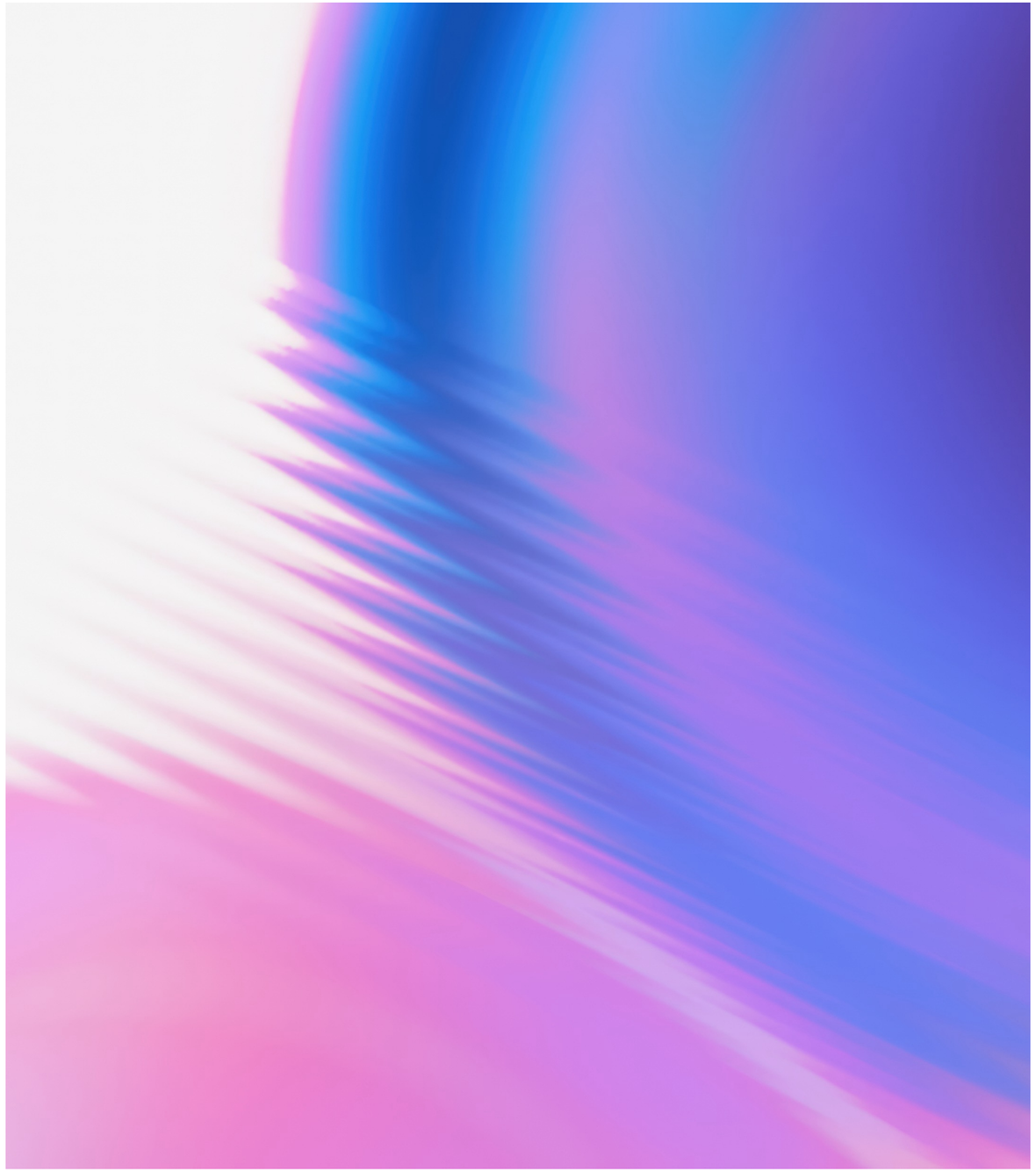
AI for Qiskit. Hackathon starter pack

BasQ Qiskit Fall Fest 2025
5 Nov 2025

Juan Cruz-Benito
AI for Quantum Product Owner
IBM Quantum



Outline

- Qiskit Code Assistant
- Qiskit-ibm-transpiler with the AI-powered transpiler passes

Qiskit Code Assistant

The Qiskit Code Assistant is a generative AI-powered coding companion built on watsonx, designed to make quantum computing more accessible and enhance the developer experience.

- Trained on Qiskit 2.x and earlier to assist with your coding tasks.
- Available as extensions for VS Code and JupyterLab.
- First introduced in 2023 as a preview; now:
  - Cloud version for IBM Quantum premium plan users.
  - Local mode for everyone via open LLMs.

More info https://quantum.cloud.ibm.com/docs/en/guides/qiskit-code-assistant
(👈 QR code)

Qiskit Code Assistant VSCode

Bring Qiskit Code Assistant into Visual Studio Code environment (VSCode):

- Use it via the cloud (recommended) or run locally.
- Perfect for coding, debugging, and exploring Qiskit.

- Run code completions using the keys Ctrl + .

Detailed setup instructions https://quantum.cloud.ibm.com/docs/en/guides/qiskit-code-assistant-vscode
(👈 QR code)

Qiskit Code Assistant - JupyterLab

Bring Qiskit Code Assistant into JupyterLab environment:

- Use it via the cloud (recommended) or run locally.
- Perfect for coding, debugging, and exploring Qiskit.
- Run code completions using the keys Alt + .

```
#Install the extension

pip install qiskit-code-assistant-jupyterlab

# Run jupyterLab

jupyter lab
```

Detailed setup instructions
https://quantum.cloud.ibm.com/docs/en/guides/qiskit-code-assistant-jupyterlab
(👈 QR code)

Qiskit Code Assistant – Local mode

Run Locally

Prefer using your own hardware instead of the cloud?

- Follow the setup guide here: https://quantum.cloud.ibm.com/docs/en/guides/qiskit-code-assistant-local (👈 QR code)

Our LLMs for Qiskit Coding are available on Hugging Face https://huggingface.co/Qiskit/

Qiskit Code Assistant – Local mode

1 command automatic setup to run the Qiskit Code Assistant in local mode:

- VSCode

```
bash <(curl -fsSL https://raw.githubusercontent.com/Qiskit/qiskit-code-assistant-vscode/main/setup_local.sh)
```

- JupyterLab

```
bash <(curl -fsSL https://raw.githubusercontent.com/Qiskit/qiskit-code-assistant-jupyterlab/main/setup_local.sh)
```

AI-powered transpilation

The AI-powered transpiler passes are passes that work as a drop-in replacement of "traditional" Qiskit passes for some transpiling tasks. They often produce better results than existing heuristic algorithms (such as lower depth and CNOT count), but are also much faster than optimization algorithms such as Boolean satisfiability solvers.

Install them:

```
pip install ‘qiskit-ibm-transpiler[ai-local-mode]’
```

More info https://quantum.cloud.ibm.com/docs/en/guides/ai-transpiler-passes
(👈 QR code)

AI-powered transpilation

An example of how to use them easily: **Hybrid heuristic-AI circuit transpilation**

```
from qiskit_ibm_transpiler import generate_ai_pass_manager
from qiskit.circuit.library import efficient_su2
from qiskit_ibm_runtime import QiskitRuntimeService


backend = QiskitRuntimeService().backend("ibm_torino")
torino_coupling_map = backend.coupling_map


su2_circuit = efficient_su2(101, entanglement="circular", reps=1)

ai_transpiler_pass_manager = generate_ai_pass_manager(
    coupling_map=torino_coupling_map,
    ai_optimization_level=3,
    optimization_level=3,
    ai_layout_mode="optimize",
)

ai_su2_transpiled_circuit = ai_transpiler_pass_manager.run(su2_circuit)]
```

More info https://quantum.cloud.ibm.com/docs/en/guides/ai-transpiler-passes
(👈 QR code)