Note S2 | VQE/VQA circuit configuration (ansatz, optimizer, decoding) for MXene EIS fitting

This work implements a continuous "variational quantum-assisted" (VQE/VQA) branch to fit the MXene EIS spectrum by directly minimizing the complex-domain discrepancy between the measured impedance and the circuit model. The key idea is to treat the circuit parameters $\theta = \{R_s, L, R_{ct}, Q_1, \alpha_1, Q_2, \alpha_2\}$ as a bounded continuous vector encoded by a variational circuit. The circuit outputs a latent vector that is deterministically decoded into physically meaningful parameters, and a classical outer-loop optimizer updates the circuit angles to minimize the objective.

S2.1 Qubit allocation and latent parameterization.

We use **7 qubits**, one per circuit parameter, so the quantum register dimension matches the model dimension. The variational parameters (rotation angles) are denoted Θ, while the decoded physical parameters are θ. The circuit produces expectation values $\langle Z_i \rangle \in [-1,1]$ (or equivalently measurement probabilities), which are mapped to normalized coordinates $u_i \in [0,1]$ via:

$$u_i = \frac{1+\langle Z_i \rangle}{2} \tag{S6}$$

This definition is hardware-agnostic and can be realized by measuring each qubit in the computational basis and using $u_i \approx p_i(1)$, where $p_i(1)$ is the measured probability of outcome "1" on qubit i. In practice, we use shots = 4096 per evaluation to estimate u with stable sampling noise.

S2.2 Ansatz (HEA-style) and circuit depth.

The variational circuit follows a hardware-efficient ansatz (HEA) tailored for near-term devices: repeated blocks of single-qubit rotations plus a fixed entangling pattern. For each layer ℓ, we apply parameterized $R_y(\Theta_{\ell,i})$ rotations on all qubits, followed by an entangling stage (e.g., nearest-neighbor CZ or CNOT chain). The circuit depth is controlled by the number of repetitions (**reps ≈ 3**), balancing expressivity and trainability. With 7 qubits and three layers, the ansatz is expressive enough to represent correlated parameter updates (important because EIS parameters are not independent), while keeping the gate count modest to limit noise accumulation and barren-plateau risk.

S2.3 Bounded decoding to physical parameter space.

The VQE/VQA branch outputs $u \in [0,1]^7$, which is decoded into θ using the same bound map as **Table S1**. Parameters spanning orders of magnitude (e.g., L, Q_1, Q_2) are decoded in log space, while resistances and exponents use linear decoding:

$$p(u) = p_{min} + u\ (p_{max} - p_{min})\ (linear), \tag{S7}$$

$$p(u) = 10^{log_{10}(p_{min}) + u[log_{10}(p_{max}) - log_{10}(p_{min})]}\ \text{(log)} \tag{S8}$$

This bounded decoding guarantees feasibility (e.g., $R_s > 0,\ 0 < \alpha \leq 1$) and prevents numerical pathologies caused by unphysical parameter excursions. Importantly, the same decoding is used across classical NLLS and QAOA branches, ensuring fair comparisons.

S2.4 Objective function and classical outer-loop optimizer.

At each iteration, the decoded θ is inserted into the EIS model $Z_{model}(\omega)$ and the loss is evaluated as complex-domain SSE:

$$\mathcal{L}(\theta) = \sum_{k=1}^{N} |Z_{model}(\omega_k; \theta) - Z_{data}(\omega_k; \theta)|^2 \tag{S9}$$

We optimize Θ using COBYLA (derivative-free), which is robust under shot noise and avoids unstable finite-difference gradients. COBYLA is well-suited here because the objective is non-convex and can be noisy due to sampling. The optimization terminates after a fixed evaluation budget (or when improvements fall below tolerance), and the best-achieved θ is recorded for Nyquist/Bode overlays and subsequent identifiability/robustness analyses.

S2.5 Practical considerations (stability and interpretability).

Two stability measures are central: (i) bounded decoding ensures every circuit evaluation corresponds to a valid circuit model; (ii) a moderate ansatz depth (reps $\approx$ 3) helps maintain trainability without requiring deep circuits. Because u is directly interpretable as a normalized parameter vector, the VQE/VQA branch yields transparent parameter trajectories and allows direct comparison with discrete QAOA outputs (decoded in the same bounded framework).