A canonical compendium of quantum machine learning code, in the year before fault tolerance

Quantum machine learning sits, in 2026, at an unusually candid inflection point: the field has matured enough to police its own claims of advantage, while the hardware beneath it has finally crossed below the surface-code threshold. The most-cited recent benchmarking work — **Bowles, Ahmed and Schuld's 2024 "Better than classical?"** — concluded, after running twelve QML models across 160 datasets, that out-of-the-box classical baselines generally win and that **removing entanglement frequently leaves performance unchanged or improves it**. In the same window, **Quantinuum and Microsoft demonstrated twelve logical qubits with end-to-end chemistry**, and **QuEra, Harvard and MIT operated 48 logical qubits**. The intellectual stakes have shifted accordingly: from "does QML work?" to "where, precisely, will it work, and on what hardware regime?" ADS + 2↗

This compendium is built for that question. It is framework-agnostic — PennyLane, Qiskit, Cirq, TensorFlow Quantum, Braket, Yao.jl, Strawberry Fields, Tequila, Mitiq, lambeq and the Julia/Rust/JAX-native frontiers all appear — and it is constructed for posterity. Several of its most valuable entries are repositories already at risk of being lost: **Peter Wittek's MOOC code** preserved by the Quantum Open Source Foundation after his death in a 2019 Himalayan avalanche; **Rigetti's Grove**, the original public reference for VQE and QAOA, archived since 2019; the **Zapata Computing repositories**, frozen in place after the company's October 2024 wind-down and partial 2025 re-emergence; **Microsoft LIQUi|⟩**, the F# predecessor to Q#; the deprecated **qiskit-aqua** API in which the first qGAN, QSVM and VQC were written. A field that loses its early code loses its memory; this document tries to keep that memory legible. GitHub↗ Wikipedia↗

What follows is organized to be read as a reference and skimmed as a map. Section 1 sets the framework substrate. Sections 2–4 catalogue research-lab and algorithm-specific repositories. Sections 5–7 cover the literature, datasets and institutions. Section 8 surveys the 2024–2026 frontier. Section 9 is the explicit recovery layer for at-risk code. Section 10 is the practical onboarding kit. A closing essay locates the whole list against the next five years.

---

1. The framework substrate

The QML stack in 2026 has consolidated around roughly a dozen frameworks, each with a distinct philosophical commitment — autodifferentiation, hardware fidelity, photonic continuous variables, tensor-network bridges or fault-tolerant resource estimation. The list below is the operational core of the field; almost every paper or notebook elsewhere in this compendium imports from one of these.

Differentiable, QML-first frameworks. PennyLane (Xanadu) — https://github.com/PennyLaneAI/pennylane — remains the de facto reference: cross-platform, autograd/JAX/PyTorch/TF compatible, with plugins to twenty hardware backends. The companion **PennyLane demos** at https://github.com/PennyLaneAI/qml and the **PennyLane Codebook** at https://pennylane.ai/codebook are arguably the single best entry points for serious learners. **PennyLane Lightning** (https://github.com/PennyLaneAI/pennylane-lightning) provides the C++/CUDA/Kokkos/ROCm/MPS backends that ran QML on Frontier-class HPC; **Catalyst** (https://github.com/PennyLaneAI/catalyst) is the MLIR-based JIT compiler that brings `@qjit` semantics to hybrid programs and is published in JOSS at https://doi.org/10.21105/joss.06720. GitLab + 5↗

The IBM stack. Qiskit 2.x (https://github.com/Qiskit/qiskit) introduced a stable C API for HPC integration in 2025. The QML application module, **Qiskit Machine Learning** (https://github.com/qiskit-community/qiskit-machine-learning, co-maintained with the UK Hartree Centre), supplies `FidelityQuantumKernel`, `QSVC`/`QSVR`, `EstimatorQNN`/`SamplerQNN`, `VQC`/`VQR` and the PyTorch `TorchConnector`. It sits beside **Qiskit Nature** (chemistry), **Qiskit Optimization** (QAOA, Grover-Optimizer), **Qiskit Algorithms** and **Qiskit Aer** (simulator). Note the **Qiskit textbook** at https://github.com/qiskit-community/qiskit-textbook is now archived in favour of https://github.com/Qiskit/textbook and the live course at https://learning.quantum.ibm.com. GitHub + 4↗

Google. The **quantumlib** organisation at https://github.com/quantumlib hosts **Cirq**, **OpenFermion**, **qsim**, **ReCirq**, **Stim** and the fault-tolerance resource-estimation library **Qualtran**. **TensorFlow Quantum**

(https://github.com/tensorflow/quantum), built on Cirq and Keras, is still maintained but at a slower cadence; the Hamiltonian-based-models companion **QHBM** lives at https://github.com/google/qhbm-library. GitHub + 5↗

Other vendor stacks. Microsoft has consolidated its QDK at https://github.com/microsoft/qsharp and https://github.com/microsoft/qdk; the popular **Quantum Katas** (https://github.com/microsoft/QuantumKatas) were archived in August 2024 and migrated into the modern QDK. **Amazon Braket** SDKs and example notebooks live under https://github.com/amazon-braket — particularly https://github.com/amazon-braket/amazon-braket-examples (with QML-in-Hybrid-Jobs, QAOA-PennyLane, QN-SPSA and QCBM tutorials). **Rigetti's pyQuil** is at https://github.com/rigetti/pyquil. **NVIDIA CUDA-Q** (https://github.com/NVIDIA/cuda-quantum) and **cuQuantum** (https://github.com/NVIDIA/cuQuantum) supply the GPU backends now powering Lightning, qsim and Aer; the academic curriculum is at https://github.com/NVIDIA/cuda-q-academic, and the QEC/solvers/dynamics extensions at https://github.com/NVIDIA/cudaqx. GitHub + 5↗

Photonic continuous-variable. Strawberry Fields (https://github.com/XanaduAI/strawberryfields) accepts only bug fixes and security patches now — Xanadu's QML focus has shifted to PennyLane — but the older photonic-QNN demos at https://strawberryfields.ai/photonics/index.html should be considered at-risk material worth snapshotting. **MrMustard** (https://github.com/XanaduAI/MrMustard) is the differentiable Gaussian/photonic successor. The Quandela stack — **Perceval** (https://github.com/Quandela/Perceval), the **MerLin** photonic-QML framework demonstrated in the **HybridAIQuantum-Challenge** (https://github.com/Quandela/HybridAIQuantum-Challenge), the **photonic-qgan** and **photonic-qcbm** repositories under the same organisation — is the most active alternative photonic stack. GitHub + 2↗

Neutral-atom, ion-trap and annealing. QuEra Bloqade (Julia: https://github.com/QuEraComputing/Bloqade.jl; Python: https://github.com/QuEraComputing/bloqade), with the dedicated **Quantum Reservoir Computing tutorials** at https://github.com/QuEraComputing/QRC-tutorials. **Pasqal** offers **Pulser** (https://github.com/pasqal-io/Pulser), the partly-paused **Qadence** (https://github.com/pasqal-io/qadence), the JAX simulator **horqrux** (https://github.com/pasqal-io/horqrux) and the graph-QML kernel library **QEK** (https://github.com/pasqal-io/qek). **Quantinuum/CQCL** publishes https://github.com/CQCL/lambeq (QNLP), https://github.com/CQCL/tket (compiler) and the next-generation Rust rewrite https://github.com/CQCL/tket2. **D-Wave Ocean** (https://github.com/dwavesystems/dwave-ocean-sdk) covers annealing-based QML. JuliaHub + 9↗

Julia, JAX and high-performance simulators. Yao.jl (https://github.com/QuantumBFS/Yao.jl) is the Julia reference, fully differentiable with GPU support; **Qulacs** (https://github.com/qulacs/qulacs) the C++/Python speed leader; **Qibo** (https://github.com/qiboteam/qibo) JIT-compiled and hardware-accelerated; **Qrack** (https://github.com/unitaryfund/qrack) the LGPL high-performance C++ simulator; **TensorCircuit** (https://github.com/tencent-quantum-lab/tensorcircuit) the multi-backend tensor-network framework; **TorchQuantum** from MIT-HAN-Lab (https://github.com/mit-han-lab/torchquantum), a Unitary Fund-supported PyTorch-native framework that powered the winning ACM Quantum Computing for Drug Discovery entry. GitHub + 6↗

Cross-cutting frameworks. Tequila (https://github.com/tequilahub/tequila) provides a backend-agnostic VQA abstraction layer atop Qulacs/Qiskit/Cirq/PyQuil; **Mitiq** from the **Unitary Foundation** (https://github.com/unitaryfoundation/mitiq, formerly unitaryfund/mitiq) reached v1.0 in 2024–2025 with ZNE, PEC, DDD, LRE, CDR, REM and PT mitigations; the **Unitary Compiler Collection** at https://github.com/unitaryfoundation/ucc is the new frontend-agnostic compiler from the same foundation. **Classiq's** algorithm library is open at https://github.com/Classiq/classiq-library; **Covalent** (https://github.com/AgnostiqHQ/covalent) is the dominant orchestration layer for hybrid workflows across cloud, Slurm and Braket Hybrid Jobs. GitHub + 8↗

---

2. Awesome-lists, curated indices and discovery surfaces

Treat the following four lists as the field's living memory; they are the closest thing QML has to a self-updating bibliography.

The KDnuggets seed article — https://www.kdnuggets.com/5-github-repositories-to-learn-quantum-machine-learning — points to **Krishnakumar Sekar's awesome-quantum-machine-learning** (https://github.com/krishnakumarsekar/awesome-quantum-machine-learning, ~3.2k stars; companion site at https://krishnakumarsekar.github.io/awesome-quantum-machine-

learning/), the broad table of contents of the field; **Antoine Mougel's tightly-curated awesome-quantum-ml** (https://github.com/artix41/awesome-quantum-ml, paper-first, CC0); the **QOSF awesome-quantum-software** (https://github.com/qosf/awesome-quantum-software with the project index at https://qosf.org/project_list/), the canonical OSS directory; and Désirée Vogt-Lee's **awesome-quantum-computing** (https://github.com/desireevl/awesome-quantum-computing) for learning resources. Sekar's list and the QOSF project list together form the most reliable baseline. GitHub + 5↗

Two complementary discovery surfaces matter for ongoing work: the live GitHub topic page at https://github.com/topics/quantum-machine-learning, sortable by stars and recency, and **Stephen Jordan's Quantum Algorithm Zoo** at https://quantumalgorithmzoo.org for algorithm-level orientation. The **PennyLane demos** index at https://pennylane.ai/qml is the most pedagogically organised editorial aggregator; the **TensorNetwork ML hub** at https://tensornetwork.org/ml/ does the same for tensor-network methods.

Three further entries deserve note. The **KDnuggets article's** other four picks — https://github.com/artix41/awesome-quantum-ml, https://github.com/quantum-machine-learning/Hands-On-Quantum-Machine-Learning-With-Python-Vol-1, https://github.com/PatrickHuembeli/Quantum-Machine-Learning-on-Near-Term-Quantum-Devices and https://github.com/qiskit-community/qiskit-machine-learning — are all live and useful, but the article underweights several primary frameworks (PennyLane, TFQ, lambeq) which this compendium restores. **Christophe Pere's Roadmap-to-QML** at https://github.com/Christophe-pere/Roadmap-to-QML is the best single curriculum sequence for a self-learner. The **bithabib/quantum_machine_learning** community list (https://github.com/bithabib/quantum_machine_learning) supplements the QOSF index with smaller, less-promoted academic repos. KDnuggets↗ GitHub↗

---

3. The major laboratories and their public code

Each significant industrial and academic group now maintains a recognisable GitHub footprint, and the *shape* of that footprint matters: it tells you whether a lab is investing in research code, hardware compilers, education, or none of the above.

Industry

Google Quantum AI at https://github.com/quantumlib spans Cirq, OpenFermion, ReCirq, qsim, Stim and Qualtran (https://github.com/quantumlib/Qualtran for fault-tolerant resource estimation). **TensorFlow Quantum** (https://github.com/tensorflow/quantum) lives separately under the TensorFlow organisation. **IBM Quantum** at https://github.com/Qiskit and https://github.com/qiskit-community completed the 2023 split that moved application modules out of the core SDK; **qiskit-machine-learning** is the QML flagship. **Xanadu** straddles two organisations: https://github.com/XanaduAI for the photonic stack and https://github.com/PennyLaneAI for the QML library. **AWS Braket** rebranded its repositories from `aws/amazon-braket-*` to https://github.com/amazon-braket; the algorithm library at https://github.com/amazon-braket/amazon-braket-algorithm-library and the Qiskit-Braket bridge at https://github.com/amazon-braket/qiskit-braket-provider are essential. **Microsoft Azure Quantum** modernised its stack at https://github.com/microsoft/qsharp; the legacy https://github.com/microsoft/Quantum and https://github.com/microsoft/QuantumLibraries remain online for archaeology. GitHub↗

Rigetti post-restructuring shows reduced activity at https://github.com/rigetti; pyQuil is maintained, **Grove** (https://github.com/rigetti/grove) is archived. **Quantinuum/CQCL** at https://github.com/CQCL is among the most active, with lambeq as its flagship QML asset. **QuEra** (https://github.com/QuEraComputing) is pivoting from Julia-first to Python-first SDKs; **Pasqal** (https://github.com/pasqal-io, ~68 repositories) is unusually transparent for a hardware vendor. **IonQ** publishes mostly samples at https://github.com/ionq-samples — note especially https://github.com/ionq-samples/qsharp-machine-learning. **Quandela** at https://github.com/Quandela has published full photonic-QML pipelines. **D-Wave** at https://github.com/dwavesystems remains the canonical annealing-ML stack. **Classiq's** library at https://github.com/Classiq/classiq-library, **NVIDIA's** https://github.com/NVIDIA/cuda-quantum, **PsiQuantum's** resource-estimator https://github.com/PsiQ/bartiq, and **Q-CTRL's** open-controls library at https://github.com/qctrl/open-controls round out the industry list. GitHub + 2↗

Two corporate notes for posterity. **Zapata Computing** ceased operations in October 2024 and re-emerged as **Zapata Quantum, Inc.** with $15M in financing in 2026; the original Orquestra repositories at https://github.com/zapatacomputing and https://github.com/zapata-engineering remain public but inactive — this is the most consequential "lost code" event of the period and is treated again in §9. **Multiverse Computing** does not maintain a public GitHub for its CompactifAI or Singularity products; tensor-network-based LLM compression is published only as Hugging Face weights. GitHub + 3↗

Academic groups and consortia

Academic QML code is structurally fragmented: most of it lives per-paper under personal accounts rather than group organisations. The exceptions worth knowing are **QuTech-Delft** at https://github.com/QuTech-Delft (with the OpenQL compiler and Quantum Inspire SDK), the **Perimeter Institute Quantum Intelligence Lab** at https://github.com/PIQuIL (especially **QuCumber** at https://github.com/PIQuIL/QuCumber for RBM-based state reconstruction), the **Aspuru-Guzik Matter Lab** at https://github.com/aspuru-guzik-group (and the spin-off **Tequila** at https://github.com/tequilahub/tequila), and **CERN's QTI** at https://github.com/CERN-IT-INNOVATION which maintains **QuASK** (https://github.com/CERN-IT-INNOVATION/QuASK) — a quantum-kernel-research framework published in *Quantum Machine Intelligence* in 2023 — and a noisy-gates simulator. **Hsin-Yuan Huang's** personal account hosts the seminal classical-shadows code at https://github.com/hsinyuan-huang/predicting-quantum-properties. **Sandia's pyGSTi** for gate-set tomography lives at https://github.com/sandialabs/pyGSTio. The DOE's NQI centres — **Q-NEXT**, **SQMS**, **QSC**, **C2QA**, **QSA** — publish code unevenly; **C2QA** at https://github.com/C2QA and the Eclipse-XACC framework at https://github.com/eclipse/xacc are the most usable entry points. GitHub + 2↗

The **Quantum Open Source Foundation** at https://github.com/qosf and the **Unitary Foundation** at https://github.com/unitaryfoundation are the two non-profit institutions whose GitHub footprints matter most for QML. QOSF maintains the Wittek MOOC archive (see §9), the monthly mentorship challenges and the canonical software list; Unitary Foundation maintains Mitiq, UCC and the Metriq benchmarking platform. Unitary↗

---

4. Specialised algorithmic repositories

The following catalogue traverses the algorithmic vocabulary of QML — VQA, kernels, GANs, RL, transformers, CNNs, NLP, federated, graph, encoding, autoencoders, Boltzmann machines, diffusion, tensor networks, error mitigation and benchmarking — and points to representative implementations.

Variational classifiers and VQA primitives. Beyond `qiskit-machine-learning` and PennyLane, **TorchQuantum** at https://github.com/mit-han-lab/torchquantum offers PyTorch-native parameterised circuits with batch GPU simulation; **Qiskit Algorithms** (https://github.com/qiskit-community/qiskit-algorithms) supplies primitive-based VQE/VQD/QAOA. **QAOA** specifically: **Entropica's OpenQAOA** (https://github.com/entropicalabs/openqaoa) is the most multi-backend implementation, integrating Mitiq for ZNE; **OpenQuantumComputing/QAOA** (https://github.com/OpenQuantumComputing/QAOA) supports custom mixers and CVaR; **qopt-best-practices** (https://github.com/qiskit-community/qopt-best-practices) is a SWAP-network reference. For **VQE**, **OpenFermion** plus **qiskit-nature** remain canonical, with **OpenVQE** (Atos/Sorbonne, accompanying arXiv:2206.08798) extending ADAPT-VQE on myQLM-Fermion. Entropicalabs + 5↗

Quantum kernels and SVMs. Beyond the framework defaults, **DmitriiNabok/qksvm** (https://github.com/DmitriiNabok/qksvm) generalises Qiskit's QKE/QKT; **Tim-Li/cuTN-QSVM** (https://github.com/Tim-Li/cuTN-QSVM) is the GPU/cuTensorNet-accelerated reference for large-scale benchmarking, published in MLST 2025; **thubregtsen's QHack 2021 entry** (https://github.com/thubregtsen/qhack) introduced trainable embedding kernels later proposed for PennyLane core. Theoretical foundation: **Schuld 2021** at https://arxiv.org/abs/2101.11020 ("Supervised quantum ML models are kernel methods"). GitHub + 4↗

Quantum GANs. The Zoufal–Lucchi–Woerner reference qGAN lives in https://github.com/qiskit-community/qiskit-machine-learning under tutorials/04_torch_qgan; **jundeli/quantum-gan** (https://github.com/jundeli/quantum-gan) implements QGAN-HG for drug discovery; **Sinestro38/qosf-qgan** demonstrates conditional QWGAN on stock-price sequences; the photonic counterparts are at https://github.com/Quandela/photonic-qgan and https://github.com/Quandela/photonic-qcbm. Qiskit + 2↗

Quantum reinforcement learning. qdevpsi3/qrl-dqn-gym (https://github.com/qdevpsi3/qrl-dqn-gym) is the faithful PennyLane reproduction of Skolik–Jerbi–Dunjko 2021; **qlan3/QuantumExplorer** implements variational soft actor–critic from arXiv:2112.11921; **javier-lazaro/SimplyQRL** (https://github.com/javier-lazaro/SimplyQRL) is a 2025 modular benchmarking library spanning PPO, DQN and the data-reuploading classifier. GitHub + 3↗

Quantum transformers and attention. The space is dominated by GSoC ML4SCI artefacts. **salcc/QuantumTransformers** (https://github.com/salcc/QuantumTransformers) accompanies Comajoan Cara et al., *Axioms* 13:323 (2024); **EyupBunlu/QViT_HEP_ML4Sci** (https://github.com/EyupBunlu/QViT_HEP_ML4Sci) accompanies Unlu et al., *Axioms* 13:187 (2024); the Quantum Orthogonal Attention paper is at https://arxiv.org/abs/2411.13520. There is no production-grade quantum transformer library as of 2026 — these are reproducible, not maintained. GitHub↗GitHub↗

Quantum CNNs. takh04/QCNN (https://github.com/takh04/QCNN) is the PennyLane benchmarking reference for Hur, Kim, Park's *Quantum Machine Intelligence* paper; **Jaybsoni/Quantum-Convolutional-Neural-Networks** reproduces Cong–Choi–Lukin for SPT phase recognition; the official **TFQ QCNN tutorial** sits inside https://github.com/tensorflow/quantum and the Qiskit reference at the qiskit-machine-learning tutorial page. Takh04 + 3↗

Quantum NLP. CQCL/lambeq (https://github.com/CQCL/lambeq) — Bobcat parser, DisCoCat/DisCoCirc, IQPAnsatz, PennyLane and PyTorch model integrations — is the only production toolkit. The underlying monoidal-category library **DisCoPy** is at https://github.com/discopy/discopy. The Lorenz et al. 2021 paper code is preserved at https://github.com/CQCL/qnlp_lorenz_etal_2021_resources; the QNLP+music demo at https://github.com/CQCL/Quanthoven illustrates the breadth. GitHub + 8↗

Quantum federated learning. Still a paper-driven space. **elucidator8918/QFL-MLNCP-NeurIPS** (https://github.com/elucidator8918/QFL-MLNCP-NeurIPS) is the NeurIPS 2024 MLNCP reference; **positivetechnologylab/Quorus** (https://github.com/positivetechnologylab/Quorus) tackles heterogeneous quantum clients; **ycchen1989/FederatedQLSTM** implements Chehimi et al. (arXiv:2312.14309); **Jindi0/QFL** explores noise-aware aggregation; **ShivaPokhrel/Deakin-Quantum-Federated-Learning** ties QFL to post-quantum cryptography. GitHub + 4↗

Quantum graph neural networks. sajen-k/Quantum_Graph_Recurrent_Neural_Network follows Verdon et al. (arXiv:1909.12264); **gerardPlanella/QGNN** integrates DMRG/PEPS with GNNs for many-body systems; **bossemel/QHack_Project** implements quantum spectral graph convolution; **smlab-niser/qgshap** is a 2025 quantum-acceleration of Shapley-GNN explanations accepted at AAAI 2026. GitHub + 3↗

Encoding and feature maps. Reference implementations live in PennyLane's `templates/embeddings` (AmplitudeEmbedding, AngleEmbedding, BasisEmbedding, IQPEmbedding, QAOAEmbedding) and Qiskit's `circuit.library` (ZFeatureMap, ZZFeatureMap, PauliFeatureMap). The 2025 hands-on tutorial **qml-tutorial** (https://qml-tutorial.github.io and arXiv:2502.01146) systematises these. Qml-tutorial↗GitHub↗

Autoencoders and Boltzmann machines. hsim13372/QCompress (https://github.com/hsim13372/QCompress) is the canonical Romero–Olson–Aspuru-Guzik implementation. For QBMs, **cameronperot/qbm** (https://github.com/cameronperot/qbm) follows Amin et al. 2018 on D-Wave; **mareksubocz/QRBM**, **haoyudoingthings/QBM** and **obliviateandsurrender/Quantum-Boltzmann-Machine** complete the picture. ADS + 5↗

Quantum diffusion and score-based generative models. This is the fastest-moving subarea. **FlorianFuerrutter/genQC** (https://github.com/FlorianFuerrutter/genQC) accompanies Fürrutter et al., *Nature Machine Intelligence* (2024); **NesyaLab/Quantum-Hybrid-Diffusion-Models** accompanies De Falco et al. 2024; **mashathepotato/GSoC-Quantum-Diffusion-Model** is the ML4SCI HEP application; **FujitsuResearch/chaotic-qdm** is the 2026 chaotic-quantum-diffusion reference; **boltzmannentropy/qonfusion** is a non-parametric quantum-circuit Gaussian RNG for stable diffusion. GitHub + 7↗

Tensor-network ML. jcmgray/quimb (https://github.com/jcmgray/quimb) is the most actively developed Python option; **ITensor** (https://github.com/ITensor) is the Julia/C++ standard; **TeNPy** (https://github.com/tenpy/tenpy) covers MPS/MPO physics; **TensorLy-Quantum** (https://github.com/tensorly/quantum) is the PyTorch-API tensor-ring framework collaboratively developed with NVIDIA; **TorchMPS** (https://github.com/jemisjoky/TorchMPS) succeeds **emstoudenmire/TNML** (https://github.com/emstoudenmire/TNML), the original Stoudenmire–Schwab NeurIPS 2016 reference. **Google's TensorNetwork** (https://github.com/google/TensorNetwork) is largely unmaintained since 2021 and

should be cited as alpha. **JoeyT1994/TensorNetworkQuantumSimulator.jl** is the 2024–2025 belief-propagation TN simulator on ITensorNetworks. GitHub + 9↗

Error mitigation. Mitiq dominates; the Qiskit-native addons at https://github.com/qiskit-community/qiskit-addon-zne and https://github.com/Qiskit/qiskit-addon-obs-approximate are complementary; **Algorithmiq's TEM** (https://github.com/Algorithmiq/TEM) is the tensor-network-based mitigation reference for arXiv:2307.11740. arxiv↗ arxiv↗

Benchmarking. XanaduAI/qml-benchmarks (https://github.com/XanaduAI/qml-benchmarks) — the Bowles–Ahmed–Schuld 2024 reference — is the most intellectually consequential entry. **MQT Bench** (https://github.com/munich-quantum-toolkit/bench), **SupermarQ** within https://github.com/Infleqtion/client-superstaq, **QED-C's** https://github.com/SRI-International/QC-App-Oriented-Benchmarks, **PNNL's QASMBench** (https://github.com/pnnl/QASMBench) and Unitary Foundation's **Metriq** at https://metriq.info round out the suite. GitHub↗

---

5. Foundational papers, books and courses

5.1 The seminal arc

The intellectual backbone of QML can be read across roughly twenty-five papers. The originating reviews are **Biamonte, Wittek, Pancotti, Rebentrost, Wiebe and Lloyd's "Quantum machine learning"** in *Nature* (2017), https://arxiv.org/abs/1611.09347, and **Schuld, Sinayskiy and Petruccione's** earlier *Contemporary Physics* introduction, https://arxiv.org/abs/1409.3097. The variational-era roadmaps are **Cerezo et al.'s "Variational quantum algorithms"** in *Nature Reviews Physics* (2021), https://arxiv.org/abs/2012.09265, and **Bharti et al.'s NISQ review** in *Reviews of Modern Physics* (2022), https://arxiv.org/abs/2101.08448. The candid corrective is **Cerezo, Verdon, Huang, Cincio and Coles, "Challenges and opportunities in quantum machine learning"** in *Nature Computational Science* (2022). PubMed + 8↗

The architectural foundations are: **Schuld and Killoran's** "Quantum machine learning in feature Hilbert spaces" (PRL 2019, https://arxiv.org/abs/1803.07128); **Havlíček et al.'s** *Nature* "Supervised learning with quantum-enhanced feature spaces" (https://arxiv.org/abs/1804.11326); **Mitarai, Negoro, Kitagawa and Fujii's** "Quantum circuit learning" (https://arxiv.org/abs/1803.00745); **Farhi and Neven's** "Classification with Quantum Neural Networks on Near Term Processors" (https://arxiv.org/abs/1802.06002); **Pérez-Salinas et al.'s** "Data re-uploading for a universal quantum classifier" in *Quantum* (https://arxiv.org/abs/1907.02085, code at https://github.com/AdrianPerezSalinas/universal_qlassifier); **Killoran et al.'s** "Continuous-variable quantum neural networks" (https://arxiv.org/abs/1806.06871, code at https://github.com/XanaduAI/quantum-neural-networks); **Romero, Olson and Aspuru-Guzik's** "Quantum autoencoders" (https://arxiv.org/abs/1612.02806); **Cong, Choi and Lukin's** "Quantum convolutional neural networks" in *Nature Physics* (https://arxiv.org/abs/1810.03787); **Grant et al.'s** "Hierarchical quantum classifiers" in *npj Quantum Information* (https://arxiv.org/abs/1804.03680); **Coecke, de Felice, Meichanetzidis and Toumi's** "Foundations for Near-Term Quantum NLP" (https://arxiv.org/abs/2012.03755), the lambeq paper. GitHub + 13↗

The trainability-and-expressivity literature crystallised around four results: **McClean et al.'s** "Barren plateaus in quantum neural network training landscapes" in *Nature Communications* (https://arxiv.org/abs/1803.11173, with PennyLane reproduction at https://pennylane.ai/qml/demos/tutorial_barren_plateaus); **Pesah et al.'s** "Absence of barren plateaus in quantum convolutional neural networks" in *PRX* (https://arxiv.org/abs/2011.02966); **Anschuetz and Kiani's** "Quantum variational algorithms are swamped with traps" in *Nature Communications* (https://arxiv.org/abs/2205.05786); **Abbas et al.'s** "The power of quantum neural networks" in *Nature Computational Science* (https://arxiv.org/abs/2011.00027). **Caro et al.'s** "Generalization in quantum machine learning from few training data" in *Nature Communications* (https://arxiv.org/abs/2111.05292) and **Du, Hsieh, Liu and Tao's** "Expressive power of parametrized quantum circuits" in *Physical Review Research* (https://arxiv.org/abs/1810.11922) close the loop. GitHub + 10↗

The advantage and dequantisation thread runs through **Liu, Arunachalam and Temme's** "rigorous and robust quantum speed-up in supervised machine learning" in *Nature Physics* (https://arxiv.org/abs/2010.02174); **Huang, Broughton, Mohseni et al.'s** "Power of data in quantum machine learning" in *Nature Communications* (https://arxiv.org/abs/2011.01938); **Huang et al.'s** "Quantum advantage in learning from experiments" in *Science* (https://arxiv.org/abs/2112.00778, with PennyLane demo at

https://pennylane.ai/qml/demos/tutorial_learning_from_experiments); **Huang, Kueng, Torlai, Albert and Preskill's** "Provably efficient machine learning for quantum many-body problems" in *Science* (https://arxiv.org/abs/2106.12627). The recent corrective is **Schuld and Killoran's** "Is quantum advantage the right goal for quantum machine learning?" in *PRX Quantum* (https://arxiv.org/abs/2203.01340). The dequantisation literature flowing from Tang's work continues through https://www.nature.com/articles/s42254-022-00511-w, https://arxiv.org/abs/2112.00811 and https://arxiv.org/abs/2304.04932; the May 2025 result by Sweke et al. on dequantisation via random Fourier features is at https://arxiv.org/abs/2505.15902. Mindat + 7↗

The 2024–2025 frontier additions include **Wang and Liu's** *Reports on Progress in Physics* review "From NISQ to Fault Tolerance" (https://arxiv.org/abs/2401.11351); **Liu et al.'s** "Towards provably efficient quantum algorithms for large-scale machine-learning models" in *Nature Communications* (https://arxiv.org/abs/2303.03428); **Cerezo et al.'s** *Nature Communications* "Does provable absence of barren plateaus imply classical simulability?" (2025); **Gil-Fuster, Eisert and Bravo-Prieto's** "Understanding QML also requires rethinking generalization" (2024); and **Du et al.'s** book-length tutorial "Quantum Machine Learning: A Hands-on Tutorial for Machine Learning Practitioners and Researchers" at https://arxiv.org/abs/2502.01146 with the live site at https://qml-tutorial.github.io. ADS + 6↗

5.2 Books

The two essential book-length QML treatments are **Schuld and Petruccione's** *Supervised Learning with Quantum Computers* (Springer, 2018, https://link.springer.com/book/10.1007/978-3-319-96424-9) and its substantially extended successor *Machine Learning with Quantum Computers* (Springer, 2021, https://link.springer.com/book/10.1007/978-3-030-83098-4). **Peter Wittek's** *Quantum Machine Learning: What Quantum Computing Means to Data Mining* (Academic Press, 2014, https://shop.elsevier.com/books/quantum-machine-learning/wittek/978-0-12-800953-6, archive copy at https://archive.org/details/quantum-machine-learning-what-quantum-computing-means-to-data-mining-wittek-2014-08-28) was the first textbook in the field and now reads as both foundational and elegiac. **Combarro and González-Castillo's** *A Practical Guide to Quantum Machine Learning and Quantum Optimization* (Packt, 2023, https://www.packtpub.com/en-us/product/a-practical-guide-to-quantum-machine-learning-and-quantum-optimization-9781804613832, code at https://github.com/PacktPublishing/A-Practical-Guide-to-Quantum-Machine-Learning-and-Quantum-Optimization) is the most useful contemporary code-first guide. **Sutor's** *Dancing with Qubits* (2nd ed., Packt, 2024, code at https://github.com/PacktPublishing/Dancing-with-Qubits-2E) added explicit NISQ and QML chapters; **Hidary's** *Quantum Computing: An Applied Approach* (2nd ed., Springer, 2021, code at https://github.com/jackhidary/quantumcomputingbook) added a QML chapter. **Nielsen and Chuang's** *Quantum Computation and Quantum Information* remains the canonical underlay. Springer + 8↗

5.3 Courses and lectures

The single most cohesive online curriculum is the **PennyLane Codebook** at https://pennylane.ai/codebook, accompanied by the QML demos hub at https://pennylane.ai/qml. **Maria Schuld's** lecture catalogue lives across the PennyLane YouTube channel at https://www.youtube.com/@PennyLaneAI and her researcher page at https://pennylane.ai/profile/mariaschuld. The **IBM Qiskit Global Summer School** has run annually since 2020, with the most recent editions preserved at https://github.com/qiskit-community/qgss-2024 and https://github.com/qiskit-community/qgss-2025 (lecture notes at https://github.com/qiskit-community/qgss-2025-lecture-notes); **QGSS 2021** was the QML-focused edition. The **Qiskit textbook**, now relocated to https://github.com/Qiskit/textbook with the live course at https://learning.quantum.ibm.com, replaces the archived https://github.com/qiskit-community/qiskit-textbook. GitHub + 2↗

University courses worth pinning: **Caltech Ph/CS 219 (Preskill)** at http://theory.caltech.edu/~preskill/ph219/ with 2024 edition at https://www.preskill.caltech.edu/ph219/ph219_2024.html and Fall 2020 lectures on YouTube at https://www.youtube.com/playlist?list=PL0ojjrEqIyPy-1RRD8cTD_lF1hflo89Iu; **MIT 8.370/8.371** on OCW at https://ocw.mit.edu/courses/8-370x-quantum-information-science-i-spring-2018/ and https://ocw.mit.edu/courses/8-371x-quantum-information-science-ii-spring-2018/; the Open Learning Library run at https://openlearninglibrary.mit.edu (UTQML101x and 8.37*x); **MIT 8.372 (QIS3)** at https://mit-qis3.gitlab.io; **Ryan O'Donnell's** CMU 15-859BB notes and lectures at https://www.cs.cmu.edu/~odonnell/quantum18/ and https://www.youtube.com/playlist?list=PLm3J0oaFux3YL5qLskC6xQ24JpMwOAeJz. The **Quantum Native Dojo** at https://github.com/qulacs/quantum-native-dojo-en (web at https://dojo.qulacs.org/en/) is the best Japanese-origin self-study path. **QHack** archives are at https://github.com/XanaduAI/QHack2022, https://github.com/XanaduAI/QHack2023 and

https://github.com/XanaduAI/QHack2024-coding-challenges; the **QOSF Mentorship Program** at https://qosf.org/qc_mentorship/ is the most reliable pathway for guided independent research. GitHub +3 + 4↗

The most consequential lost course in the field — **Wittek's edX UTQML101x** — survives only through community preservation. That recovery story is documented in §9. GitHub↗

---

6. Datasets, benchmarks and physical-science training corpora

QML's data layer differs from classical ML in that the same dataset is often used in two registers — once as a classical input to be encoded into a quantum state, and once as the *output* of a quantum experiment to be classified. The key resources span both.

The framework-native catalogues are **PennyLane Datasets** at https://pennylane.ai/datasets (with the QML benchmarks collection at https://pennylane.ai/datasets/collection/qml-benchmarks and the Low-Depth Image Circuits collection encoding MNIST, Fashion-MNIST, CIFAR-10 and Imagenette as quantum circuits at https://pennylane.ai/datasets/collection/low-depth-image-circuits, accompanying arXiv:2505.03399), the **TensorFlow Quantum tutorials** at https://www.tensorflow.org/quantum/tutorials (canonical 3-vs-6 MNIST and Fashion-MNIST PQK reproductions of Huang et al.) and the **qiskit-machine-learning datasets** API at https://qiskit-community.github.io/qiskit-machine-learning/apidocs/qiskit_machine_learning.datasets.html. arxiv↗Qiskit↗

The chemistry corpora — central to any QML-for-chemistry pipeline — are aggregated at **quantum-machine.org's dataset hub** at http://quantum-machine.org/datasets/, hosting QM7, QM7b, QM7-X, QM8, QM9, MD17, rMD17, MD22, ANI-1, ANI-1x and ANI-1ccx. **PubChemQC** (RIKEN) is at http://pubchemqc.riken.jp/. The **Materials Project** at https://next-gen.materialsproject.org/ holds DFT-computed properties for 200,000+ inorganic materials. **OpenFermion's** `MolecularData` HDF5 files ship inside https://github.com/quantumlib/OpenFermion. ACS Publications + 2↗

The Hamiltonian and circuit benchmark suites are: **HamLib** at https://portal.nersc.gov/cfs/m888/dcamps/hamlib/ (Sawaya et al., *Quantum* 8, 1559 (2024), arXiv:2306.13126) — a library of 2-to-1000-qubit Hamiltonians spanning Heisenberg, Hubbard, electronic and combinatorial-optimisation problems with standardised encodings; **MQT Bench** at https://www.cda.cit.tum.de/mqtbench/ and https://github.com/munich-quantum-toolkit/bench; **SupermarQ** at https://github.com/SupertechLabs/SupermarQ (HPCA 2022 best paper); **QASMBench** at https://github.com/pnnl/QASMBench (Li et al., *ACM TQC* 2022); **QED-C's QC-App-Oriented-Benchmarks** at https://github.com/SRI-International/QC-App-Oriented-Benchmarks; **Metriq** at https://metriq.info. Two specifically-QML benchmark assets matter: **MNISQ** at https://github.com/FujiiLabCollaboration/MNISQ-quantum-circuit-dataset (arXiv:2306.16627; large-scale QASM circuits encoding MNIST/Fashion-MNIST/Kuzushiji as quantum states) and **QDataSet** at https://github.com/eperrier/QDataSet (52 datasets from 1- and 2-qubit simulations covering control, tomography and spectroscopy, *Scientific Data* 2022). CoLab + 7↗

---

7. Conferences, journals and communities

The two journals to read first are **Quantum** (https://quantum-journal.org), the open-access community-run arXiv-overlay venue founded in 2017 that publishes most of the rigorous algorithmic work, and **PRX Quantum** (https://journals.aps.org/prxquantum/), the highly-selective APS open-access counterpart. **npj Quantum Information** (https://www.nature.com/npjqi/) and **Quantum Science and Technology** (https://iopscience.iop.org/journal/2058-9565) are the next tier, with **Quantum Machine Intelligence** (https://link.springer.com/journal/42484) the dedicated quantum-AI journal — it hosts the QTML special issues. **IEEE Transactions on Quantum Engineering** (https://ieee-tqe.org), **Communications Physics**, **Reviews of Modern Physics** (for the Cerezo/Bharti review tier) and **Nature Reviews Physics** complete the venue map. Quantum + 4↗

The conferences fall into three clusters. The theoretical-physics cluster — **QIP** at https://qipconference.org (QIP 2026 in Riga; ~1000 attendees; no proceedings), **TQC** at https://tqc-conference.org with proceedings in LIPIcs at https://drops.dagstuhl.de/entities/volume/LIPIcs-volume-350, the **APS March Meeting** (DQI division) at https://march.aps.org, **AQIS** at http://aqis-conf.org and **QCMC** at http://qcmc.org. The applied-engineering cluster — **IEEE**

Quantum Week (QCE) at https://qce.quantum.ieee.org/2026/, **Q2B** at https://q2b.qcware.com (Q2B26 co-hosting the Chicago Quantum Summit). The QML-specific cluster — **QTML** (the field's flagship; QTML 2025 in Singapore at https://qtml2025.cqt.sg, QTML 2026 in Stellenbosch); **QHack** at https://qhack.ai; the **NeurIPS Quantum Tensor Networks for ML** workshops at https://tensorworkshop.github.io/NeurIPS2020/ and the **NeurIPS Machine Learning and the Physical Sciences** workshop series at https://ml4physicalsciences.github.io. Qureca + 5↗

The community institutions matter as much as the publications. **Unitary Foundation** at https://unitary.foundation runs the microgrant program (more than 100 awards), Mitiq, the Metriq benchmarks platform and unitaryHACK; **QOSF** at https://qosf.org runs the Quantum Computing Mentorship Program (cohort 11 in 2025), monthly challenges and the canonical software list; **Womanium** at https://womanium.org/Quantum/Program runs the Global Quantum Program with 4,000+ alumni and paid fellowships; **QWorld** at https://qworld.net runs the QBronze/QSilver/QNickel workshops and the QCousins regional network (QPoland, QTurkey, QPakistan, QGhana and ~25 others); the **QED-C** at https://quantumconsortium.org, the **NSF Quantum Leap Challenge Institutes** at https://www.nsf.gov/funding/opportunities/qlci-quantum-leap-challenge-institutes and the **National Quantum Initiative** umbrella at https://www.quantum.gov coordinate the institutional infrastructure in the United States. The day-to-day discussion lives on **Stack Exchange Quantum Computing** at https://quantumcomputing.stackexchange.com, the **PennyLane Discussion Forum** at https://discuss.pennylane.ai, the **Qiskit Slack** at https://qisk.it/join-slack, the Xanadu Slack at https://u.strawberryfields.ai/slack, **r/QuantumComputing** at https://www.reddit.com/r/QuantumComputing/, and the editorial outlets **Quantum Computing Report** at https://quantumcomputingreport.com, **The Quantum Insider** at https://thequantuminsider.com, **Quantum Zeitgeist** at https://quantumzeitgeist.com and **Scott Aaronson's Shtetl-Optimized** at https://scottaaronson.blog. Unitary + 4↗

---

8. The 2024–2026 frontier

Five intellectual currents define the QML frontier as this compendium goes to press, and each has begun to acquire a defensible evidence base rather than a speculative one.

Quantum NLP and proto-foundation models. The most mature thread. **lambeq** (https://github.com/CQCL/lambeq) is the only production-grade QNLP library and its experimental DisCoCirc module shipping under `lambeq.experimental.discocirc` is the first whole-paragraph compositional encoder targeting trapped-ion hardware. The transformer side is a constellation of research artefacts — **HQViT** (https://arxiv.org/abs/2504.02730), the quantum-walk-based **GQWformer** (https://arxiv.org/abs/2412.02285), the molecular-generation hybrid transformer (https://arxiv.org/abs/2502.19214), the AAAI 2026-candidate vectorised attention paper (https://arxiv.org/abs/2508.18464) — but no group has yet released what one could honestly call a "quantum LLM" at GPT scale, and that label should be used with care. GitHub↗Quantinuum↗

Hybrid programming at scale. NVIDIA CUDA-Q (https://github.com/NVIDIA/cuda-quantum) and the CUDA-QX libraries (https://github.com/NVIDIA/cudaqx) have decisively become the GPU+QPU substrate; **PennyLane Catalyst** (https://github.com/PennyLaneAI/catalyst, JOSS 2024) supplies the MLIR-based JIT layer; **Lightning Kokkos** and **Lightning GPU MPI** handle distributed simulation. The PennyLane Kokkos blog at https://pennylane.ai/blog/2023/04/pennylane-goes-kokkos-a-novel-hardware-agnostic-parallel-backend-for-quantum-simulations/ and the HPC paper at https://arxiv.org/abs/2403.02512 are the best pointers. **Yao.jl** with **CuYao.jl** is the Julia counterpart; **TensorCircuit** the JAX/TF/PyTorch tensor-network counterpart. GitHub + 4↗

Fault-tolerant QML on early FTQC. The hardware demonstrations that matter for QML's medium-term future are: **QuEra–Harvard–MIT's 48 logical qubits** in *Nature* (https://www.nature.com/articles/s41586-023-06927-3, December 2023); **Quantinuum + Microsoft's 4 logical qubits with 800× error suppression** (April 2024) and the **12-logical-qubit end-to-end chemistry demonstration** (https://azure.microsoft.com/en-us/blog/quantum/2024/09/10/microsoft-and-quantinuum-create-12-logical-qubits-and-demonstrate-a-hybrid-end-to-end-chemistry-simulation/, September 2024); **Quantinuum H2's universal fault-tolerant gate set with magic-state code-switching** (June 2025) reaching a sub-physical logical non-Clifford error rate of 5.1×10^{-4}, reported at https://thequantuminsider.com/2025/06/27/quantinuum-crosses-key-quantum-error-correction-threshold-marks-turn-from-nisq-to-utility-scale/; and the November 2025 below-threshold *Nature* demonstration. The most accessible FTQC-research stack is **NVIDIA cudaqx** plus **Google's Qualtran** for resource estimation. None of these constitutes industrial-scale fault-tolerant QML; all of them shift the timeline. Wikipedia + 5↗

The advantage debate, sharpened. The 2024 **Bowles–Ahmed–Schuld benchmark** (arXiv:2403.07059, code at https://github.com/XanaduAI/qml-benchmarks, blog at https://pennylane.ai/blog/2024/03/benchmarking-near-term-quantum-algorithms-is-an-art) is the most important methodological paper of the period: across twelve QML models, six binary classification tasks and 160 datasets, *out-of-the-box classical models generally win, and removing entanglement frequently leaves performance unchanged*. Combined with Schuld and Killoran's PRX Quantum essay (https://arxiv.org/abs/2203.01340) and the dequantisation literature culminating in Sweke et al. 2025 (https://arxiv.org/abs/2505.15902), the empirical picture for *near-term* QML advantage in classical-data tasks is sober: claimed advantages survive scrutiny only in narrow regimes. The contrary signal comes from learning-from-experiments tasks — Huang et al.'s *Science* result (https://arxiv.org/abs/2112.00778) and the IonQ Aria entanglement-induced advantages experiment (https://arxiv.org/abs/2410.03094) — and from the rigorous polynomial-separation results in **arXiv → Quantum 2026** at https://quantum-journal.org/papers/q-2026-01-20-1976/. The **Recio-Armengol–Ahmed–Bowles "train classically, deploy quantum" 1000-qubit Born machine** (https://arxiv.org/abs/2503.02934) suggests the most credible near-term route is to use quantum hardware as a sampler for a classically-trained model. arXiv + 4↗

Tensor networks and quantum diffusion models. Tensor networks are now the principal classical bridge to QML: **quimb** (https://github.com/jcmgray/quimb), **ITensor** (https://github.com/ITensor), **TeNPy** (https://github.com/tenpy/tenpy) and **TensorLy-Quantum** (https://github.com/tensorly/quantum) are the working stack, with the **TensorNetworkQuantumSimulator.jl** belief-propagation simulator (https://github.com/JoeyT1994/TensorNetworkQuantumSimulator.jl) the most interesting 2024–2025 entrant. Quantum diffusion models — Fürrutter et al. in *Nature Machine Intelligence* (2024, code at https://github.com/FlorianFuerrutter/genQC), the De Falco et al. hybrid-diffusion line (https://github.com/NesyaLab/Quantum-Hybrid-Diffusion-Models), the Fujitsu chaotic-quantum-diffusion 2026 paper at https://arxiv.org/abs/2602.22061 with code at https://github.com/FujitsuResearch/chaotic-qdm — have produced the first cases where quantum-circuit synthesis is itself learned by a diffusion model. Readthedocs + 4↗

---

9. Recovering the field's lost code

A research field's intellectual integrity depends on the survival of its early code. Several QML repositories of historical importance are now at risk of disappearance, archival rot or institutional disinvestment. This section is a recovery layer.

Peter Wittek's legacy. Wittek — who coined "quantum machine learning" in his 2014 textbook, taught the first edX QML course (UTQML101x), and directed CDL's Quantum Stream — died in a Trishul avalanche on 29 September 2019. His GitHub profile at https://github.com/peterwittek?tab=repositories has been frozen since; it contains 21 repositories including https://github.com/peterwittek/quantum_machine_learning_figures (figure code for the 2014 book), https://github.com/peterwittek/qml-rg (the ICFO QML reading-group archive), the **ncpol2sdpa** noncommutative-polynomial-optimisation library at https://github.com/peterwittek/ncpol2sdpa still cited in current SDP literature, and the **somoclu** parallel SOM library at https://github.com/peterwittek/somoclu. The MOOC lectures live on YouTube at https://www.youtube.com/playlist?list=PLmRxgFnCIhaMgvot-Xuym_hn69lmzIokg. The MOOC notebooks themselves — in both Qiskit and Forest variants, with D-Wave Ocean exercises — are preserved by QOSF at https://github.com/qosf/qml-mooc and remain the most actively maintained surviving copy. The 2014 textbook is mirrored at https://archive.org/details/quantum-machine-learning-what-quantum-computing-means-to-data-mining-wittek-2014-08-28. The recommended preservation move is to mirror all 21 `peterwittek/*` repositories and the QOSF fork via Software Heritage at https://archive.softwareheritage.org/. ScienceDirect + 14↗

Rigetti Grove. The original public reference for VQE, QAOA, quantum phase estimation, swap-test and Bernstein–Vazirani in pyQuil — heavily cited in 2017–2019 QML literature — has been archived since 2019 at https://github.com/rigetti/grove (last release v2.0.0b0). The old documentation is at http://grove-docs.readthedocs.io/en/latest/. Software Heritage's snapshot at https://archive.softwareheritage.org/browse/origin/?origin_url=https://github.com/rigetti/grove is the recommended canonical mirror. **pyQuil's** historical docs at https://pyquil-docs.rigetti.com/en/v2.0.0/ and https://pyquil-docs.rigetti.com/en/v2.24.0/ contain migration material valuable for reproducing 2018–2020 papers. GitHub + 2↗

Zapata Computing's repositories. Zapata Computing announced cessation of operations on 7–9 October 2024 (8-K filing reported at https://www.hpcwire.com/2024/10/14/zapata-computing-early-quantum-ai-software-specialist-ceases-operations/) and re-emerged as Zapata Quantum Inc. on 23 April 2026 with $15M Triatomic-led financing

(https://thequantuminsider.com/2025/09/03/viva-zapata-zapata-quantum-reemerges-after-bankruptcy-filing-with-restructuring-and-new-financing/). The original repositories at https://github.com/zapatacomputing — including z-quantum-core, z-quantum-qaoa, z-quantum-vqe, z-quantum-qcbm, z-quantum-optimizers, tutorial-0-welcome, TN-GEO-car-plant-optimization and darpa-circuits — remain public but unmaintained since 2023; the meta-package https://github.com/zapata-engineering/orquestra-core is the surviving entry point to Orquestra. The **z-quantum-qcbm** repository in particular preserves an early reference Quantum Circuit Born Machine implementation that should be archived externally. Investing.com + 7↗

Microsoft LIQUi|⟩. The F# predecessor to Q# is preserved read-only at https://github.com/StationQ/Liquid (formerly msr-quarc/Liquid), with the project page at https://www.microsoft.com/en-us/research/project/language-integrated-quantum-operations-liqui/ and the static site at https://stationq.github.io/Liquid/. The msr-quarc archive at https://github.com/msr-quarc holds the ReVerC reversible-computing tools and the qsharp.sty LaTeX style. **Microsoft Quantum Katas** at https://github.com/microsoft/QuantumKatas was archived in August 2024 with content migrated into https://github.com/microsoft/qsharp under `katas/`. GitHub + 2↗

ProjectQ. ETH Zürich's hardware-agnostic compiler at https://github.com/ProjectQ-Framework/ProjectQ is alive but slow-moving; the ML-relevant **FermiLib** at https://github.com/ProjectQ-Framework/FermiLib has not been updated since 2018. The 0.5-petabyte 45-qubit simulation paper and high-performance simulator remain useful baselines. GitHub + 3↗

qiskit-aqua. The April-2021-deprecated original IBM application module at https://github.com/qiskit-community/qiskit-aqua contains the first reference qGAN, QSVM and VQC implementations under their pre-Terra `qiskit.aqua.algorithms` API. The migration guide notebook is at https://github.com/qiskit-community/qiskit-aqua/blob/main/docs/tutorials/Qiskit%20Algorithms%20Migration%20Guide.ipynb. Pre-2021 tutorials referencing `qiskit.aqua.*` frequently break against the live docs site; the appropriate Wayback queries are https://web.archive.org/web/*/qiskit.org/documentation/aqua_tutorials/* and https://web.archive.org/web/*/qiskit.org/documentation/stable/0.30/stubs/qiskit.aqua.*. GitHub↗GitHub↗

Strawberry Fields photonic-QML demos. The repository at https://github.com/XanaduAI/strawberryfields accepts only bug fixes; the photonics gallery at https://strawberryfields.ai/photonics/index.html and the Borealis tutorial at https://strawberryfields.ai/photonics/demos/run_tutorial_borealis.html should be snapshotted for the historical record of GBS-based kernel methods and continuous-variable QNNs.

Stoudenmire's TNML. The Stoudenmire–Schwab 2016 NeurIPS reference at https://github.com/emstoudenmire/TNML was succeeded by **TorchMPS** (https://github.com/jemisjoky/TorchMPS) and lives on as one of the foundational tensor-network ML references; both should be preserved as a pair. arXiv↗

Other deposit channels. Zenodo at https://zenodo.org/search?q=%22quantum+machine+learning%22 and Figshare at https://figshare.com/search?q=quantum+machine+learning host paper-companion data and code with DOIs — including snapshots of `XanaduAI/qml-benchmarks` and several `qiskit-community/*` releases, plus the *Nature Communications* data files for Huang et al.'s "Power of data" paper. **Software Heritage** at https://archive.softwareheritage.org is the recommended permanent archive substrate for any of the at-risk repositories above.

---

10. Practical setup and starter kits

For someone new to the field, the fastest route to a working QML environment is one of three paths. The **PennyLane demos** at https://pennylane.ai/qml/demonstrations/ run as-is on Google Colab via the "Open in Colab" buttons. The **Qiskit tutorials** at https://learning.quantum.ibm.com pair with cloud execution on the free IBM Quantum Open plan at https://quantum.ibm.com (which now includes 100+-qubit Heron and Eagle backends). **NVIDIA CUDA-Q Academic** at https://github.com/NVIDIA/cuda-q-academic ships Colab-friendly Jupyter modules including a `01_quick_start_to_quantum.ipynb` and the divide-and-conquer QAOA, ADAPT-QAOA and QAOA-GPT notebooks. The **QOSF qml-mooc** at https://github.com/qosf/qml-mooc still works on Colab with `pip install qiskit pyquil dwave-ocean-sdk`. GitHub + 3↗

For containerised deployment: **PennyLane** publishes images at https://hub.docker.com/u/pennylaneai with backend-specific Lightning containers; **Qiskit** at https://hub.docker.com/r/qiskit/qiskit; **CUDA-Q** through the NVIDIA NGC catalogue at https://catalog.ngc.nvidia.com/orgs/nvidia/teams/cuda-quantum/containers/cuda-quantum. The PennyLane Lightning Dockerfile lives at https://github.com/PennyLaneAI/pennylane-lightning/tree/master/docker; the CUDA-Q devcontainer template at https://github.com/NVIDIA/cuda-quantum/tree/main/.devcontainer.

For cloud sandboxes: **IBM Quantum Platform** at https://quantum.ibm.com (Composer + Lab Jupyter, free Open plan); **Xanadu Cloud** at https://cloud.xanadu.ai (X-series photonic and Borealis); **AWS Braket** at https://aws.amazon.com/braket/ (Rigetti, IonQ, IQM, QuEra Aquila, simulators); **Azure Quantum** at https://learn.microsoft.com/en-us/azure/quantum/ with VS Code at https://vscode.dev/quantum (IonQ, Quantinuum, Pasqal, Rigetti); **qBraid Lab** at https://qbraid.com (multi-vendor Jupyter); **Strangeworks** at https://strangeworks.com. For browser-based exploration: **IBM Quantum Composer** at https://quantum.ibm.com/composer, **Quirk** at https://algassert.com/quirk, **Quantum Inspire** at https://www.quantum-inspire.com, and **Quantastica's Quantum Programming Studio** at https://quantum-circuit.com.

The next five years

Three forecasts emerge naturally from the resources catalogued above. First, the centre of mass of credible QML claims is migrating from *classical-data classification* — where Bowles, Ahmed and Schuld have made it expensive to claim quantum advantage without rigorous ablations — toward *quantum-data tasks* (Hamiltonian learning, quantum-state classification, classical shadows pipelines, learning-from-experiments) where polynomial separations are now provable. Expect QTML, QIP and *Quantum* to publish the most consequential work; expect the *PRX Quantum* and *Nature Communications* corrective tradition to continue pruning the literature.

Second, the substrate beneath QML is consolidating around three layers: a *device* layer of trapped-ion, neutral-atom, photonic and superconducting hardware now crossing the logical-qubit threshold; a *compiler* layer dominated by MLIR-based JITs (PennyLane Catalyst, CUDA-Q, tket2, Guppy/HUGR); and an *application* layer of differentiable, autograd-aware Python and Julia frameworks. The "framework war" of 2018–2021 has effectively ended; PennyLane, Qiskit Machine Learning, Cirq+TFQ, lambeq and Yao.jl now coexist with clear functional niches. The interesting competition has moved to compilers and to GPU-accelerated tensor-network simulation, where Lightning-Kokkos, cuStateVec/cuTensorNet and quimb are racing.

Third, the most important question for the next five years is not whether QML achieves "quantum advantage" but how the field rebuilds its evaluation infrastructure to detect such advantage honestly when it appears. The XanaduAI/qml-benchmarks suite, MQT Bench, SupermarQ, HamLib, QED-C benchmarks and Metriq are the early architecture of that evaluation layer. Combined with a healthy dequantisation tradition that keeps closing speculative gaps, the field is acquiring the methodological maturity that classical ML benchmarks took two decades to develop — and is doing so with far less self-deception. That is, in the end, the real news from 2024–2026: quantum machine learning has stopped being primarily a discipline of promises, and has become primarily a discipline of evidence.

The repositories collected here are the substrate of that transition. They will outlive the news cycles and the company restructurings; if archived properly, they will outlive their original maintainers. That is the function of this list: not to capture a moment, but to make sure the moment can still be read in twenty years.