Quantum Science and Technology

QDesignOptimizer based on ANMod: an automated, physics-guided, multi-parameter design optimizer for superconducting quantum devices

Quantum Science and Technology

OPEN ACCESS

RECEIVED
3 November 2025

REVISED
2 June 2026

ACCEPTED FOR PUBLICATION
9 June 2026

PUBLISHED
23 June 2026

PAPER

QDesignOptimizer based on ANMod: an automated, physics-guided, multi-parameter design optimizer for superconducting quantum devices

Axel M Eriksson[1,*], **Lukas J Splitthoff**[1], **Harsh Vardhan Upadhyay**[1], **Pietro Campana**[1,2], **Oscar Lundström**[3], **Niranjan Pittan Narendiran**[1], **Kunal Helambe**[1], **Linus Andersson**[1] and **Simone Gasparinetti**[1,*]

[1] Department of Microtechnology and Nanoscience, Chalmers University of Technology, 412 96 Gothenburg, Sweden
[2] Department of Physics, University of Milano-Bicocca, 20126 Milan, Italy
[3] Plain Rocks AB, 413 26 Gothenburg, Sweden
[*] Authors to whom any correspondence should be addressed.

E-mail: axel.eriksson@chalmers.se and simoneg@chalmers.se

Keywords: automated design, quantum circuits, energy participation ratio, HFSS, ANMod, nonlinear optimization, superconducting circuits

Abstract

Designing superconducting quantum circuits involves optimizing the layout to achieve certain target parameters. This optimization process usually depends on iterative electromagnetic simulations, which are computationally expensive and require manual intervention to adjust the layout parameters. Here, we present a method to efficiently automate the optimization of superconducting circuits, which significantly reduces the need for manual intervention. The method's efficiency arises from approximate nonlinear model-driven (ANMod) parameter updates, which are constructed from the user's physical knowledge. Additionally, we provide a full implementation using the ANMod-method as an open-source Python package, QDesignOptimizer. The package automates the design workflow by combining high-accuracy electromagnetic simulations in ansys HFSS and energy participation ratio (pyEPR) analysis integrated with the design tool quantum-metal (formerly known as Qiskit-Metal). Our implementation supports modular and flexible subsystem-level analysis and is easily extensible to optimize for additional parameters. The ANMod-method is not specific to superconducting circuits; as such, it can be applied to a range of nonlinear optimization problems across science and technology.

1. Introduction

Nonlinear superconducting circuits, built around Josephson junctions, form the backbone of superconducting quantum processors [1–4] and parametric amplifiers [5]. The process of translating an initial concept of a superconducting circuit (*circuit-level design*) into a finalized physical implementation (which requires a *layout-level design*) is inherently complex, iterative, and time-consuming [2, 6, 7].

The circuit design process usually begins with a user-defined circuit Hamiltonian, which determines all target physical quantities, such as mode frequencies, nonlinear couplings and decay rates, which lead to the intended functionality. The target circuit is then mapped onto a physical layout composed of lumped and distributed circuit elements. Common design platforms for this task include quantum-metal (formerly known as Qiskit-Metal) [8], IQM's KQCircuits [9], and gdsfactory [10].

For initial approximate circuit designs, lumped-element circuits are efficiently analyzed using simplified circuit models and fast component-wise simulations [11–15]. However, accounting for the distributed nature of components requires more sophisticated eigenmode, capacitance, or scattering simulations, for example using open-source [16, 17] or commercial simulation software [18, 19]. These

simulations are time-consuming and computationally intensive but provide high predictive power, especially needed for distributed circuits. To converge on a final design, some update strategy must be applied which iteratively brings the design closer to the target. The update strategy is often manual, which slows down the design process.

To accelerate the design process, many research groups working on superconducting circuits have established their internal workflows for layouts and simulations, often tailored to their device scale and complexity. One notable effort towards open-source optimization of physical layouts has led to the automation of superconducting coupler architecture designs [20]. Another, open-source contribution is the *SQuADDS* database of superconducting quantum devices serving as a starting point for customized devices [21]. However, these contributions are limited to specific circuit geometries, which have been studied or pre-simulated making them difficult to extend to novel designs.

Here, we present a method which automates the manual update step through an efficient, physics-guided, multi-parameter optimization framework. The user supplies their physical understanding of the circuit as an approximate custom nonlinear model to guide design variables updates. The framework automates the optimization for arbitrary device geometries, naturally incorporating parameter inter-dependencies and supporting easy extension to new models and targets. Our method integrates with quantum-metal-based designs [8] and high-accuracy electromagnetic simulations in ansys HFSS [18], which are further analyzed with energy-participation ratio studies in pyEPR [22]. The underlying method of solving an approximate nonlinear model for fast convergence can be applied to any computationally heavy, multidimensional optimization task, making it relevant for a vast range of engineering problems. The optimization is feasible under two main conditions, the first one being that the user has an approximate understanding of the dominant relations between design variables and target parameters, and that simulations or measurements can be executed to obtain information of what parameters the current design results in. We will refer to this approach as the **A**pproximate **N**onlinear **Mo**del-**d**riven (**ANMod**) optimization method.

We provide the full implementation of the optimizer as an open-source Python package QDesignOptimizer [23], which offers a flexible and modular structure for efficient subsystem optimizations. Simulation routines and analytical models are dynamically compiled on the basis of user-defined settings. To demonstrate the practical applicability of our package, we include several examples for single and two-qubit chip design optimizations.

2. The ANMod-optimization method

The workflow of our ANMod-optimization framework, depicted in figure 1, consists of the problem formation step, the actual optimization step, and the validation step. In the problem formulation step, the user defines (i) the target parameters, (ii) an initial guess for values of the design variables, and (iii) the approximate physical relations as a multi-parameter nonlinear model between the parameters and design variables. In the optimization step, the set of design variables is iteratively updated based on the solution of the approximate nonlinear model. As such, ANMod-method has methodological similarities with so called surrogate models as discussed in appendix G. In every iteration, the current set of parameters is computed from simulations and further analysis. In the validation step, the result is finally manually inspected for convergence and general feasibility. In the following subsections, we introduce the general mathematical framework of the ANMod-method. In the next section we introduce the QDesignOptimizer which is a concrete integration with quantum-metal, HFSS and energy participation ratio (EPR)-analysis.

2.1. Problem formulation

The problem formulation begins with the definition of N design variables $\overrightarrow{x} = \{x_1, \ldots, x_N\}$, which are varied to reach the targets of the N parameters $\overrightarrow{y} = \{y_1, \ldots, y_N\}$. Furthermore, the physical system is described by

$$y_i \propto f_i^{\text{exact}}\left(\underset{\text{excl.}y_i}{\overrightarrow{y}}, \overrightarrow{x}\right) \approx f_i\left(\underset{\text{excl.}y_i}{\overrightarrow{y}}, \overrightarrow{x}\right) \tag{1}$$

specifying the exact proportionalities f_i^{exact} between each parameter and the design variables as well as (other) parameters in the exact relation. From here on, we omit the subscript 'excl. y_i' for brevity. The

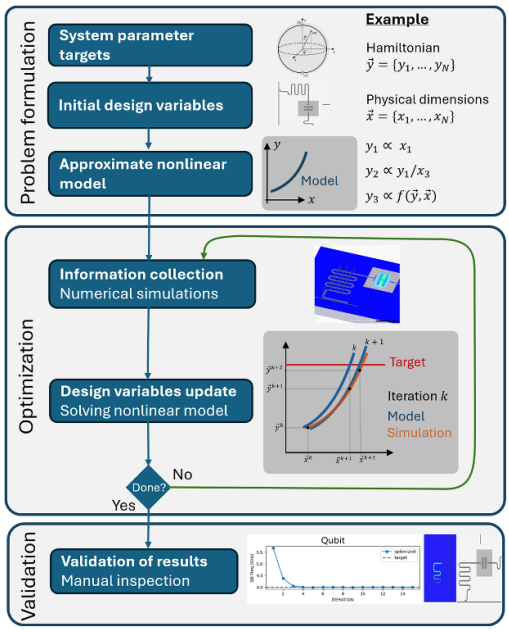

Figure 1. Design flow via the ANMod-optimization method consisting of problem formulation, optimization and validation. In the problem formulation, the user specifies the system target parameters and initial values of the design variables as well as the nonlinear relationship between them. The optimization starts by a numerical simulation using the current design variables $\vec{x}^k$ to study which parameter values $\vec{y}^k$ they correspond to. The design variables are then updated by solving the nonlinear model evaluated with the currently available information, resulting in the updated values x^{k+1}. The numerical simulation runs again, to predict y^{k+1} providing a point closer to the target at which the model can be evaluated. Finally, the user manually inspects the convergence, mode shapes and resulting design to validate that it fulfills any additional constraint not explicitly formulated.

exact relations are generally unknown but can be approximated by the nonlinear models f_i built from the user's knowledge about the physical system.

The framework enables the user to easily change both the physical relationships and the dimensionality of the problem by selecting which parameters to optimize. The implementation requires each design variable to be associated with one target parameter such that if the parameter is unselected, the number of parameters and design variables remains equal. In complex systems, the model refinement could potentially be guided by an empirical numerical study of the system's behavior. Finally, as an initial guess for the optimization, the user has to provide sensible values for the design variables $\vec{x}$. During optimization, constraints can be imposed on the design variables to prevent structures of different modes to geometrically overlap.

2.2. Information collection—simulate design variables

To obtain the parameter values $\vec{y}^k$ for the current set of design variables $\vec{x}^k$, the optimization framework needs to run some detailed simulations and further analysis (or potentially measure physical devices). In the QDesignOptimizer, detailed electro-magnetic eigenmode or capacitance simulations are run in ansys HFSS, and followed by an EPR analysis. The parameters $\vec{y}^k$, in the kth iteration, are then extracted from the simulation and analysis results.

2.3. Update design variables

Here, we present the core idea of the ANMod-optimization method. To reach the parameter targets through the optimization, we aim to efficiently update the design variables from the ones used in the simulation $\vec{x}^k$ in iteration k to $\vec{x}^{k+1}$, such that $\vec{y}^{k+1}$ gets close to the target value $\vec{y}^{\text{target}}$. Given the proportionality between y_i and f_i, the parameter corresponding to the updated design variable can be estimated by

$$\tilde{y}_i^{k+1} = y_i^k \frac{f_i\left(\vec{\tilde{y}}^{k+1}, \vec{x}^{k+1}\right)}{f_i\left(\vec{y}^k, \vec{x}^k\right)}. \tag{2}$$

Here, we distinguish the quantity $\vec{\tilde{y}}$ (with $\tilde{}$) estimated by the approximate model (equation (1)) and the simulated quantity $\vec{y}$, which typically is more accurate. Hence, the more accurate the nonlinear

model set in equation (1), the smaller is the discrepancy $||\vec{y} - \vec{\tilde{y}}||$ and the faster the optimization converges. To obtain the updated design variables $\vec{x}^{k+1}$, we minimize the cost function

$$C = \sum_{i=1}^{N} \left| \frac{\tilde{y}_i^{k+1}}{y_i^{\text{target}}} - 1 \right|^2, \tag{3}$$

by finding the optimal $\vec{x}^{k+1}$. If the problem is correctly formulated, the minimization will reach

$$\tilde{y}_i^{k+1} = y_i^{\text{target}} \tag{4}$$

for all N targets in the optimization. (Note however that user constraints on the design variables might result in a cost $C > 0$). Hence, the closed expression minimized by the ANMod-method in each iteration is

$$C = \sum_{i=1}^{N} \left| \frac{y_i^k}{y_i^{\text{target}}} \frac{f_i\left(\vec{y}^{\text{target}}, \vec{x}^{k+1}\right)}{f_i\left(\vec{y}^k, \vec{x}^k\right)} - 1 \right|^2 \tag{5}$$

which only depends on the three inputs: (i) the values of the parameters in the previous step $\vec{y}^k$, (ii) the target parameter values $\vec{y}^{\text{target}}$, and (iii) the design variables in the previous step $\vec{x}^k$. The optimizer then finds the updated design variables $\vec{x}^{k+1}$ which minimize the cost C. Hence, the step size of the design variables are dynamically evaluated by minimizing equation (5) in each iteration such that, if the model is exact, the optimizer converges to the target in a single *optimal* step. However, in most cases, the nonlinear model is only approximate and taking the full step $\Delta\vec{x}$ as predicted by minimizing C might lead to unstable optimization. Therefore, the QDesignOptimizer implementation supports low-pass filtering of the step $\Delta\vec{x} \to \zeta\Delta\vec{x}$ with a factor ζ to reduce the risk of overshooting at the cost of slower convergence.

Given this ansatz, parameters can be flexibly included or excluded from the optimization by assuming that parameters without specified optimization targets remain unaffected by changes in the design variables, i.e. $\tilde{y}_i^{k+1} = y_i^k$ for $i > N$. This functionality is implemented in the QDesignOptimizer (see next section) to alleviate the need to rewrite physical relations that depend on excluded parameters.

To further highlight the robustness of the method against model inaccuracies, we decompose the model relation and clarify the role of approximation errors. Detailing the model relation in equation (1), the exact proportionality can be expressed as

$$\frac{y_i}{c_i} = f_i^{\text{exact}}\left(\vec{y}, \vec{x}\right) = f_i^{\text{approx}}\left(\vec{y}, \vec{x}\right) f_i^{\text{error}}\left(\vec{y}, \vec{x}\right) \tag{6}$$

where c_i is the proportionality factor and $f_i^{\text{approx}} = f_i$ denotes the approximate model and $f_i^{\text{error}} \approx 1$ represents the associated error factor. Substituting y_i^k in the update rule (equation (2)) with the decomposition (equation (6)) and evaluating the rule at the target $\tilde{y}_i^{k+1} = y_i^{\text{target}}$ yields the following equation for $\vec{x}^{k+1}$,

$$\frac{y_i^{\text{target}}}{c_i} = f_i^{\text{approx}}\left(\vec{y}^{\text{target}}, \vec{x}^{k+1}\right) f_i^{\text{error}}\left(\vec{y}^k, \vec{x}^k\right). \tag{7}$$

Thus, the update rule effectively incorporates the error factor, but evaluated at the current (instead of the target) position. I.e. by this fixed-point-iteration construction, the update rule does not entirely neglect the error factor as the approximate nonlinear relations in equation (1) do. This property enables the ANMod-method to achieve rapid convergence even with coarse approximate models. Furthermore, equation (7) constraints the types and magnitudes of model inaccuracies.

2.4. Design validation

The final step involves a manual inspection of the optimization results to ensure the feasibility of the optimized design. This includes verifying the convergence of the information-collecting simulations as well as the overall optimization towards the target parameters, inspecting the spatial mode shapes to confirm correct localization, and evaluating the resulting design to ensure that it meets any requirements that were not explicitly specified in the optimization targets. Note that the convergence of the optimizer towards the target parameters is not guaranteed in general (see section 3.2).

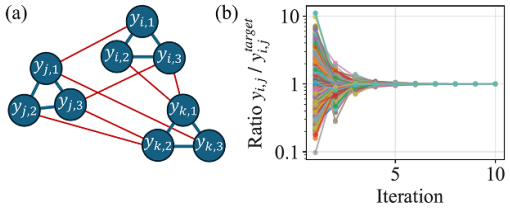

Figure 2. ANMod applied to a high-dimensional test system. (a) Illustration of a nonlinearly coupled cluster system with strong local interactions (blue) within each cluster and weak global interactions (red) between clusters. (b) Convergence trajectory for the 3000 parameters (in different colors) normalized to their targets. The test case consists of 1000 clusters with 3 parameters each. The user-defined approximate model approximately retains the strong local interactions but completely neglects the weak global interactions.

2.5. Scalability of the ANMod-method

The ANMod-method leverages on the ability to approximate the full nonlinear problem with smaller independent problems that can be solved using conventional (e.g. gradient based or derivative-free) methods. To illustrate the efficiency and scalability of the ANMod-method, we apply it to a high-dimensional fictive system (not mimicking a qubit design) with 3000 nonlinearly coupled parameters (see appendix F for details). The system (sketched in figure 2(a)) is characterized by strong local interactions, which are approximately retained in the approximate model, and weaker global interactions, which are neglected. The approximate model should be decomposed into sufficiently small, independent subproblems, such that the solver in the ANMod update step does not itself become the computational bottleneck. Despite the large dimensionality, the ANMod-method converges rapidly in the setting of this example as shown in figure 2(b). The model interactions were randomly re-sampled 100 times (see appendix F); within 10 iterations, 25 realizations fully converged below a relative error of 10^{-3} and 15 diverged. Of the remaining cases, 50 reached the convergence criteria for at least 2990 parameters and the final 10 cases have not yet reached the convergence criteria. Note that the rate and ability to converge depends on the strength of the neglected weak-global interactions in addition to the inaccuracy of the approximate model of the local interactions: if these are sufficiently small, the method remains robust even for very high-dimensional systems.

3. The QDesignOptimizer—application in circuit quantum electrodynamics

The QDesignOptimizer package using the ANMod-method can be applied to a broad range of systems in circuit quantum electrodynamics, such as superconducting qubits, resonators, couplers, and combinations of those. In this section, we demonstrate its usefulness and limitations on two representative examples. We begin with the fundamental building block of superconducting quantum processors—a planar qubit–resonator system—to introduce the central ideas of eigenmode-analysis-based optimization. We then examine how nonlinear models and initial conditions can influence the optimization outcomes. To illustrate the scalability of the method to larger design spaces, we apply the optimizer to a more complex qubit–coupler–qubit device with five modes. Finally, we discuss how modular optimization and capacitance–simulation-based optimization extend the applicability of the QDesignOptimizer. For a detailed code-level tutorial, we refer the reader to the online documentation of our QDesignOptimizer Python package [23].

3.1. Eigenmode-analysis-based optimization

For the optimization of the single qubit–resonator system, the problem formulations follows the ANMod-method outlined in section 2.1. First, a sketch of the system is designed in quantum-metal. Then, the optimizer takes the information summarized in table 1. In the given case, the five target parameters are the resonator frequency, resonator decay rate, qubit frequency, qubit anharmonicity and resonator–qubit dispersive shift. The same number of design variables are introduced, each with an initial value that can be varied to reach the target. Note that depending on the choice of design variables, one design variable might affect several parameters, e.g. w_{qb} is primarily used to set the anharmonicity,

Table 1. Summary of the problem formulation for the single qubit–resonator system. The design variables are resonator length l_{res}, qubit Josephson junction inductance L_{qb}, qubit width w_{qb}, resonator–qubit coupling width $w_{\mathrm{res-qb}}$ and resonator to transmission line coupling length $l_{\mathrm{res-tl}}$. To further detail the model of the dispersive shift, we directly input the parameters anharmonicity α and difference frequency $\Delta = f_{\mathrm{qb}} - f_{\mathrm{res}}$.

Parameter	Symbol	Target	Proportional to	Design variable	Initial value
Resonator frequency	f_{res}	6 GHz	$1/l_{\mathrm{res}}$	l_{res}	7500 μm
Qubit frequency	f_{qb}	4 GHz	$1/\sqrt{L_{\mathrm{qb}} \cdot w_{\mathrm{qb}}}$	L_{qb}	12.1 nH
Anharmonicity	$-\alpha/2\pi$	200 MHz	$1/w_{\mathrm{qb}}$	w_{qb}	400 μm
Dispersive shift	$-\chi_{\mathrm{qb-res}}/2\pi$	1 MHz	$(w_{\mathrm{res-qb}}/w_{\mathrm{qb}})^2 \cdot \alpha/[\Delta(\Delta-\alpha)]$	$w_{\mathrm{res-qb}}$	100 μ
Resonator decay rate	κ_{res}	1 MHz	$l^2_{\mathrm{res-tl}}$	$l_{\mathrm{res-tl}}$	400 μm

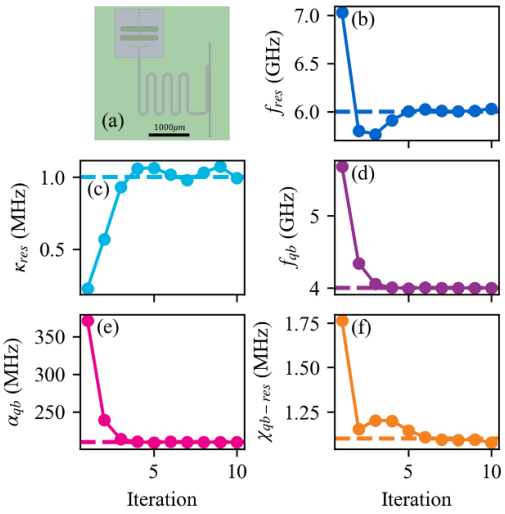

Figure 3. Eigenmode-analysis-based optimization of a qubit–resonator-waveguide system with the problem formulation described in table 1. (a) Screen capture of the simulated structure in ansys HFSS. (b)–(f) Optimization results versus iterations for the five target parameters. The dashed lines indicate the parameter targets. In each iteration, HFSS performed 8 mesh refinement passes thereby yielding approximately equal compute time per iteration.

but it simultaneously affects the qubit frequency (see also appendix E). Finally, an approximate physical relation is provided for each parameter, describing its relationship to the design variables and other system parameters. For instance, the resonator frequency is parameterized by its physical length.

The subsequent optimization follows the iterative approach of information collection and design variable updates described in sections 2.2 and 2.3, respectively. First, the design is simulated using ansys HFSS's eigenmode solver, followed by EPR analysis through pyEPR, both accessed directly via the quantum-metal integration. While the present example uses the integrated HFSS/pyEPR toolchain, the QDesignOptimizer can be extended to other simulation backends to support diverse academic and industrial workflows.

The results for the optimization of the whole subsystem described in table 1 and sketched in figure 3(a) are shown in the figures 3(b)–(f) per iteration. All five target parameters converge within a few iterations. The resonator decay rate, κ exhibits fluctuations even after several iterations (figure 3(c)). We ascribe these to the inaccuracy of the κ estimate from the imaginary part of the resonator mode analysis, which, in turn, increases the error in the design variable update (see appendix B). The observed fluctuations of κ also indirectly affect the other targets, especially the resonator frequency.

3.2. Effect of nonlinear models and initial conditions

Following the eigenmode-analysis-based optimization example in the previous section, a natural question arises: how accurate must the nonlinear model be to ensure efficient and rapid convergence? To address this, we compare several optimization runs with identical initial conditions but different nonlinear models for the resonator–qubit dispersive shift χ.

Strikingly, both a very precise model and a variety of less accurate ones converge in a few iterations, as shown in figures 4(a) and (b). As seen in figure 4(a), the most accurate expression used here $\chi = (w/w_{\mathrm{qb}})^2\alpha/(\Delta(\Delta-\alpha))$ [24], accurately accounts for the simultaneous optimization of the mode frequencies (affecting Δ and therefore χ), leading to the fastest convergence. We use $w \equiv w_{\mathrm{res-qb}}$ for brevity in this section. In contrast, none of the other models captures this phenomenon and therefore

overshoots considerably after the first design variable update. After two updates, the other parameters are close to their target, which allows simpler models to be effective from here on as well. Among these, the model $\chi \propto w$ proves itself to be the most effective and rapidly converges to the targeted value. The model $\chi \propto w^2$ assumes a too sensitive dependence on w and therefore suggests a too small update step, leading to slower convergence. This effect is even stronger for the model $\chi \propto w^3$. As the counterpart, the model $\chi \propto \sqrt{w}$ assumes a too weak dependence on w. It therefore suggests a too large update step, which overshoots to the extent that the optimization diverges. Nevertheless, combining the too inaccurate model $\chi \propto \sqrt{w}$ with low pass filtering via the adjustment rate γ of the optimization, convergence can be restored. Furthermore, in figure 4(b), we study a progressively degraded model and show that ANMod can tolerate substantial overshooting while still converging. Note that the users may implement their own physical models to obtain even faster convergence, e.g. including the χ-model discussed in [21]. Furthermore, by studying how convergence is reached, the user acquires a better understanding of a more appropriate model. To summarize, in general, as long as the nonlinear model captures the correct qualitative trends and the step size is suitably chosen, the optimizer will converge.

Since compute time is dominated by the HFSS simulation, the total optimization time scales linearly with the number of iterations for a fixed number of mesh refinement passes. For reference, the single-qubit example requires approximately 280 min for 10 iterations on a single core of an Intel Core i7-14 700 processor with 63.7 GB of usable RAM using ansys electronics desktop HFSS 2022 R2.3, using 8 mesh refinement passes per iteration with an initial fine mesh around the feedline-resonator segment. The study in figure 4(b) was done with the target for κ and fine initial mesh disabled, and with 12 passes. In the end, the user needs to balance the manual effort between refining the nonlinear models versus the waiting time of the simulations or investing in a more powerful workstation.

A second question concerns the robustness of the optimization with respect to different initial conditions, since convergence is generally not guaranteed for nonlinear systems. To investigate this, we perform multiple optimizations of the same resonator–qubit system defined in table 1, but with varying initial values of the design variables. The choice of initial conditions was random, but constrained to the geometric limitations allowed by the design, i.e. the box of the split-transmon and the separation between the launchpads, see appendix B.1. Figures 4(c) and (d) show the convergence trajectories for two parameters of our example design under ten different choices of initial design variables. The same set of initial conditions are encoded with the same shade of blue in the two sub-figures. In eight out of ten tested cases, the optimizer quickly reaches the target parameter values despite the wide range of initial conditions, demonstrating strong robustness against variations in the initial conditions. However, in two out of ten at random tested cases, the optimization did not converge over ten iterations or got aborted by the optimizer.

To avoid the most common failure modes of the optimization, some scenarios should be avoided. First, geometric overlaps during the automatic routing of structures must be prevented; this can be enforced by specifying design variable bounds to avoid physically infeasible layouts. Second, identical frequency ordering of the M modes specified in the optimizer setup and obtained in the HFSS simulation should be maintained throughout the optimization to avoid incorrect parameter update due to misassigned modes. This problem can be identified from the field distribution in HFSS. Third, if the underlying physical relationships in the nonlinear model are fundamentally incorrect, convergence cannot be expected, as the optimizer's predictions will consistently point away from or overshoot beyond the true optimum. In the case of non-monotonic nonlinear models or of counter-acting design variables, the optimization might get stuck in a local minimum.

3.3. Optimization of a Two-qubit device

To demonstrate the effectiveness of the QDesignOptimizer in tackling larger, multi-mode systems, we apply it to a two-qubit quantum processor as in [25]. The system comprised of two X-mon qubits with nonlinear coupler and attached readout resonators includes five modes and 12 target parameters. The detailed problem formulation is described in appendix C. Given this setup, the optimization quickly converges for all parameters as shown in the convergence trajectories versus iterations in figure 5. Notably, the simultaneous 12 parameter nonlinear optimization converges within 5 simulations, i.e. effectively less than 1 simulation per parameter, manifesting a significant reduction in number of simulations (and therefore compute time) compared to a typical manual tune up. As another tool tip, once a suitable problem formulation has been found, it is often possible to re-run the optimization with new parameter targets without further adjustments, given that the mode order in frequency is preserved. This significantly reduces the effort required to change the target parameters. Note that it took one of our

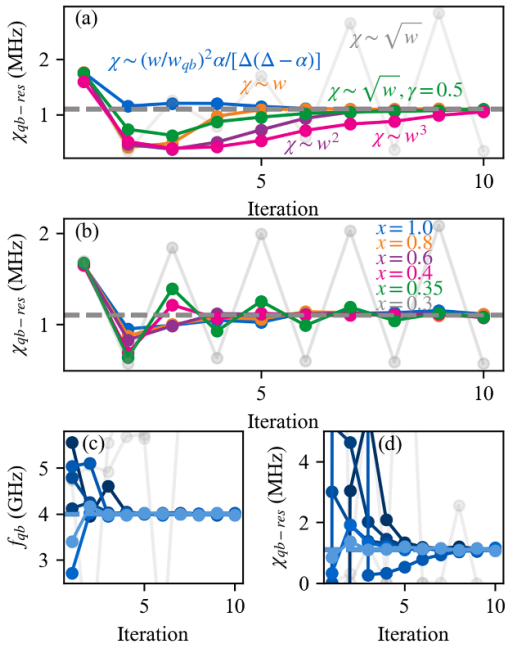

Figure 4. Effect of different nonlinear models and different initial conditions on optimization results obtained from an eigenmode-analysis-based optimization of a resonator–qubit system coupled to a feedline as in figure 3. (a) Optimization result for the resonator–qubit χ for five different nonlinear models. All models lead to convergence except $\chi \propto \sqrt{w}$, which diverges since this model heavily underestimates the sensitivity on w, here denoting $w_{\mathrm{res-qb}}$. However, reducing the adjustment rate to 50%, i.e. low-pass filtering the updates, restores convergence even for this model. For each iteration, 8 simulation passes were used. (b) Optimization result for the resonator–qubit model $\chi = ((w/w_{\mathrm{qb}})^2\alpha/[\Delta(\Delta-\alpha)])^x$ with progressively deviating exponents x = [1.0, 0.8, 0.6, 0.4, 0.35, 0.3]. (c)–(d) Optimization results for (c) the qubit frequency and (d) resonator–qubit dispersive shift χ for ten different initial conditions using the χ expression from table 1 as input to the nonlinear model. Each set of initial conditions is encoded with the same shade of blue for the eight successful optimization runs. The two failed optimizations we label in gray. The dashed lines indicate the parameter target. In each iteration, HFSS performed 4 mesh refinement passes.

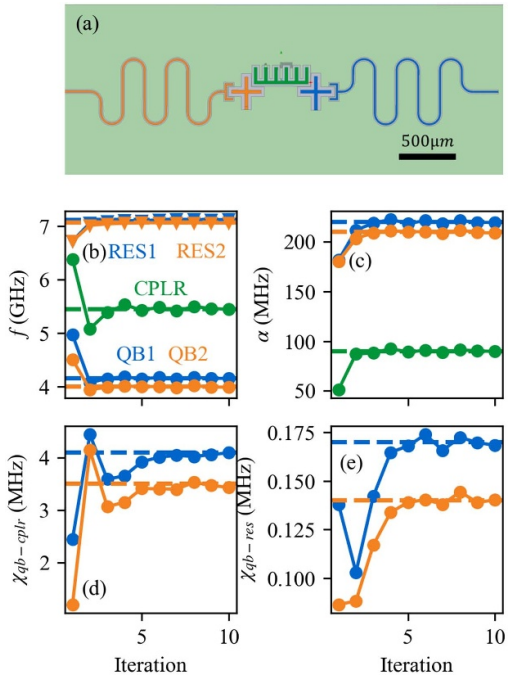

Figure 5. Eigenmode-analysis-based optimization of a system composed of two qubits coupled via a tunable coupler and two readout resonators, as studied in [25]. The problem formulation is described in appendix C consisting of simultaneous optimization of 12 parameters for 5 modes.

QDesignOptimizer developers less than two work days to create the design, setup the problem formulation and reach acceptable convergence. Hence, we believe that the QDesignOptimizer package will prove itself effective when tuning up qubit circuits such as in [26–30]. An improvement to the example shown here could be the addition of a direct capacitive coupling between the two qubits to cancel their cross ZZ interaction as yet another target parameter [31].

3.4. Modular optimization

Simulating larger processor designs with multiple qubits and couplers quickly becomes infeasible on a desktop computer. To address this issue, the QDesignOptimizer workflow supports a modular optimization strategy: first, the chip design can be partitioned into subsystems by selecting relevant quantum-metal components, ports, and modes. Then, the subsystems are iteratively optimized for overlapping sets of coupled modes. Since circuit layouts vary widely, there is no general recipe for the decomposition into subsystems for arbitrary design configurations. As an example in appendix D, we partition the two-qubit design in figure 5 into RES1-QB1-CPLR, RES2-QB2-CPLR and QB1-CPLR-QB2. Partitioning the circuit is one way to use high performance computing workflows, where all partitions are simulated in parallel, and the results are merged in the ANMod step in each iteration. In practice, strongly interacting subsystems are optimized in isolation under the assumption that the influence of excluded structures is negligible. If computationally feasible, the resulting design can be validated through a single, more expensive simulation of the larger system. For instance, instead of simulating a complete multi-qubit device with couplers and readout resonators, one may first optimize smaller building blocks—such as the qubit–resonator system in figure 3(a)—before addressing larger subsystems like the qubit–coupler–qubit configuration in figure 5(a). For even larger scale systems, it might be effective to use pre-simulated libraries [21] to obtain initial design variables close to the target design and then apply the ANMod-method to compensate for the small nonlinear corrections.

3.5. Capacitance-simulations-based optimization

In addition to eigenmode-analysis-based optimization, which supports targets for mode frequencies, inter- and intra-mode nonlinearities, and mode decay rates, the QDesignOptimizer supports optimization workflows based on capacitance simulations. This feature enables the direct targets of capacitances between islands, and derived targets, such as the resonator-feedline coupling rate κ or the T_1 limit due to energy decay from a transmon qubit into a charge line. We present an example for the capacitance-simulation-based optimization in appendix A. Already at present, users may define their own custom targets, leveraging results from both eigenmode and capacitance simulations in the same optimization cycle. For example, a more reliable estimate of the decay rate of readout resonators into the transmission line may be obtained through a capacitance study which accounts for the capacitive load of the coupled qubit. In future developments, we plan to extend the framework to support scattering-parameter simulations to efficiently simulate the resonator-to-transmission line coupling.

4. Conclusion

We have presented QDesignOptimizer, a physics-guided, multi-parameter optimization framework for automated design of superconducting quantum devices, demonstrated on qubit optimization in circuit QED. It integrates high-accuracy electromagnetic simulations with user-defined nonlinear models, achieving stable, fast convergence via the introduced ANMod-method. The implementation is tightly coupled to the quantum-metal/pyEPR toolchain but also provides stand-alone, general-purpose ANMod functionality. In addition, it significantly improves research efficiency by enabling unattended optimization runs, making late-stage design modifications more feasible and reducing the effort required for (re-)optimizing complex superconducting quantum devices. The ANMod-method might also be useful when adjusting for nonlinear corrections when starting from pre-simulated structures in large scale systems.

The ANMod method represents a paradigm shift in optimization: instead of constructing a full self-consistent surrogate model, it directly estimates a suitable update direction and step size for the design variables. The strategy uses physics-informed approximate relations to set the updates so that the predicted relative changes match the required ratios $y_i^{\text{target}}/y_i^{\text{current}}$. As such, the update rule *effectively* includes an approximation of the unmodeled error. The ANMod-method's simplicity makes it feasible

to integrate into established simulation tools used for device design. Consequently, ANMod offers a general and versatile methodology for optimizing complex nonlinear systems across physics and engineering. As a final remark, we foresee a new pathway in which generative AI extracts physical insight from the literature and contributes meaningfully to nonlinear optimization by implementing ANMod models.

Acknowledgment

We thank the other members of the 202Q-lab for valuable discussions and testing. This work was supported by the Knut and Alice Wallenberg Foundation through the Wallenberg Centre for Quantum Technology (WACQT) and by the European Union's Horizon Europe Framework Programme (EIC Pathfinder Challenge project Veriqub) under Grant Agreement No. 101114899. S G acknowledges financial support from the European Research Council (Grant No. 101041744 ESQuAT). External interest disclosure: S G is a co-founder and equity holder in Sweden Quantum AB.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/202Q-lab/QDesignOptimizer [23].

Code availability

The code for the QDesignOptimizer is available at [23], including the stand-alone ANMod-functionality and the implementation with HFSS and quantum-metal plus installation instructions. The repository includes examples for single-qubit, two-qubit (both full and partitioned), and high-dimensional cluster optimization, as well as publication scripts for reproducing the results in this paper.

Author contributions

A M E developed the ANMod-method and created the basis of the QDesignOptimizer. A M E and O L constructed the high-dimensional test system problem. L S led the work on refactoring code, creating online documentation including single and two-qubit examples, with contributions from all authors. L S constructed the simulations for the single-qubit example, including the capacitance and convergence studies. A M E extended L S's two-qubit example and constructed the corresponding example in this article. A M E and L S wrote the paper with contributions from all authors.

Appendix A. Capacitance–simulations–based optimization

The QDesignOptimizer also supports capacitance targets or targets derived thereof. Capacitance optimizations can be run separately or jointly with the eigenmode simulations. Two already built-in derived target parameters are the T_1 limit due to energy decay from a transmon qubit into a charge line, and the resonator-feedline coupling rate κ. In future work additional derived targets, including Purcell-limited decay and dephasing due to thermal photons in the readout resonator could be added. Already at present, users may define their own custom targets, leveraging results from both eigenmode and capacitance simulations.

Figure S1(a) shows an example setup consisting of a coplanar waveguide (CPW) resonator capacitively coupled to a finite-length CPW transmission line. The optimization goal is to achieve a specified target capacitance between the two structures. As the physical model, we assume an inverse square-root dependence of the capacitance on the coupling distance, which serves as the design variable in this case. Note that the relation $C \sim \sqrt{d}$ is an empirical approximation for the coplanar coupling geometry. The resulting optimization, shown in figure S1(c), reaches the target value in approximately five iterations for the given initial conditions.

In a second example, we consider a split-transmon qubit with a nearby charge line, as illustrated in figure S1(b). Here, the objective is to optimize the T_1 limit set by energy decay into the charge line, using the position of the charge-line tip as the design variable. The nonlinear model assumes a cubic dependence of the target parameter, the T_1 limit, on the relative position of the charge line. Notably, defining design variables in terms of relative coordinates or distances, rather than absolute positions, often improves optimization robustness, as it reduces sensitivity to global coordinate shifts or layout changes and greatly reduces the complexity of the physical relationship. We obtain the T_1 limit from the evaluation of a classical model for a single mode decay into a single decay channel taking the capacitance matrix, the mode frequency and the line impedance as input [32]. The optimization result, shown in figure S1(d), converges to the target within roughly five iterations, demonstrating the efficiency of capacitance-based optimization workflows for both direct and derived target quantities.

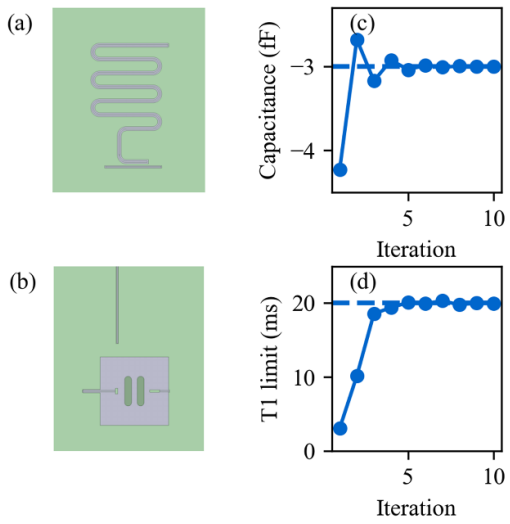

Figure S1. Capacitance-simulation-based optimizations. (a), (c) Screen capture and optimization result for the coupling capacitance between a coplanar waveguide resonator to a finite-length feedline. (b), (d) Screen capture and optimization result for the T_1 limit of the split-transmon decaying into the neighboring charge line at the top in the presence of other coupling ports to the left and right side. The T_1 limit is derived from the full capacitance matrix of the circuit. Per iteration 10 simulation passes were used.

Appendix B. Effects of simulation setup on optimization

To understand the effect of the accuracy of the parameter data obtained from the simulations on the convergence of the optimization, we perform independent optimizations with 3 to 10 mesh refinement passes in HFSS and compare the final design variables obtained after ten iterations. The resulting difference between design variables is shown in figure S2(a). While the resonator length and the qubit's Josephson inductance remain relatively stable across different pass numbers, the qubit pad length and the resonator–qubit coupling segment length exhibit variations of approximately 3%. In contrast, the resonator-feedline coupling segment length exhibits variations of up to 10% relative to the most accurately simulated values obtained with the largest number of passes for this set of initial conditions. From this graph, it becomes apparent that all design variables require eight or more passes to obtain accurate optimization results in this particular simulation setup. Note that the exact number of required passes depends on the feature sizes, the initial mesh, and the mesh refinement rate. Thus, there is a trade-off: a higher number of passes improves accuracy, but also increases overall optimization time. Especially the resonator-feedline coupling segment length (lightblue) shows a non-monotonic trend with increasing number of passes, which is primarily due to the inherent limitations in extracting κ from HFSS eigenmode simulations via the imaginary part of the resonator eigenmode. HFSS's adaptive meshing per simulation pass prioritizes regions of high electric field rather than areas requiring accuracy for coupling parameter extraction, such as the coupling segment geometry. Hence, a high mesh refinement yielding

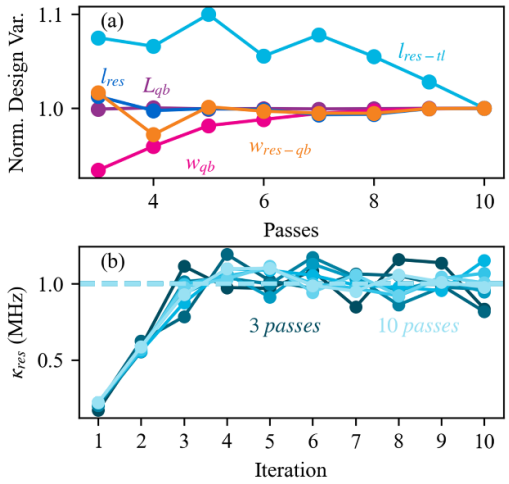

Figure S2. Eigenmode-analysis-based optimization of a resonator–qubit system coupled to a feedline for different numbers of passes. (a) Relative difference of the five design variables obtained after ten iterations of optimization with different numbers of passes normalized to the results obtained from the largest number of passes. The colors of the design variables match the colors of the optimization targets in figure 3. (b) Optimization results versus iterations for the resonator κ for different numbers of passes used during the simulation. Each number of passes between 4 and 10 is encoded in a different shade of lightblue.

an accurate κ estimate is only reached at a high number of passes. As a result of this meshing problem, κ exhibits the largest fluctuations, especially for a few number of passes, compared to all other parameters, directly affecting the stability of its corresponding design variable during optimization. The fluctuations of κ are visible in figure S2(b), which shows the κ estimates throughout the optimizations for different number of passes encoded in different shades of lightblue. To partially mitigate the problem of insufficient mesh refinement in the resonator-feedline coupling region, we manually specify a finer, initial meshing map locally around the coupling segment in addition to the coarse default initial mesh assigned by HFSS in all simulations shown in this paper. With this additional fine initial mesh, a ten-iteration optimization with eight mesh refinement passes each requires approximately 280 min on one of our office workstations equipped with an Intel Core i7-14 700 processor and 63.7 GB of usable RAM using ansys electronics desktop HFSS 2022 R2.3.

B.1. Design variable constraints for initial conditions used in single qubit example

The ten different sets of initial conditions used for figures 4(c) and (d) were uniformly and independently sampled within the boundaries listed in table 2. These limits were set to ensure geometric validity, e.g. such that the qubit island does not extend beyond the box dimensions.

Table 2. Design variable constraints used to sample the initial conditions seeding the optimizations shown in figures 4(c) and (d).

Design Variable	Min	Max
Qubit pad eidth	100 μm	1100 μm
Qubit Josephson inductance	5 nH	25 nH
Resonator length	4000 μm	12 000 μm
Qubit–Resonator coupling length	100 μm	1100 μm
Resonator- Feedline coupling length	100 μm	1400 μm

Appendix C. Optimization setup used in the two-qubit example

The problem formulation for the two-qubit-coupler example discussed in section 3.3 is summarized in table 3. The setup is notably similar to the problem formulation of the single-qubit–resonator example. The target parameter values were derived from [25]. For the final optimization, one optimization iteration took less than 30 min on an Intel Core i7-10 700 consuming less than 30 GB of RAM-memory.

Table 3. Summary of the problem formulation for the two qubit quantum processor studied in [25]. The qubit-coupler couplings target values are translated from [25] assuming the relation $\chi = g^2/\Delta$. The parameters for the qubit–resonator systems are the same as in table 1 with subscripts $i = 1, 2$, apart from that $w_{\mathrm{res-qb,i}}$ is here the length of the horizontal claw segment constrained to $< 80\mu$m. Additionally, we have introduced the design variables coupler inductance L_c, coupler finger length w_c, and diagonal distance of the coupler-qubit metal strip separation $w_{\mathrm{c-qb,i}}$.

Parameter	Symbol	Target	Proportional to	Design variable	Initial value
Resonator frequency	$f_{\mathrm{res},1}$	7.12 GHz	$1/l_{\mathrm{res},1}$	$l_{\mathrm{res},1}$	4200 μm
Resonator frequency	$f_{\mathrm{res},2}$	7.07 GHz	$1/l_{\mathrm{res},2}$	$l_{\mathrm{res},2}$	4200 μm
Qubit frequency	$f_{\mathrm{qb},1}$	4.16 GHz	$1/\sqrt{L_{\mathrm{qb},1} \cdot w_{\mathrm{qb},1}}$	L_{qb}	10 nH
Qubit frequency	$f_{\mathrm{qb},2}$	4.00 GHz	$1/\sqrt{L_{\mathrm{qb},2} \cdot w_{\mathrm{qb},2}}$	$L_{\mathrm{qb},2}$	12 nH
Coupler frequency	f_c	4.00 GHz	$1/\sqrt{L_c \cdot w_c}$	L_c	2.0 nH
Anharmonicity	$-\alpha_1/2\pi$	220 MHz	$1/w_{\mathrm{qb},1}$	$w_{\mathrm{qb},1}$	170 μm
Anharmonicity	$-\alpha_2/2\pi$	210 MHz	$1/w_{\mathrm{qb},2}$	$w_{\mathrm{qb},2}$	170 μm
Anharmonicity	$-\alpha_c/2\pi$	90 MHz	$1/w_c$	w_c	250 μm
Dispersive shift	$-\chi_1/2\pi$	0.17 MHz	$(w_{\mathrm{res-qb},1}/w_{\mathrm{qb},1})^2 \cdot \alpha_1/[\Delta_1(\Delta_1 - \alpha_1)]$	$w_{\mathrm{res-qb},1}$	30 μm
Dispersive shift	$-\chi_2/2\pi$	0.14 MHz	$(w_{\mathrm{res-qb},2}/w_{\mathrm{qb},2})^2 \cdot \alpha_2/[\Delta_2(\Delta_2 - \alpha_2)]$	$w_{\mathrm{res-qb},2}$	30 μm
Coupler-qubit coupling	$-\chi_{c1}/2\pi$	4.1 MHz	$1/[w_{\mathrm{c-qb},1}(f_c - f_{\mathrm{qb},1})]$	$w_{\mathrm{c-qb},1}$	13 μm
Coupler-qubit coupling	$-\chi_{c2}/2\pi$	3.5 MHz	$1/[w_{\mathrm{c-qb},2}(f_c - f_{\mathrm{qb},2})]$	$w_{\mathrm{c-qb},2}$	15 μm

Appendix D. Partitioned two-qubit example

Here we show how the two-qubit design in figure 5 can be partitioned into RES1-QB1-CPLR, RES2-QB2-CPLR and QB1-CPLR-QB2 which are simulated independently and the results are merged in the approximate nonlinear model-driven (ANMod) update step, as presented in figure S3. The setup for the partitions in figure S3 is identical to the setup used in appendix C for the full system. Partition 1 updates $f_{\mathrm{res},1}, f_{\mathrm{qb},1}, \alpha_1$ and χ_1, partition 2 updates $f_{\mathrm{res},2}, f_{\mathrm{qb},2}, \alpha_2$ and χ_2 and partition 3 updates f_c, α_c, χ_{c1} and χ_{c2}. Note that partitioning the system may bias the information collection step, as the partitions only approximate the full system. However, the convergence of the full (figure 5) and partitioned (figure S3) systems are very similar but the partitioning usually takes longer time since it simulates 3+3+3 modes in comparison to a single simulation of 5 modes for the full system.

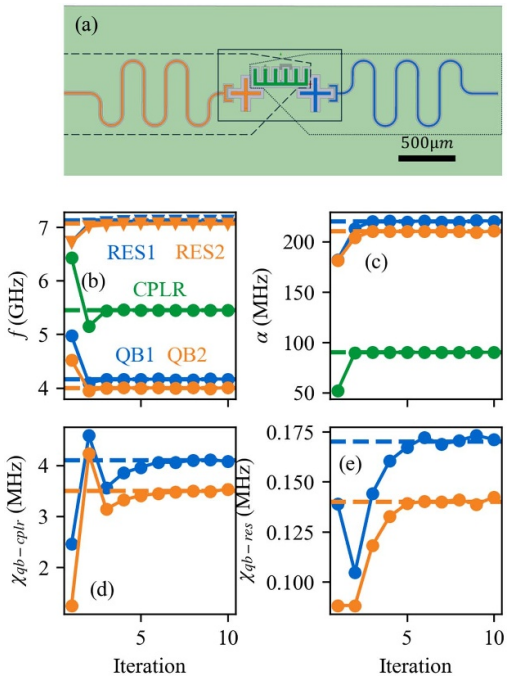

Figure S3. Partitioned simulation of a multi-component chip. The setup is identical to 5 but instead of simulating all five modes simultaneously, the simulation is split up into three independently simulated partitions containing the structures indicated by the dashed, dotted and solid sections.

Even smaller partitioning can be useful to stabilize the optimization further by first tuning up single mode properties such as frequencies and then gradually increasing cross-mode couplings, which often are comparably weak in the dispersive regime. Partitioning the full circuit is useful when the full circuit

is too resource demanding to be simulated as a whole, usually due to RAM-memory constraints. For very large repetitive circuits, it can be beneficial to use pre-simulated knowledge for example using [21] and then finetune the system with a few ANMod iterations.

Appendix E. Recommendations for efficient optimizations

For the setup of efficient optimizations through our framework, we present in the following a few recommendations.

E.1. Make design variables independent

The optimization problem will be much more stable and easier to describe mathematically if the inter-dependence of design variables and parameters are reduced to the level that they can be neglected. Strive to ensure that each design variable has a significant effect on only one target parameter. Note that one design variable can control multiple physical dimensions in a design, which might help in making the design variables independent. For example, a length-type variable could in principle simultaneously determine the width and length of a capacitance pad.

E.2. Factorize the update step for independent design variables

The nonlinear minimization step is simplified by noting that the parameters f_{res} and E_{c} only depend on l_{res} and w_{qb}, respectively. Hence, we can reduce the dimension of the minimization problem by running equation (5) first for the one-dimensional problems $(f_{\mathrm{res}}, l_{\mathrm{res}})$, $(\kappa_{\mathrm{res}}, l_{\mathrm{res-tl}})$ and $(f_{\mathrm{qb}}, w_{\mathrm{qb}})$ to obtain l_{res}^{k+1} and w_{qb}^{k+1} and then minimize the remaining (two-dimensional) problem for $(f_{\mathrm{qb}}, \chi, L_{\mathrm{qb}}, w_{\mathrm{res-qb}})$. In this way we have run the smaller problem of dimensions 1, 1, 1, and 2 instead of running the full 5 dimensional problem, which generally would take longer to solve. As an example, if we define the $l_{\mathrm{res-tl}}$ coupling length such that it does not affect the total length of the resonator, we (approximately) decouple the optimization of f_{res} and κ_{res}.

E.3. Make effects of design variables predictable

Depending on how a design variable affects a design, it might have a simpler or more complicated (even non-monotonic) relationship to its parameter. It is generally advisable to keep the relations between design variables and parameters as simple as possible, and to ensure that the chosen design variables are the layout properties that affect each parameter the most when changed. For example, the coupling capacitance between a qubit and a resonator scales linearly with the width of its coupling pad only when the pad is much wider than the resonator's center conductor.

E.4. Advanced design variable parameterization

The current implementation of ANMod requires an equal number of design variables and parameters, ensuring a properly constrained single-objective problem. However, it is often useful to make a design variable affect several aspects of the design. A representative example is resonator linewidth, which can be tuned through both coupling length and distance to the transmission line. Only optimizing the coupling length while the distance remains too large may prevent the target from being reached. A useful strategy is to connect the design variable x to affect both the distance $d = d_0/x$ and the length $l = xl_0$ (where d_0 and l_0 are the initial values). Doing so, the coupling increases with increasing x by both bringing the coupling closer and making it longer and a wider range of targets can be reached. It can also be useful, for example, to in a similar manner adjust the relative positions of components when scaling their sizes.

E.5. Investigate unknown physical relationships

In the case where it is difficult to construct a predictable design, the QDesignOptimizer can also be used to investigate the physical relationship between design variables and parameters, which is useful when the relationship is unknown or needs to be verified. To do this, the optimization is run without targets and the design variable of interest is given explicit values in each iteration, see the documentation [23] for details. The obtained trajectory can then be fitted to a simple model, such as a power law, to obtain an approximate physical relation. An example for such a parameter sweep is shown in figure S4.

E.6. Faster convergence

To reduce the total simulation time, the convergence criteria (for example the number of passes in the HFSS simulation) can be successively increased for each iteration, such that a more detailed simulation only is run when the design variables and parameters are close to target.

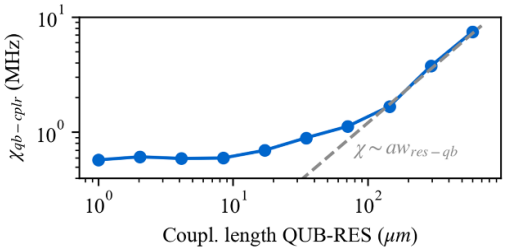

Figure S4. Extracting the physical relationship between the qubit–resonator coupling length $w_{\mathrm{res-qb}}$ and the dispersive shift $\chi_{\mathrm{qb-res}}$. The coupling length was swept from 1 to 600 μm using 10 optimizer iterations and 12 HFSS mesh refinement passes per iteration. For very short coupling lengths, the coupling is dominated by the arm to the coupling pad, i.e. in this regime $w_{\mathrm{res-qb}}$ is not a suitable design variable. The coupling then increases approximately linearly with $w_{\mathrm{res-qb}}$, consistent with a power-law dependence $\chi \sim a \cdot w_{\mathrm{res-qb}}$ (dashed line). Note, that a does not enter in the proportionality relation fed into ANMod. At the longest lengths, the coupling pad becomes as wide as the split-transmon.

E.7. Tracking of design variable updates

To aid the debugging and help understanding parameter updates within the large design space, the evolution of the design variables through the optimization can also be tracked, as shown in figures S5(a) and (b) for the case of the qubit Josephson inductance and the resonator length for two different sets of initial conditions. The final design variable values correspond to the parameters close to the desired parameter targets. Similarly, the tracking of a system parameter versus the corresponding design variable exhibits the dynamics of the optimization. A corresponding example of tracking the dispersive shift $\chi_{\mathrm{qb-res}}$ versus the coupling length between qubit and resonator $w_{\mathrm{res-qb}}$ is shown in figure S5(c). The evolution through the parameters space differs for different nonlinear models (those used in figure 4(a)). The optimization trajectory is most direct and monotonic for the most accurate nonlinear model, here represented by the blue line.

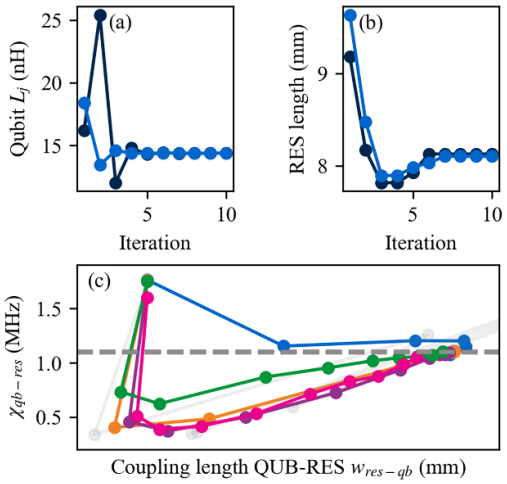

Figure S5. Tracking design variables. (a) Qubit Josephson inductance and (b) resonator length versus iteration for two different sets of initial conditions matching the same colored sets used in figure 3(b)–(c). (c) Tracking of the dispersive shift $\chi_{\mathrm{qb-res}}$ versus the design variable coupling length $w_{\mathrm{res-qb}}$ between qubit and resonator for different nonlinear models as presented with matching colors in figure 4(a).

Appendix F. ANMod-optimization scalability

In this section, we illustrate that the ANMod-optimization method can exhibit excellent scaling capabilities by solving a coupled nonlinear system with 3000 parameters and corresponding design variables. The code is available in [23]. The example emphasizes that, as long as the strong nonlinear couplings can be approximated and solved in local smaller problems (by conventional e.g. gradient-based methods), the perturbative approach of the ANMod-method can handle both errors in the strong local model as well as weaker global couplings neglected by the approximate model.

In this example, we construct clusters of parameters such that within each cluster there are strong nonlinear interactions between all parameters and design variables. In addition, there are weak couplings between parameters from different clusters. The system then takes the form

$$y_{i,j} = g_{i,j}\left(\overrightarrow{x}_i, \overrightarrow{y}_i\right) h_{i,j}\left(\overrightarrow{y}, \overrightarrow{x}\right) \tag{F1}$$

where there is a strong nonlinear relation $g_{i,j}$ for all parameters $\overrightarrow{y}_i$ and design variables $\overrightarrow{x}_i$ within cluster i, but only weak nonlinear interaction $h_{i,j}$ to parameters and variables in other clusters. As a concrete example, assume that we have $n = 1000$ clusters each with $m = 3$ strongly interacting parameters, resulting in a nonlinearly coupled system with 3000 parameters and 3000 design variables in total. We will first construct an exact system (which will be used in the *information collection* stage) and then describe the approximate model used by the ANMod-method. To construct the exact system, let us assume some explicit relations

$$g_{i,j}\left(\overrightarrow{x}_i, \overrightarrow{y}_i\right) = \left(\prod_{k=1}^{m} x_{i,k}^{\alpha_{i,j\leftarrow k}}\right)\left(\prod_{k=1,k\neq i}^{m} y_{i,k}^{\beta_{i,j\leftarrow k}}\right), \tag{F2}$$

which is inspired by that many physical systems have power law relations. For example, most relations in table 3 are of this form. The coefficients $\alpha_{i,j\leftarrow k}$ are sampled uniformly from

$$U_{\alpha_{i,j\leftarrow k}} = \begin{cases} [2.0, 2.5], & \text{if } j = k, \\ [-1.0, -0.1], & \text{if } j \neq k, \end{cases} \tag{F3}$$

i.e. $y_{i,j}$ depends more strongly on $x_{i,j}$ than the other design variables in the cluster such that the problem is well conditioned (note that this is not a strict requirement). This reflects the fact that the user is able to define suitable design variables which have a strong impact on the corresponding parameter. The coefficients $\beta_{i,j\leftarrow k}$ are sampled uniformly from

$$U_{\beta_{i,j\leftarrow k}} = [-0.5, -0.1] \cup [0.1, 0.5], \tag{F4}$$

where we explicitly exclude exponents close to 0 to keep high connectivity within the clusters, since in practice an exponent close to 0 would eliminate the dependency between the parameters. The subscript $j \leftarrow k$ describes how parameter j depends on parameter k within cluster i. Furthermore, we assume the weaker global relations

$$h_a\left(\overrightarrow{y}, \overrightarrow{x}\right) = 1 + \epsilon y_{b_a}^{\gamma_{b_a}} y_{c_a}^{\gamma_{c_a}} y_{d_a}^{\gamma_{d_a}} \tag{F5}$$

where the small parameter $\epsilon = 0.05$ manifests the isolation between separate clusters. Here, each parameter with composite index $a = (i, j)$ has a weak coupling to the product of three other parameters from random places in the total system to enhance the nonlinear order of the problem. These interactions are indicated with the chained composite indices b_a with exponents γ_{b_a}, which is sampled from

$$U_{\gamma_{i,j\leftarrow k}} = [-1.5, -0.1] \cup [0.1, 1.5]. \tag{F6}$$

The composite indices fulfill $a \neq b \neq c \neq d$, and b, c, d are not from the same cluster i as a. Note that we could let $y_{i,j}$ also depend weakly on other design variables but that is omitted here for brevity.

As for the approximate model, we assume it approximately captures the strong coupling exponents within each cluster but misses the additional nonlinear dependencies on the parameters from other clusters such that

$$\tilde{y}_{i,j} \propto f_{i,j}^{\text{approx}}\left(\overrightarrow{y}_i, \overrightarrow{x}_i\right) = g_{i,j}^{\text{approx}}\left(\overrightarrow{y}_i, \overrightarrow{x}_i\right) \tag{F7}$$

$$= \left(\prod_{k=1}^{m} x_{i,k}^{R\left(\alpha_{i,j\leftarrow k}\right)}\right)\left(\prod_{k=1,k\neq i}^{m} y_{i,k}^{R\left(\beta_{i,j\leftarrow k}\right)}\right) \tag{F8}$$

where

$$R(\xi) = \text{round}(N\xi)/N, \quad N = 4. \tag{F9}$$

The higher the integer N, the more accurate the approximate model is to the actual system relations.

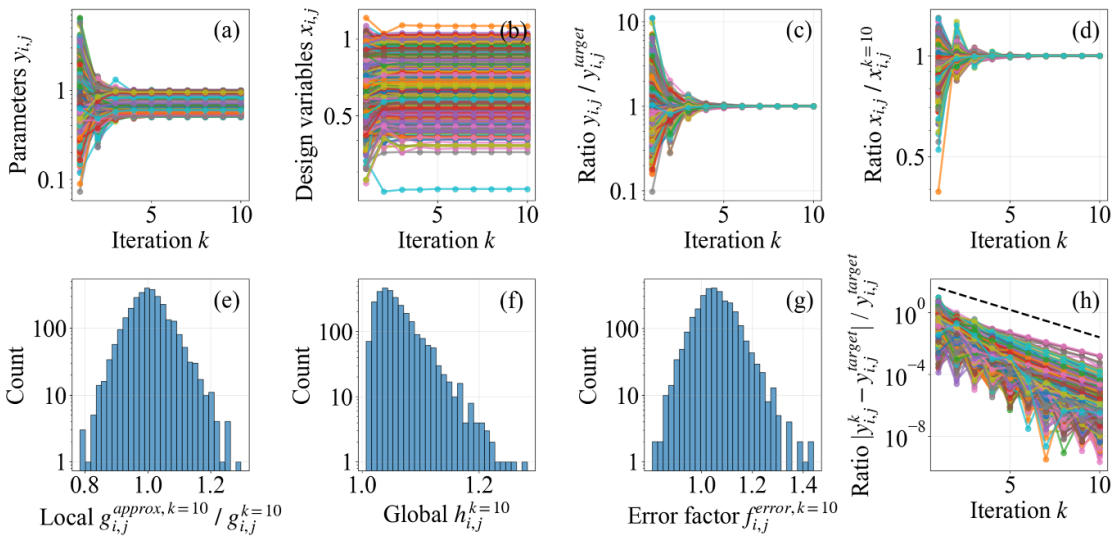

Figure S6. Convergence of the 3000-parameter example and indication of the accuracy of the approximate model. (a) Convergence of parameters towards their absolute target values. (b) Convergence of design variables towards their absolute values. (c) Convergence of parameters towards their values relative to target. (d) Convergence of design variables relative to their converged values. (e) Relative error between the approximate model and the strong local couplings at the last iteration. (f) Global-interaction error factor at the last iteration. (g) Error between approximate and full model evaluated at the last iteration. I.e. combining the local error from (e) with the global error in (f). (h) Convergence of the relative error. The dashed line is a guide to the eye of the exponential decay $(\max(|f_{i,j}^{\text{error},k=10} - 1)|)^k \approx 0.44^k$ from (g).

Finally, we assume both target parameters $y_{i,j}^{\text{target}}$ and initial design variables $x_{i,j}^0$ to be uniformly sampled from

$$U_i = [0.5, 1.0] \tag{F10}$$

reflecting that the design variables and parameters have suitable normalization.

Subsequently, we run the *information collection* step and then update the design variables using the ANMod-method. To execute the information collection, i.e. which $y_{i,j}$ we get for $\overrightarrow{x}$, we iteratively evaluate the forward problem

$$y_{ij}^{k+1} = g_{i,j}\left(\overrightarrow{x}_i, \overrightarrow{y}_i^k\right) h_{i,j}\left(\overrightarrow{y}^k, \overrightarrow{x}\right) \tag{F11}$$

starting from $y_{i,j}^0$ which is given by solving the local strongly nonlinear problem $y_{i,j}^0 = g_{i,j}(\overrightarrow{x}_i, \overrightarrow{y}_i^0)$ for each cluster independently. Since $h_i \approx 1$, the information collection step given by equation (F11) will converge. That is, we have constructed a nonlinear system which we can solve numerically via equation (F11), and used this to mimic the information collected we would usually have obtained from lengthy simulations or measurements of a nonlinear system we only know approximately.

For the update of $x_{i,j}$, we use the ANMod-method with the approximate model described above. The ANMod-optimizer solves each cluster's approximate nonlinear model independently with a traditional gradient-based solver. Although we have a coupled nonlinear system of dimension 3000, the ANMod-method converges within a few iterations, as seen in figure 2. Note that it is very difficult to give any conditions for when the ANMod-method will fail, since it depends strongly on the nonlinearities, target values, initial conditions as well as how crude the approximate model is. Figure S6 presents key metrics for the convergence of the 3000-parameter example. Figures S6(a)–(d) shows the convergence for the absolute and scaled parameter as well as design variable values. Figure S6(e) Describes the ratio between the exact model's local factor g and the approximate model, whereas figure S6(f) describes the global correction factor which is unmodeled in the approximate model. That is, these distributions describe how much of the error in the approximate model comes from local and global corrections. Figure S6(g) shows the error factor

$$f_{i,j}^{\text{error},k} = \frac{g_{i,j}\left(\overrightarrow{y}_i^k, \overrightarrow{x}_i^k\right) h_{i,j}\left(\overrightarrow{y}_i^k, \overrightarrow{x}_i^k\right)}{g_{i,j}^{\text{approx}}\left(\overrightarrow{y}_i^k, \overrightarrow{x}_i^k\right)} \tag{F12}$$

of the approximate model, which is on the order of up to 40% at the last iteration. These error factors are often strongly correlated with how slowly the relative errors converge, as seen in figure S6(h). The

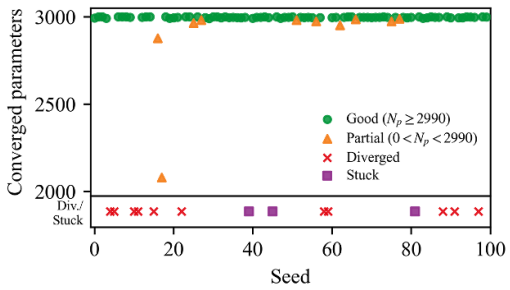

Figure S7. Convergence robustness of the 3000-parameter optimization across 100 unique chosen seeds. Each marker indicates the number of converged parameters N_p for a single seed. Successful runs ($N_p \geqslant 2990$, green circles) and partial runs ($0 < N_p < 2990$, orange triangles) are shown in the main panel. Seeds that diverged or became stuck during optimization are shown in the lower strip.

model interactions were randomly re-sampled 100 times (see full implementation in [23]); within 10 iterations, 25 realizations fully converged below a relative error of 10^{-3} and 15 diverged. Of the remaining cases, 50 reached the convergence criteria for at least 2990 parameters and the final 10 cases have not yet reached the convergence criteria, as visualized in figure S7. This fictive example takes approximately 5 min per seed with ten iterations on a single CPU core. Hence, even at these strong interaction strengths and high dimensionality, the ANMod-method can be able to optimize the design.

Appendix G. Methodology comparison between the ANMod-method and surrogate models

The types of optimization problems addressed in this article are generally characterized by an accurate or *fine* model $y = f^{\text{fine}}(x)$ which is expensive to compute. Consequently, it is infeasible to directly apply a conventional gradient-based or gradient-free optimizer to the problem since each parameter evaluation of the fine model is time-consuming. To mitigate expensive computation times and reduce the number of required parameter evaluation steps, a common approach is to develop a cheap-to-compute approximate or *crude* model often called a surrogate model [33], which guides the parameter updates. Surrogate models can span from fully data-driven [34] (model-less) such as Gaussian process regression, to fitting simple polynomials or artificial neural networks, and more model-full approaches. In microwave engineering, a popular (usually) model-full approach is the so-called space mapping technique which assumes that a simple physical model $y^{\text{crude}} = f^{\text{crude}}(x^{\text{crude}})$ can be formulated, which approximates the fine system. Generally, f^{crude} can implicitly depend on y^{crude}. By sampling the fine model, linear (or weakly nonlinear) maps are fitted to map the input and output spaces of the fine and crude models $x^{\text{crude}} = M_{\text{in}}(x)$ and $y = M_{\text{out}}(y^{\text{crude}})$. In addition, the crude model may have auxiliary variables which can be optimized such that the crude model approximates the fine model more accurately. Space-mapping techniques can typically handle weak nonlinear behavior not modeled in the crude model (similar to the global interactions in the example in section 2.5). The reason is that these weak nonlinearities perturb the system slightly, which locally is well approximated as small skews, scalings and biases. These are absorbed into the maps of the space-mapping technique. After convergence, the fine model and space-mapped surrogate model $f^{\text{surrogate}}(x) = M_{\text{out}}(f^{\text{crude}}(M_{\text{in}}(x)))$ locally produces the same optimization minima. Similarly, most model-full approaches aim to develop a data informed surrogate model $y = f^{\text{surrogate}}(x)$ (which at least locally) behaves approximately as the fine model. The design variable updates are then obtained by iteratively updating and optimizing the surrogate model.

The ANMod-optimization is a model-full approach, but in contrast to surrogate models ANMod relaxes the requirement of producing a self-consistent map $y = f^{\text{surrogate}}(x)$. Such a mapping is not necessary to locate the optimum of the fine model. However, required is an update rule of design variables that suggests a suitable direction and step size, based on the relative change needed from the current position. Hence, the ANMod-method does not yield a self-consistent surrogate model, but instead employs the approximation $y \propto\approx f^{\text{ANMod}}(x, y)$. The design-variable updates are not obtained by minimizing a surrogate function, but by optimizing the relative change predicted when evaluating the approximate model at the current and target position, as described in equation (5).

ORCID iDs

Axel M Eriksson 0000-0002-2757-9804
Lukas J Splitthoff 0000-0001-9328-5604
Harsh Vardhan Upadhyay 0009-0000-0733-5733
Pietro Campana 0009-0004-8271-242X
Niranjan Pittan Narendiran 0009-0001-7877-7448
Kunal Helambe 0009-0000-5619-8018
Linus Andersson 0009-0003-7603-6587
Simone Gasparinetti 0000-0002-7238-693X

References

[1] Blais A, Huang R-S, Wallraff A, Girvin S M and Schoelkopf R J 2004 Cavity quantum electrodynamics for superconducting electrical circuits: an architecture for quantum computation *Phys. Rev. A* **69** 062320
[2] Krantz P, Kjaergaard M, Yan F, Orlando T P, Gustavsson S and Oliver W D 2019 A quantum engineer's guide to superconducting qubits *Appl. Phys. Rev.* **6** 021318
[3] Bravyi S, Cross A W, Gambetta J M, Maslov D, Rall P and Yoder T J 2024 High-threshold and low-overhead fault-tolerant quantum memory *Nature* **627** 778
[4] Acharya R AI Collaborators 2025 Quantum error correction below the surface code threshold *Nature* **638** 920
[5] Castellanos-Beltran M A and Lehnert K W 2007 Widely tunable parametric amplifier based on a superconducting quantum interference device array resonator *Appl. Phys. Lett.* **91** 083509
[6] Gao Y Y, Rol M A, Touzard S and Wang C 2021 Practical guide for building superconducting quantum devices *PRX Quantum* **2** 040202
[7] Levenson-Falk E M and Shanto S A 2024 (arXiv:2411.16967[quant-ph])
[8] Minev Z K et al 2021 Qiskit Metal: an open-source framework for quantum device design & analysis (available at: https://doi.org/10.5281/zenodo.4618153)
[9] Cucurachi D et al 2021 KQCircuits (available at: https://doi.org/10.5281/zenodo.4944796)
[10] GDSfactory 2025 GDSfactory (available at: https://github.com/gdsfactory/gdsfactory)
[11] Gely M F and Steele G A 2020 Qucat: quantum circuit analyzer tool in python *New J. Phys.* **22** 013025
[12] Groszkowski P and Koch J 2021 Scqubits: a python package for superconducting qubits *Quantum* **5** 583
[13] Tanamoto T, Ishikawa T, Inomata K, Masuda S, Omuma T and Kawabata S 2023 Classical spice simulation of superconducting quantum circuits *Appl. Phys. Express* **16** 034501
[14] Kunasaikaran J, Mato K and Wille R 2024 A framework for the design and realization of alternative superconducting quantum architectures (available at: https://doi.org/10.1109/ISMVL60454.2024.00027)
[15] Zaccaria S and Gnudi A 2025 QuLTRA: quantum hybrid lumped and transmission lines circuits analyzer (arXiv:2509.03651[quant-ph])
[16] CSC 2025 Elmer (available at: https://www.elmerfem.org/blog/)
[17] Amazon Web Services 2024 Palace: parallel finite element code for complex assemblies (available at: https://github.com/awslabs/palace)
[18] Ansys 2021 Ansys (available at: https://www.ansys.com/products/electronics/ansys-hfss)
[19] S. Software 2025 Sonnet (available at: https://www.sonnetsoftware.com/)
[20] Li F-Y and Jin L-J 2023 Quantum chip design optimization and automation in superconducting coupler architecture *Quantum Sci. Technol.* **8** 045015
[21] Shanto S, Kuo A, Miyamoto C, Zhang H, Maurya V, Vlachos E, Hecht M, Shum C W and Levenson-Falk E 2024 SQuADDS: a validated design database and simulation workflow for superconducting qubit design *Quantum* **8** 1465
[22] Minev Z K, Leghtas Z, Mundhada S O, Christakis L, Pop I M and Devoret M H 2021 Energy-participation quantization of josephson circuits *npj Quantum Inf.* **7** 131
[23] Eriksson A M, Splitthoff L J, Vardhan Upadhyay H, Campana P, Lundström O, Pittan Narendiran N, Helambe K, Andersson L and Gasparinetti S 2025 QDesignOptimizer (available at: 202Q-lab/QDesignOptimizer)
[24] Blais A, Grimsmo A L, Girvin S M and Wallraff A 2021 Circuit quantum electrodynamics *Rev. Mod. Phys.* **93** 025005
[25] Sung Y *et al* 2021 Realization of high-fidelity cz and *zz*-free iswap gates with a tunable coupler *Phys. Rev. X* **11** 021058
[26] Sheldon S, Magesan E, Chow J M and Gambetta J M 2016 Procedure for systematically tuning up cross-talk in the cross-resonance gate *Phys. Rev. A* **93** 060302
[27] McKay D C, Filipp S, Mezzacapo A, Magesan E, Chow J M and Gambetta J M 2016 Universal gate for fixed-frequency qubits via a tunable bus *Phys. Rev. Appl.* **6** 064007
[28] Noguchi A, Osada A, Masuda S, Kono S, Heya K, Wolski S P, Takahashi H, Sugiyama T, Lachance-Quirion D and Nakamura Y 2020 Fast parametric two-qubit gates with suppressed residual interaction using the second-order nonlinearity of a cubic transmon *Phys. Rev. A* **102** 062408
[29] Moskalenko I N, Simakov I A, Abramov N N, Grigorev A A, Moskalev D O, Pishchimova A A, Smirnov N S, Zikiy E V, Rodionov I A and Besedin I S 2022 High fidelity two-qubit gates on fluxoniums using a tunable coupler *npj Quantum Inf.* **8** 130
[30] Warren C W *et al* 2023 Extensive characterization and implementation of a family of three-qubit gates at the coherence limit *npj Quantum Inf.* **9** 44
[31] Fors S P, Fernández-Pendás J and Kockum A F 2024 Comprehensive explanation of zz coupling in superconducting qubits (arXiv:2408.15402[quant-ph])
[32] Pechal M 2016 Microwave photonics in superconducting circuits *Doctoral Thesis* ETH Zürich (available at: https://doi.org/10.3929/ethz-a-010735338)
[33] Roy C and Wu K 2024 A review of electromagnetics-based microwave circuit design optimization *IEEE Microwave Mag.* **25** 16
[34] Yelten M B, Zhu T, Koziel S, Franzon P D and Steer M B 2012 Demystifying surrogate modeling for circuits and systems *IEEE Circuits Syst. Mag.* **12** 45