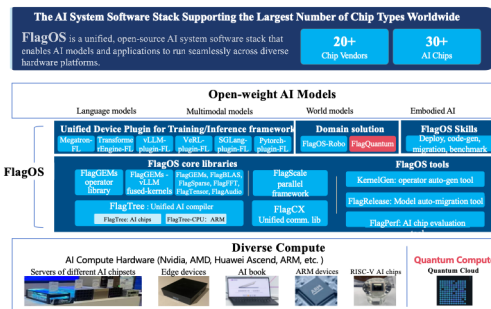


Wei LIU · Beijing Academy of Artificial Intelligence (BAAI)

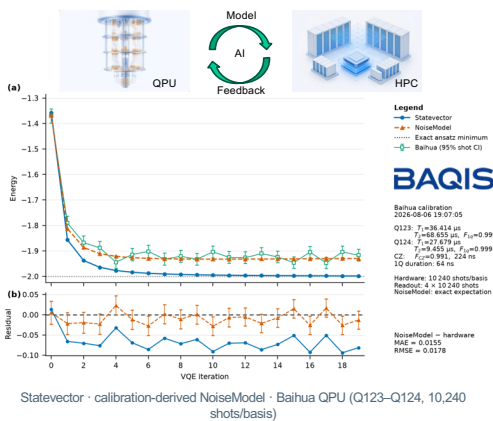
Quantum compute in FlagOS

FlagOS: Open-sourced System SW for Multi-Vendor AI chipsets
URL: flagos.io, GitHub: github.com/flagos-ai



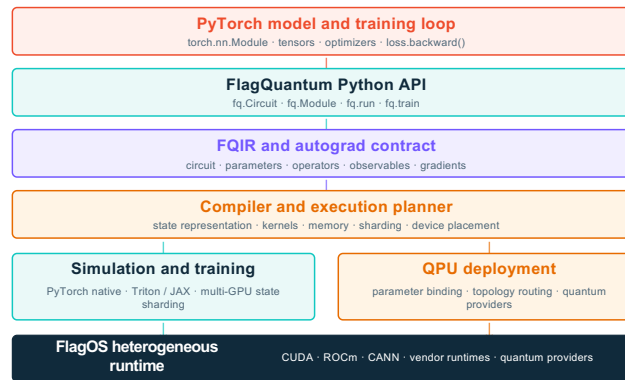
FlagQuantum is highlighted in the domain-solution layer, linking PyTorch-native quantum workloads with FlagOS libraries, tools, and heterogeneous compute.

Example: QPU digital twin



Open source: github.com/flagos-ai/FlagQuantum

FlagQuantum architecture: PyTorch to QPU

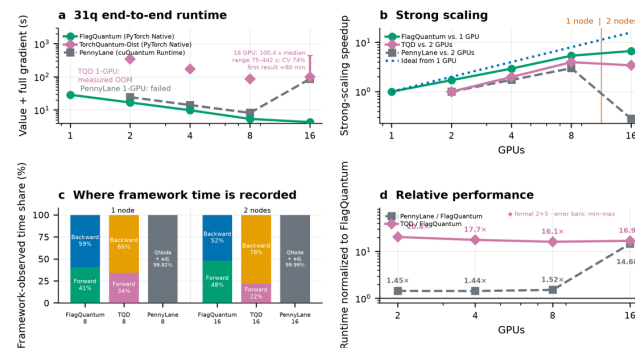


The PyTorch program and autograd interface stay fixed while FlagQuantum selects the state representation and execution target.

31 qubits differentiable statevector

16 GPUs · cross-node autograd · end-to-end value + gradient

Matched differentiable full-width linear HEA
31 qubits · 8 layers · 248 trainable RY parameters · 488 gates
Directed nearest-neighbor CNOT chain · $\langle Z_{15} \rangle$ · complex64



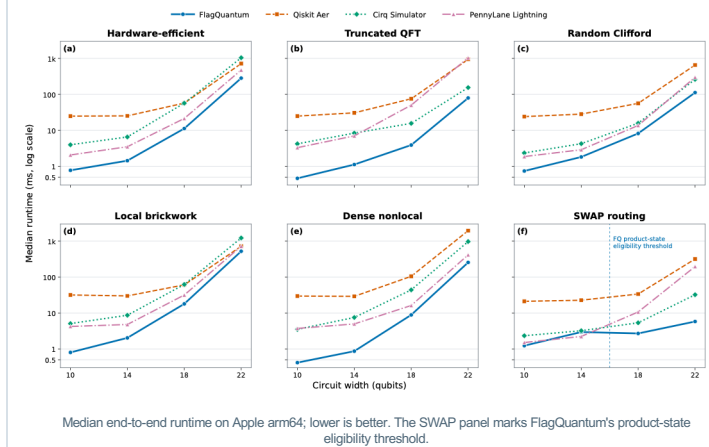
Scaling and performance evidence

Distributed MPS capacity 131,072 MPS sites on 16x A800



Exact-statevector CPU runtime across quantum simulation frameworks

Apple arm64 | 1 CPU thread | complex128 | median end-to-end runtime | lower is better



Median end-to-end runtime on Apple arm64; lower is better. The SWAP panel marks FlagQuantum's product-state eligibility threshold.

Evidence shown: lower is better where runtime is plotted

FlagQuantum Execution & Research Continuum:
Local Simulation → Distributed Training → Digital Twins → QEC Memory → Fault-Tolerant Quantum Computing