

PPO-QGP: Automated Quantum Kernel Design for Gaussian Process Regression

Kerem Bükrü^{1,*}, Ralf Sturm¹, Tjark Siefkes¹

¹Institute of Vehicle Concepts, German Aerospace Center (DLR),
Pfaffenwaldring 38–40, 70569 Stuttgart, Germany

*Corresponding author: kerem.buekrue@dlr.de

Abstract. Gaussian process regression is a widely used kernel-based supervised machine learning method. The kernel choice highly affects the regression quality and usually requires extensive hyperparameter tuning. Quantum fidelity kernels have emerged as expressive alternatives for regression tasks. However, designing a suitable circuit requires domain-specific knowledge and is often done heuristically. We propose a fully automatic quantum kernel architecture search routine for Gaussian process regression, based on proximal policy optimization. Unlike previously proposed approaches that only optimize the gate sequence and number of layers, our method additionally optimizes the number of qubits, enabling a significantly larger search space. We define a custom reward function that directly minimizes the negative log marginal likelihood. Furthermore, we demonstrate that our method generates quantum kernels that surpass established benchmark kernels found in the literature on a synthetic regression problem.

Keywords: quantum machine learning, quantum architecture search, quantum kernels, Gaussian processes, reinforcement learning

1 Introduction

Gaussian processes (GPs) are a probabilistic class of machine learning models that utilize Bayesian statistics to compute predictive distributions with uncertainty estimates (Williams and Rasmussen, 2006). Their non-parametric nature makes them highly flexible and useful for discovering complex patterns in data. The output of a GP is governed by its kernel function, which determines both the model’s expressivity and regression quality (Duvenaud et al, 2013; Wilson and Adams, 2013). Finding a suitable kernel for a given regression problem is non-trivial as it must be expressive enough to capture the underlying structure of the data. Quantum fidelity kernels have emerged as expressive alternatives since they operate in high-dimensional Hilbert spaces (Havlíček et al, 2019; Schuld and Killoran, 2019; Rapp and Roth, 2024). These kernels are defined by quantum feature maps, whose architectures are typically chosen heuristically, motivating the development of automated architecture search methods.

Automatic quantum architecture search has been explored using a variety of methods, such as model-based reinforcement learning (Rapp et al, 2025), proximal policy optimization (PPO) (Kuo et al, 2021; Foderà et al, 2024; Zhu and Hou, 2023), SARSA (Incudini et al, 2024), LLM-based optimization (Sakka et al, 2025), and hybrid genetic optimization (Pellow-Jarman et al, 2024). However, none of these approaches optimize the number of qubits, which restricts the search space and thus the expressivity of the resulting quantum kernels.

We introduce PPO-QGP, a data-driven approach that automatically optimizes the architecture of quantum fidelity kernels for Gaussian process regression (GPR). Our method optimizes the gate sequence, the number of layers, and the number of qubits of a quantum feature map that defines the quantum fidelity kernel. We demonstrate that our method produces quantum kernels with a resulting negative log marginal likelihood (LML) that is lower than that of established quantum kernels from the literature, for a benchmark test function.

2 Preliminaries

2.1 Gaussian process regression

A GP is a collection of random variables following a multivariate normal distribution (Williams and Rasmussen, 2006). Given a training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ with inputs $\mathbf{x}_i \in \mathbb{R}^d$ and corresponding

targets $y_i \in \mathbb{R}$, GPR aims to learn a latent function $f : \mathbb{R}^d \rightarrow \mathbb{R}$. The training samples are generated as $y_i = f(\mathbf{x}_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$ denotes Gaussian noise with variance σ_ϵ^2 . A GP is fully defined by its mean function $m(\mathbf{x})$, and covariance function $k(\mathbf{x}, \mathbf{x}')$, where the latter is often referred to as the kernel. The posterior distribution is obtained by conditioning the prior on the data, yielding the predictive mean and covariance

$$\mathbb{E}[f_*] = \mathbf{k}_*^\top (\mathbf{K} + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{y}, \quad \mathbb{V}[f_*] = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top (\mathbf{K} + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{k}_*. \quad (1)$$

The model is trained by maximizing the LML

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2} \mathbf{y}^\top (\mathbf{K} + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{y} - \frac{1}{2} \log \det (\mathbf{K} + \sigma_\epsilon^2 \mathbf{I}) - \frac{N}{2} \log 2\pi, \quad (2)$$

with respect to the kernel hyperparameters, which we denote as $\boldsymbol{\theta}$.

2.2 Quantum kernels

A quantum feature map $\phi : \mathcal{X} \rightarrow \mathcal{H}$ embeds a classical data point $\mathbf{x}$ into a quantum state $|\phi(\mathbf{x})\rangle = U(\mathbf{x}, \boldsymbol{\theta})|0\rangle$, in the 2^n -dimensional Hilbert space of an n -qubit quantum circuit. Computing the overlap between two embedded states defines the quantum fidelity kernel as the Hilbert-Schmidt inner product

$$k(\mathbf{x}, \mathbf{x}') = |\langle \phi(\mathbf{x}') | \phi(\mathbf{x}) \rangle|^2, \quad (3)$$

whose parameters $\boldsymbol{\theta}$ are optimized by maximizing the LML (Eq. (2)).

2.3 Proximal policy optimization

PPO (Schulman et al, 2017) is a model-free reinforcement learning algorithm, which utilizes a clipped objective function to avoid overly large policy updates

$$L^{\text{CLIP}}(\psi) = \mathbb{E}_t \left[\min \left(r_t(\psi) \hat{A}_t, \text{clip}(r_t(\psi), 1 - \delta, 1 + \delta) \hat{A}_t \right) \right], \quad (4)$$

where δ denotes the clipping parameter, $r_t(\psi) = \pi_\psi(a_t|s_t)/\pi_{\psi_{\text{old}}}(a_t|s_t)$ is the probability ratio between the current and previous policy, and $\hat{A}_t$ is the estimated advantage at timestep t .

3 PPO-QGP

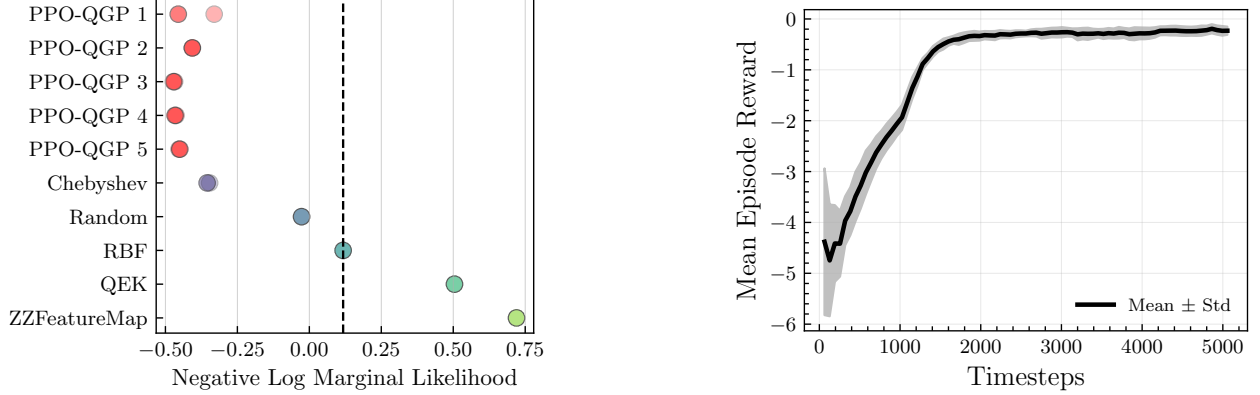
We consider a quantum kernel environment in which each circuit is characterized by its gate sequence $(g_1, \dots, g_T)$ with $g_t \in \mathcal{G}$ (see Table 1), the number of qubits n , and the number of layers L . Parameterized gates either use only a trainable parameter $R_\alpha(\theta)$, or also incorporate data encoding $R_\alpha(\theta \varphi(x))$, where $\varphi(x) = \arccos(x)$ (Kyriienko et al, 2021), and $\theta \in \boldsymbol{\theta}$. The observation space $\mathcal{O} = \mathcal{O}_{\text{gate}} \times \mathcal{O}_{\text{qubit}} \times \mathcal{O}_{\text{layer}}$, with $\mathcal{O}_{\text{gate}} = [-1, 1]^T$ and $\mathcal{O}_{\text{qubit}}, \mathcal{O}_{\text{layer}} \in [-1, 1]$, encodes the gate sequence $(g_1, \dots, g_T)$, the number of qubits n , and the number of layers L , where T denotes the maximum number of timesteps per episode. Gate positions are initialized with -1 and filled incrementally. Gate types are integer encoded, while qubit and layer counts are log-scaled, with all features ultimately mapped to the interval $[-1, 1]$. The action space $\mathcal{A} = \mathcal{A}_{\text{gate}} \cup \mathcal{A}_{\text{qubit}} \cup \mathcal{A}_{\text{layer}}$ consists of three sub-spaces: $\mathcal{A}_{\text{gate}}$ selects a gate $g_t \in \mathcal{G}$ and applies it to the circuit, where single-qubit gates are applied to all qubits and two-qubit gates to neighboring qubit pairs from top to bottom. $\mathcal{A}_{\text{qubit}}$ adds a qubit to the circuit, and $\mathcal{A}_{\text{layer}}$ appends a new layer. We use a reward function consisting of three components

$$r_t = r_t^{\text{lml}} + r_t^{\text{bon}} - r_t^{\text{pen}}, \quad (5)$$

where $r_t^{\text{lml}} := \log p(\mathbf{y}|\mathbf{X})$ denotes the current LML value. The bonus term equals $r_t^{\text{bon}} = 1$ if a new lowest negative LML value is reached, and 0 otherwise. Furthermore, a penalty of $r_t^{\text{pen}} = 0.1$ is applied if the circuit does not contain an encoding gate (Rapp et al, 2025). After each circuit modification, the resulting quantum fidelity kernel is used for GPR, where the negative LML is minimized to compute r_t^{lml} .

Table 1 Gate set $\mathcal{G}$ ($G = 18$ gates).

Single-qubit gates			Two-qubit gates		
X	$R_x(\theta)$	$R_x(\theta \varphi(x))$	CX	$CR_x(\theta)$	$CR_x(\theta \varphi(x))$
Y	$R_y(\theta)$	$R_y(\theta \varphi(x))$	CY	$CR_y(\theta)$	$CR_y(\theta \varphi(x))$
Z	$R_z(\theta)$	$R_z(\theta \varphi(x))$	CZ	$CR_z(\theta)$	$CR_z(\theta \varphi(x))$

**Fig. 1** Left: negative LML values across all models (lower is better). Each model is evaluated three times with independently initialized kernel parameters. Five PPO-QGP variants with different PPO seeds are shown in red. The dashed vertical line denotes the negative LML value of the RBF kernel, which serves as a classical baseline. Right: mean episode reward over all five PPO-QGP seeds. The shaded region denotes $\pm$ one standard deviation.

4 Results

We compare the performance of our PPO-QGP method to the ZZFeatureMap (Javadi-Abhari et al, 2024), the Chebyshev feature map (Kyriienko et al, 2021; Rapp and Roth, 2024), the quantum embedding kernel (QEK) (Hubregtsen et al, 2022), the classical RBF kernel, and a random baseline kernel. The benchmark quantum kernels use a fixed architecture of 4 qubits and 2 layers. The random baseline kernel samples a gate sequence of length four uniformly from $\mathcal{A}$. Note that the ZZFeatureMap has no trainable parameters, yielding identical LML values across all evaluation runs. All experiments use a fixed dataset of $N = 10$ training points, sampled from the test function $f(x) = x \sin(x)$ with a noise variance of $\sigma_\epsilon^2 = 0.01$, with both inputs and targets scaled to $[-1, 1]$.

For the likelihood optimization, we use Adam (Kingma and Ba, 2014) with a learning rate of 0.1, a maximum of 500 optimization steps, and a patience parameter of 5. The PPO agent uses an entropy coefficient of 0.05, a value function coefficient of 0.5, a clipping parameter $\delta = 0.2$, a batch size of 32, $T = 10$ actions per episode, 64 policy optimization steps, and 5000 total environment timesteps. To ensure robustness to training variability, we run five independent PPO seeds.

The resulting negative LML values are shown in Fig. 1 (left). PPO-QGP outperforms all benchmark kernels. The ZZFeatureMap and QEK kernels perform worst, followed by the RBF and random kernels. The Chebyshev kernel has a low negative LML, yet every PPO-QGP seed surpasses it when the best LML optimization run is considered. The averaged mean episode reward for the best run of each PPO-QGP model is shown in Fig. 1 (right). The initial mean reward value is below -4 , and it gradually increases, plateauing near 0 after approximately 2000 timesteps, indicating that the agent has learned a stable policy.

5 Conclusion

We presented PPO-QGP, a model-free, reinforcement learning-based framework for the automated design of quantum fidelity kernels for GPR. The PPO agent jointly optimizes the gate sequence, number of layers, and number of qubits, without requiring domain-specific knowledge. We demonstrated on the benchmark function $f(x) = x \sin(x)$ that PPO-QGP consistently outperforms established quantum fidelity kernels from the literature. Future work will extend the evaluation to multi-dimensional input spaces, larger datasets, and additional benchmark functions, which we expect to further demonstrate the generalization capability of PPO-QGP.

References

- Duvenaud D, Lloyd J, Grosse R, et al (2013) Structure discovery in nonparametric regression through compositional kernel search. In: International Conference on Machine Learning, PMLR, pp 1166–1174
- Foderà S, Turati G, Nembrini R, et al (2024) Reinforcement learning for variational quantum circuits design. arXiv preprint arXiv:240905475
- Havlíček V, Córcoles AD, Temme K, et al (2019) Supervised learning with quantum-enhanced feature spaces. *Nature* 567(7747):209–212. <https://doi.org/10.1038/s41586-019-0980-2>
- Hubregtsen T, Wierichs D, Gil-Fuster E, et al (2022) Training quantum embedding kernels on near-term quantum computers. *Physical Review A* 106(4):042431
- Incudini M, Bosco DL, Martini F, et al (2024) Automatic and effective discovery of quantum kernels. *IEEE Transactions on Emerging Topics in Computational Intelligence* pp 1–10. <https://doi.org/10.1109/TETCI.2024.3499993>, 2209.11144 [quant-ph]
- Javadi-Abhari A, Treinish M, Krsulich K, et al (2024) Quantum computing with Qiskit. <https://doi.org/10.48550/arXiv.2405.08810>, 2405.08810
- Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
- Kuo EJ, Fang YLL, Chen SYC (2021) Quantum architecture search via deep reinforcement learning. arXiv preprint arXiv:210407715
- Kyriienko O, Paine AE, Elfving VE (2021) Solving nonlinear differential equations with differentiable quantum circuits. *Phys Rev A* 103:052416. <https://doi.org/10.1103/PhysRevA.103.052416>, URL <https://link.aps.org/doi/10.1103/PhysRevA.103.052416>
- Pellow-Jarman R, Pillay A, Sinayskiy I, et al (2024) Hybrid genetic optimization for quantum feature map design. *Quantum Machine Intelligence* 6(2):45. <https://doi.org/10.1007/s42484-024-00177-w>
- Rapp F, Roth M (2024) Quantum gaussian process regression for bayesian optimization. *Quantum Machine Intelligence* 6(1):5
- Rapp F, Kreplin DA, Huber MF, et al (2025) Reinforcement learning-based architecture search for quantum machine learning. *Machine Learning: Science and Technology* 6(1):015041. <https://doi.org/10.1088/2632-2153/adaf75>, 2406.02717 [quant-ph]
- Sakka K, Mitarai K, Fujii K (2025) Automating quantum feature map design via large language models. <https://doi.org/10.48550/arXiv.2504.07396>, 2504.07396
- Schuld M, Killoran N (2019) Quantum Machine Learning in Feature Hilbert Spaces. *Physical Review Letters* 122(4):040504. <https://doi.org/10.1103/PhysRevLett.122.040504>
- Schulman J, Wolski F, Dhariwal P, et al (2017) Proximal policy optimization algorithms. arXiv preprint arXiv:170706347
- Williams CK, Rasmussen CE (2006) Gaussian processes for machine learning, vol 2. MIT press Cambridge, MA
- Wilson A, Adams R (2013) Gaussian process kernels for pattern discovery and extrapolation. In: International conference on machine learning, PMLR, pp 1067–1075
- Zhu X, Hou X (2023) Quantum architecture search via truly proximal policy optimization. *Scientific Reports* 13(1):5157