.gradient.Forward`. This mode offer a speed advantage when the number of parameters is fewer than the number of outputs, which is true in our case. The resulting Jacobian blocks are used to construct the cross-talk compensation tensor via pseudoinversion.

M Compute Resources and Cost

We iterated on various modifications of QADAPT on a cluster of NVIDIA RTX A4000 GPUs with 20 GB of GPU memory. All our hero runs were performed on NVIDIA 8xH100 GPUs with 80 GB of GPU memory. The cost of these hero runs add up to 2500 USD on the modal platform.

N Code Availability

All the code for QADAPT is available through the anonymized url: `https://anonymous.4open.science/r/rl-agent-for-qubit-array-tuning-71D1/README.md`. We present the modal entry points to run this code as well.