

# Scalable self-configuring local Gaussian-process optimizer for variational quantum circuits

Viacheslav Kuzmin<sup>1</sup>, Luca Arceci<sup>2,3</sup>, Alexey Melnikov<sup>1</sup>, Rick van Bijnen<sup>2,3,4</sup>

<sup>1</sup>Terra Quantum AG, 9000 St. Gallen, Switzerland

<sup>2</sup>Institute for Theoretical Physics, University of Innsbruck, A-6020 Innsbruck, Austria

<sup>3</sup>Institute for Quantum Optics and Quantum Information of the Austrian Academy of Sciences, A-6020 Innsbruck, Austria

<sup>4</sup>PlanQC GmbH, 85748 Garching, Germany

vk@terraquantum.swiss

**Motivation.** Training variational quantum circuits [1], QAOA [2], VQE [3], or variational quantum neural networks (QNN) [4] models, is a stochastic optimization problem whose queries are *expensive* and *noisy*: every cost evaluation consumes measurement shots, and the achievable precision scales as the inverse square root of their number. The de-facto standard optimizer, SPSA [5], is simple but very powerful. Many works on new optimization strategies compare against SPSA with default hyperparameters and report substantial improvements [6, 7]. Crucially, however, SPSA performs properly only once its hyperparameters (HPs) are tuned to the specific problem at hand [8, 9, 10], and it is hard to beat SPSA when its HPs are optimal. Yet, tuning is itself a shot-hungry optimization. We present a *scalable local* optimizer for variational quantum circuits based on Gaussian-process model (GPM) surrogate with circuit-specific kernels (Fig. 1) that encodes prior information about a given problem: smoothness, periodicity, or explicitly the number of frequencies along specific dimensions. During the optimization the accumulating noisy observables are used to train the GP model (Fig. 2), which in turn self-consistently guides *every* internal decision (the number and locations of new measurements in full parameter space in each iteration, and the number of shots spent on them), resulting in no overhead for problem-specific hyperparameter optimization. The developed optimizer outperforms or matches SPSA with optimized hyperparameters over the range of tested problems (Fig. 3).

Gaussian-process surrogates have already been applied to variational optimization: as global Bayesian optimizers [11, 12, 13], with kernels matched to the trigonometric structure of the cost function [14, 15], and with posterior-uncertainty-driven shot allocation [16, 17, 18, 19, 20, 21]. These schemes are predominantly global or coordinate-wise and have been limited to few parameters. Our contribution is a *scalable local* optimizer that *unifies* probe placement, probe count, shot budget, and noise calibration into a single tuning-free posterior-driven loop, currently successfully tested on circuits up to 80 parameters (larger problems ongoing).

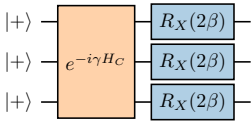
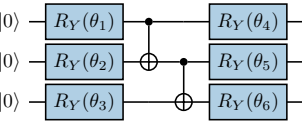
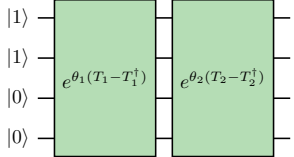
|                        | QAOA                                                                                                                          | HEA                                                                                                      | UCC                                                                                                |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <b>Circuit</b>         |                                            |                      |               |
| <b>Gate properties</b> | $\gamma$ : many frequencies (gaps of $H_C$ )<br>$\Rightarrow$ smooth;<br>$\beta$ : integer frequencies $\Rightarrow$ periodic | each $\theta_i$ : one Pauli rotation,<br>$P^2 = \mathbb{I} \Rightarrow$ single frequency<br>$\omega = 1$ | each $\theta_i$ : paired excitation<br>(Givens) $\Rightarrow$ frequencies<br>$\omega \in \{1, 2\}$ |
| <b>Kernel</b>          | $k = \prod_{p=1}^L e^{-\delta_{\gamma p}^2 / 2\ell_\gamma^2} e^{-2\sin^2(\delta_{\beta p}/2)/\ell_\beta^2}$                   | $k = \prod_i (c_0 + c_1 \cos \delta_i)$                                                                  | $k = \prod_i (c_0 + c_1 \cos \delta_i + c_2 \cos 2\delta_i)$                                       |

Figure 1: Representative circuits of the three ansatz families (QAOA; hardware-efficient ansatz, HEA; unitary coupled cluster, UCC) and the spectral property of each parameterized gate that fixes its kernel factor.  $H_C$  is the problem (cost) Hamiltonian, whose spectral gaps set the continuous  $\gamma$  frequencies;  $R_X, R_Y$  are single-qubit Pauli rotations ( $\omega=1$ );  $T_1, T_2$  are fermionic single-/double-excitation operators, whose anti-Hermitian generators  $T_i - T_i^\dagger$  give Givens rotations with  $\omega \in \{1, 2\}$ . In the kernels,  $\delta$  is the coordinate difference between two points, and the length-scales  $\ell_\gamma, \ell_\beta$  and coefficients  $c_0, c_1, c_2$  are hyperparameters fit by marginal likelihood.

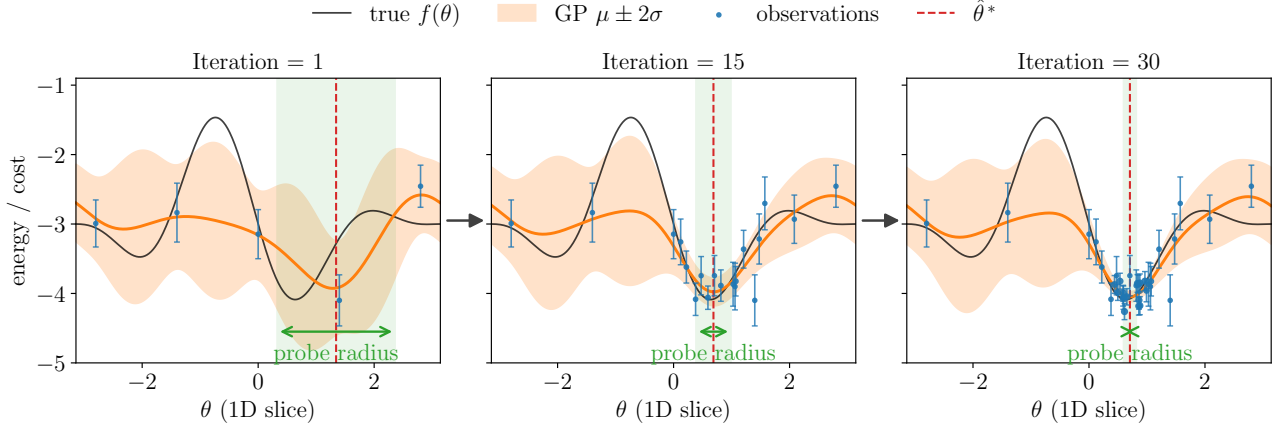


Figure 2: GPM principle on a 1D slice of an example QAOA problem. The GP posterior (orange,  $\mu \pm 2\sigma$ ) with an ansatz-matched kernel is trained on shot-noisy observations (blue; error bars =  $1\sigma$  of the shot noise). The dashed line marks  $\hat{\theta}^*$ , the argmin of the confidence bound; the green double arrow shows the probe radius  $r = 3\sigma[\hat{\theta}^*]$  within which the next D-optimal probes are placed. From iteration 1 to 30 the radius self-contracts onto the basin and the shot rule concentrates measurements there. GPM optimiser identifies best placement of new observables within the characteristic length-scale from the current minima that gives more information on the landscape than direct measurements at the minima.

**Method.** The expectation value of a fixed observable after a parameterized circuit is a trigonometric polynomial in each rotation angle [22], with frequencies fixed by the generators of the parameterized gates. This gives us prior knowledge about the basis functions of the cost function, which we use to *derive* a matching kernel  $k$  for a GP surrogate  $f \sim \mathcal{GP}(k)$  [15, 23] (Fig. 1): single-frequency for Pauli-rotation parameters (Euler angles), periodic for generators with equidistant spectrum (QAOA mixer), two-frequency cosine factors for unitary-coupled-cluster (UCC) pair excitations, and a squared-exponential factor for continuous-spectrum smooth cost angles (e.g., a QAOA Hamiltonian). We model shot noise in the GPM kernel as heteroscedastic Gaussian noise  $\sigma_i^2 = C/n_i$  for an observation averaged over  $n_i$  shots; the constant  $C$  is *self-calibrated* from the sample variances of the warm-up observations, which are budget-counted and added into GP training set.

During the optimization the GPM is fed with observations obtained by sampling a circuit, trained, and in turn provides the posterior mean  $\mu(\theta)$ , variance  $\sigma^2(\theta)$ , and the analytic gradient covariance  $\Sigma_g(\theta)$  of the derivative GP used to perform next optimization iteration (Fig. 2):

1. **Argmin estimate:**  $\hat{\theta}^* = \arg \min_{\theta} \mu(\theta) + \kappa \sigma(\theta)$  (lower-confidence bound,  $\kappa = 1$ ), a *noise-free* optimization of the surrogate.
2. **Define trust radius:**  $r = 3\sigma[\hat{\theta}^*]$ , the posterior uncertainty of the *location* of the minimum (Laplace approximation). The radius self-expands when the model is uncertain and self-contracts as the basin is identified.
3. **Observables placement:** the next  $N$  measurement points are chosen to minimize  $\log \det \Sigma_g(\hat{\theta}^*)$ , a D-optimal experiment design that asks “where should I measure to learn the gradient at the current optimum best?” used as a proxy for the landscape refining near the suggested optimum  $\hat{\theta}^*$ .
4. **Choice of the probe number and shots:**  $N$  is tested on a small grid by the same information objective to find optimal number of observations to make, and the total shot budget spent in the iteration on these  $N$  measurements follows a posterior-matching rule  $n = C/\sigma^2(\hat{\theta}^*)$ : early iterations are cheap and exploratory, late iterations concentrate shots where precision pays (see Fig. 2). Notably, the selected  $N \approx 6\text{--}24$  does *not* scale with the parameter count: the GP aggregates information across iterations, so per-iteration observations need only refresh the local model, in contrast to the  $\mathcal{O}(D)$  evaluations of finite differences.

Old observations are retired from the GP kernel through a “First In, First Out” buffer to keep the computation of the predictions tractable, and kernel hyperparameters (amplitude, length-scales) are

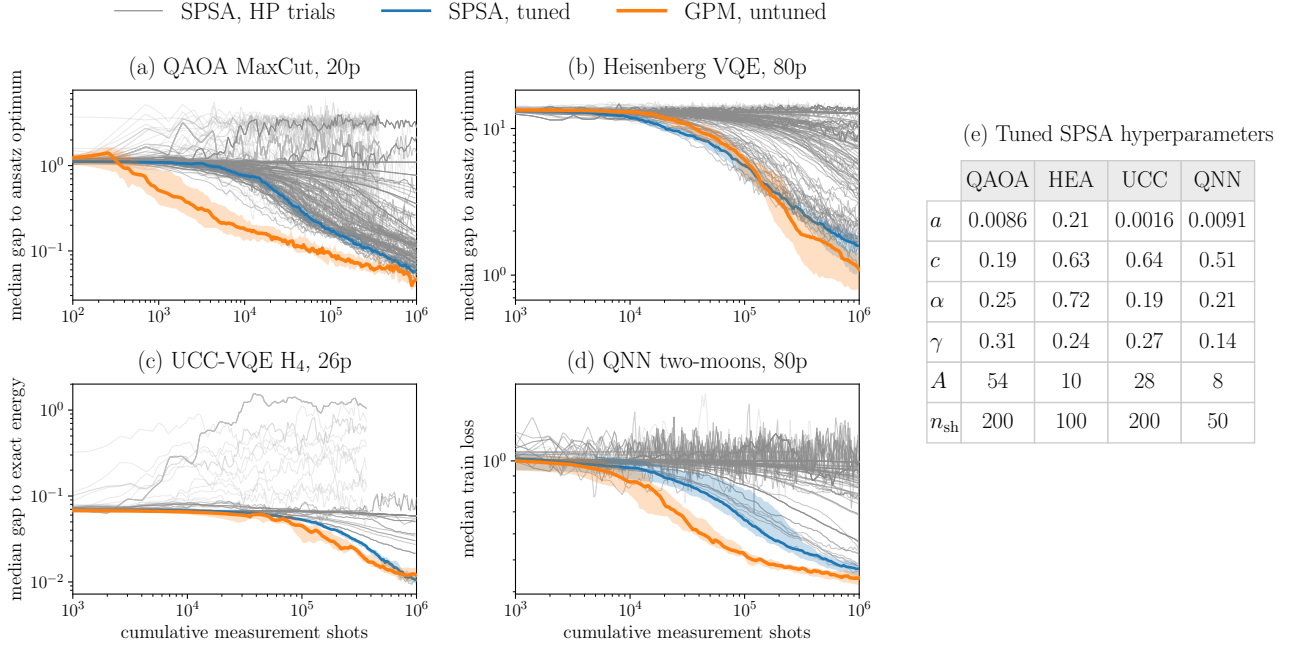


Figure 3: Gray: individual SPSA HP-tuning trajectories with target  $10^6$  shots budget for each trajectory (one per Optuna trial, 3 seeds-averaged), whose lower envelope is the tuned optimum (blue, 16-seed median  $\pm$  IQR); orange: untuned GPM optimizer (16-seed median  $\pm$  IQR). Untuned GPM optimizer tracks or beats the tuned SPSA optimum throughout and reaches a given accuracy in consistently fewer shots, matching the tuned endpoint on all four problems. (e) The resulting tuned SPSA HPs (gain  $a$ , perturbation  $c$ , their decay exponents  $\alpha, \gamma$ , stability constant  $A$  in  $a_k = a/(A+k+1)^\alpha$ , and shots per gradient step  $n_{\text{sh}}$ ) vary markedly across problems, being Optuna-tuned per problem; GPM optimizer instead uses one configuration throughout, identical up to the ansatz-derived kernel.

refit by marginal likelihood on the fly. The method has no step-size schedule, no decay exponents, and no problem-dependent knobs: every decision is information-driven, adapting to the empirically observed landscape of the current problem and iteration.

**Benchmarks.** We compare the GPM optimizer against SPSA with HPs tuned per problem by dedicated Optuna [24] searches (tens to over a hundred trials each), at a uniform  $10^6$ -shot budget on four problems spanning combinatorial optimization, VQE for a quantum many-body system, chemistry, and QNN (Fig. 3). Untuned GPM optimizer matches or exceeds reasonably tuned SPSA on every problem, with two advantages that require no tuning. First, *shot efficiency*: GPM optimizer reaches a given accuracy in consistently fewer shots (e.g. it attains tuned SPSA’s final QNN training loss with about  $3\times$  fewer shots), and its single untuned trajectory tracks or sits below the best SPSA trials of the *entire* tuning ensemble (Fig. 3). Second, *hyperparameter robustness*: GPM optimizer is identical across problems up to the ansatz-derived kernel, so it never pays SPSA’s per-problem tuning cost and cannot be mis-tuned.

**Outlook.** GPM turns variational-circuit training into sequential Bayesian experiment design in which sampling effort itself is a controlled variable. Because all decisions derive from the posterior, the framework extends naturally to measured-variance-aware allocation, principled early stopping via the argmin-value uncertainty, and batch/shot-noise unification in QNN training (our two-moons benchmark treats finite-batch noise and shot noise through a single self-calibrated noise scale). We will present the method, the benchmark suite, and ablations isolating which information channels carry the gains.

## References

- [1] M. Cerezo *et al.*, *Nat. Rev. Phys.* **3**, 625 (2021).
- [2] E. Farhi, J. Goldstone, S. Gutmann, arXiv:1411.4028 (2014).

- [3] A. Peruzzo *et al.*, *Nat. Commun.* **5**, 4213 (2014).
- [4] M. Schuld and F. Petruccione, *Machine Learning with Quantum Computers*, Springer (2021).
- [5] J. C. Spall, *IEEE Trans. Autom. Control* **37**, 332 (1992).
- [6] A. Pellow-Jarman, I. Sinayskiy, A. Pillay, F. Petruccione, *Quantum Inf. Process.* **20**, 202 (2021).
- [7] M. Wiedmann *et al.*, *IEEE Int. Conf. Quantum Comput. Eng. (QCE)* (2023).
- [8] X. Bonet-Monroig *et al.*, *Phys. Rev. A* **107**, 032407 (2023).
- [9] B. D. M. Jones, L. Mineh, A. Montanaro, arXiv:2411.13742 (2024).
- [10] L. Tecot, C. Hsieh, arXiv:2310.09468 (2023).
- [11] S. Tibaldi, D. Vodola, E. Tignone, E. Ercolessi, *IEEE Trans. Quantum Eng.* **4**, 1 (2023).
- [12] D. R. Jones, *J. Global Optim.* **21**, 345 (2001).
- [13] F. Sorourifar, D. Chamaki, N. M. Tubman, J. A. Paulson, D. E. Bernal Neira, arXiv:2406.14627 (2024).
- [14] K. A. Nicoli, C. J. Anders, L. Funcke, T. Hartung, K. Jansen, S. Kühn, K.-R. Müller, P. Stornati, P. Kessel, S. Nakajima, *Adv. Neural Inf. Process. Syst. (NeurIPS)* (2023); arXiv:2406.06150.
- [15] L. Arceci, V. Kuzmin, R. van Bijnen, Gaussian Process Model Kernels for Noisy Optimization in Variational Quantum Algorithms (2025).
- [16] C. J. Anders, K. A. Nicoli, B. Wu *et al.*, arXiv:2502.01704 (2025).
- [17] S. Tamiya and H. Yamasaki, *npj Quantum Inf.* **8**, 90 (2022).
- [18] J. Müller, W. Lavrijsen, C. Iancu, W. de Jong, *IEEE Int. Conf. Quantum Comput. Eng. (QCE)* (2022), p. 215.
- [19] R. Shaffer, L. Kocia, M. Sarovar, *Phys. Rev. A* **107**, 032415 (2023).
- [20] J. M. Kübler, A. Arrasmith, L. Cincio, P. J. Coles, *Quantum* **4**, 263 (2020).
- [21] K. J. Sung *et al.*, *Quantum Sci. Technol.* **5**, 044008 (2020).
- [22] M. Schuld, R. Sweke, J. J. Meyer, *Phys. Rev. A* **103**, 032430 (2021).
- [23] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press (2006).
- [24] T. Akiba *et al.*, Optuna, *KDD* (2019).