

# Quantum circuit synthesis with diffusion models

F. Fürutter, G. Muñoz-Gil, H.J. Briegel, *Nat. Mach. Intell.* 6, 515–524 (2024)

In this work, we use generative denoising diffusion models (DMs) to produce desired quantum operations within gate-based quantum circuits, based on some input text defining the goal operation. DMs allow to sidestep during training the exponential overhead inherent in the classical simulation of quantum dynamics, a consistent bottleneck in preceding ML techniques. We demonstrate the model's capabilities across two tasks: entanglement generation and unitary compilation. The model excels at generating new circuits and supports typical DM extensions such as masking and editing to, for instance, align the circuit generation to the constraints of the targeted quantum device. Given their flexibility and generalization abilities, we envision DMs as pivotal in quantum circuit synthesis, enhancing both practical applications but also insights into theoretical quantum computation.

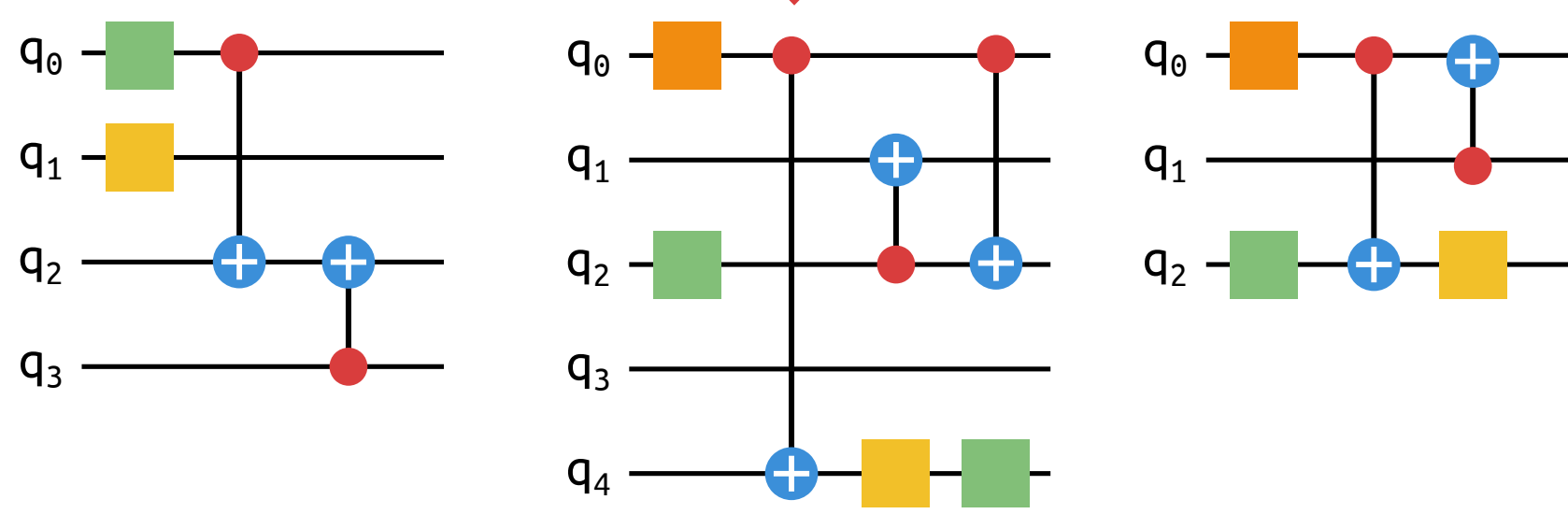
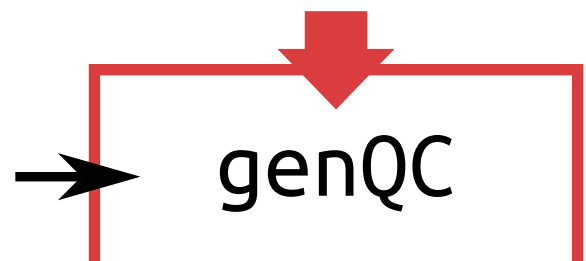
## Goal

I want a circuit that ...

... *compiles this unitary*

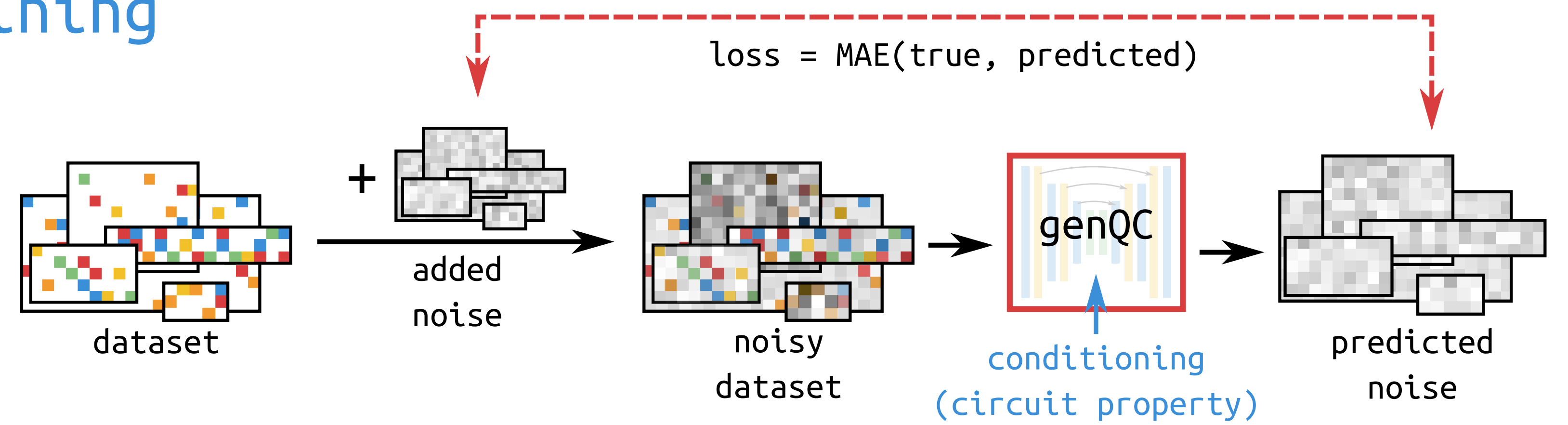
... *prepares this entangled state*

based on Stable diffusion



Flexible model able to produce circuits on demand for varied number of qubits and circuit lengths.

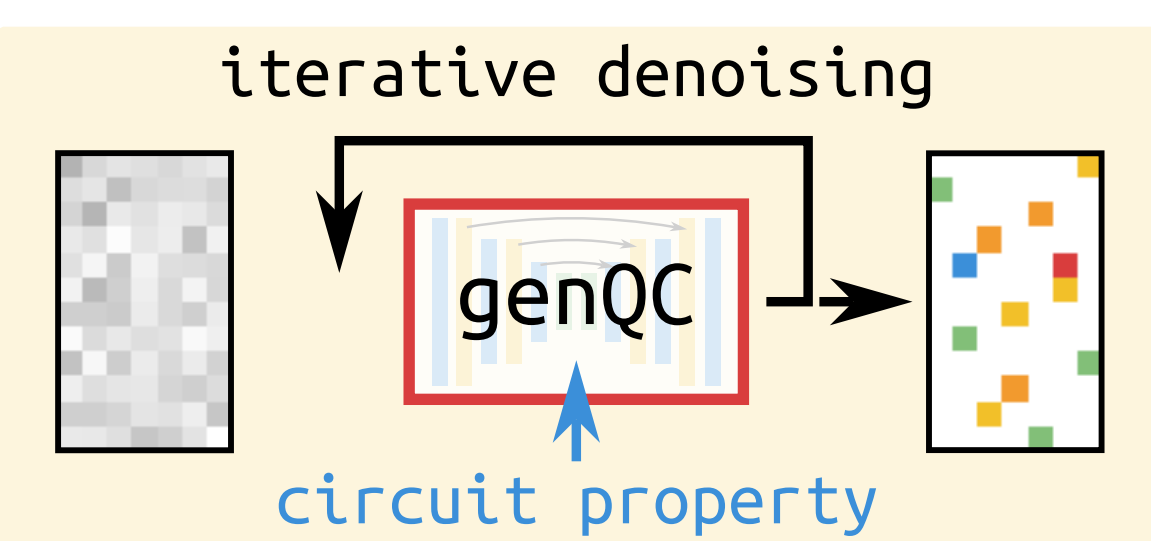
## Training



**Dataset:** pairs of circuit + property used as conditioning (i.e. unitary).

**Loss:** compares added white noise to the predicted one. No need of quantum computation here, hence no exponential bottleneck!

**Inference:** new circuits can be generated from purely noisy tensors. The input tensor defines circuit size and further constraints (see Masking / Editing).



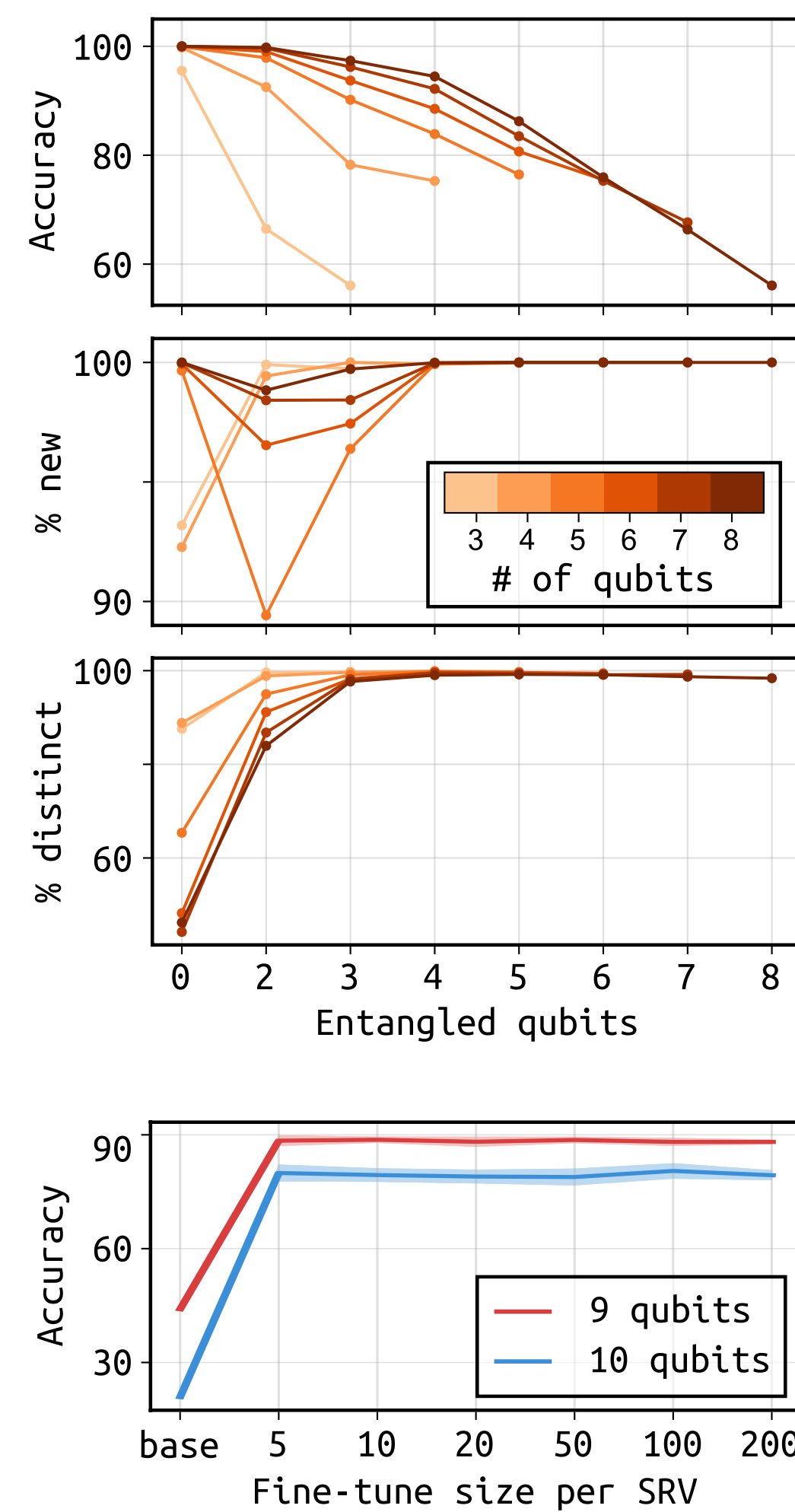
## Entanglement generation

**Goal:** generate circuits that create quantum states with entanglement, given by their Schmidt Rank Vector (SRV).

**Result:** genQC produces accurate, new and distinct circuits! Accuracy decreases with entanglement, due to circuit complexity.

**Scalability:** creating a training set for big circuits is prohibitively expensive.

Instead, we train with *easy* examples (low qubit number) and fine-tune the model with few *hard* ones (i.e., large qubit number).

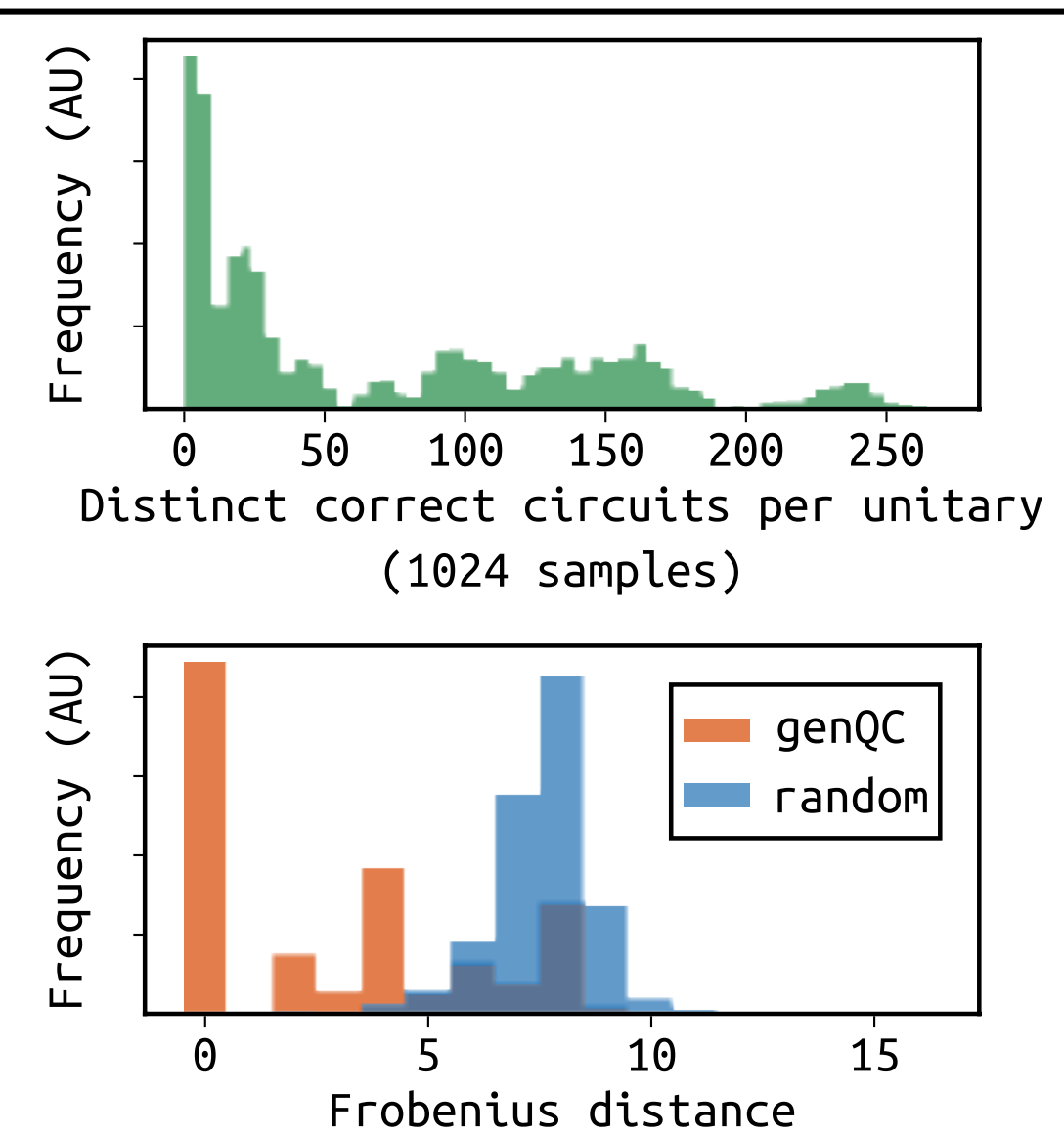
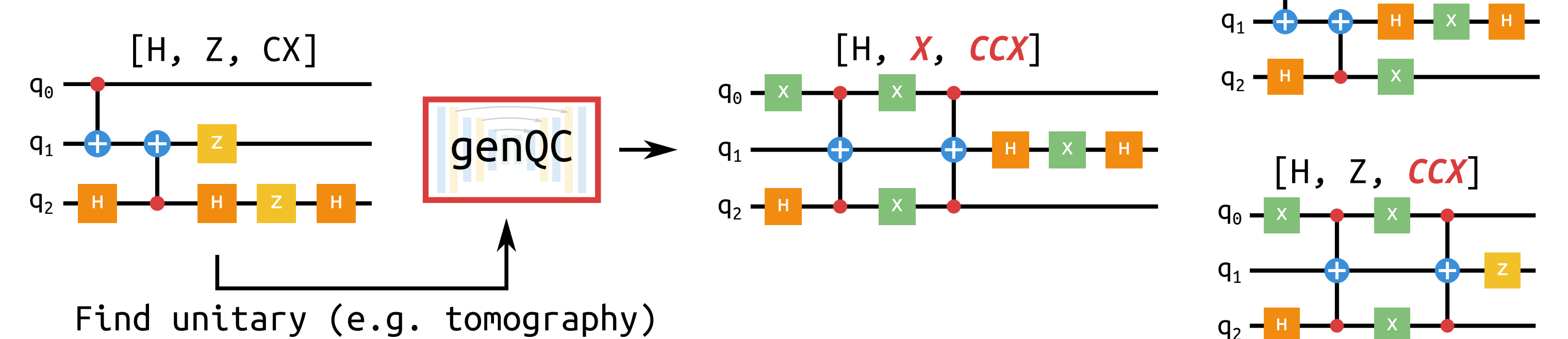


## Unitary compilation

**Goal:** generate circuits that compile a unitary  $U$  with given gate set (discrete gates here).

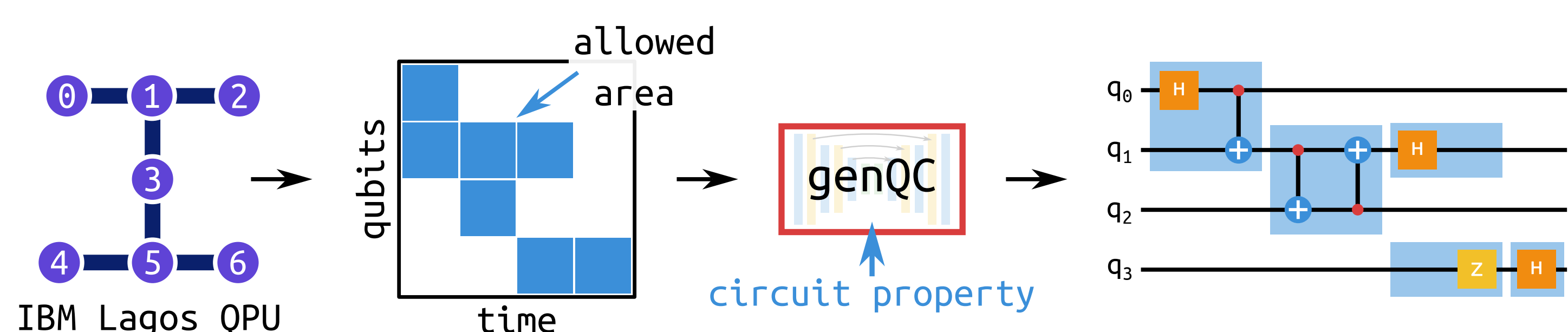
**Result:** genQC generates multiple valid circuit that *exactly* compile the given unitary. Wrong circuits are close and can be used as ansätze.

**Transpilation:** as gate set is an input, we can generate same unitaries with different gates and explore.



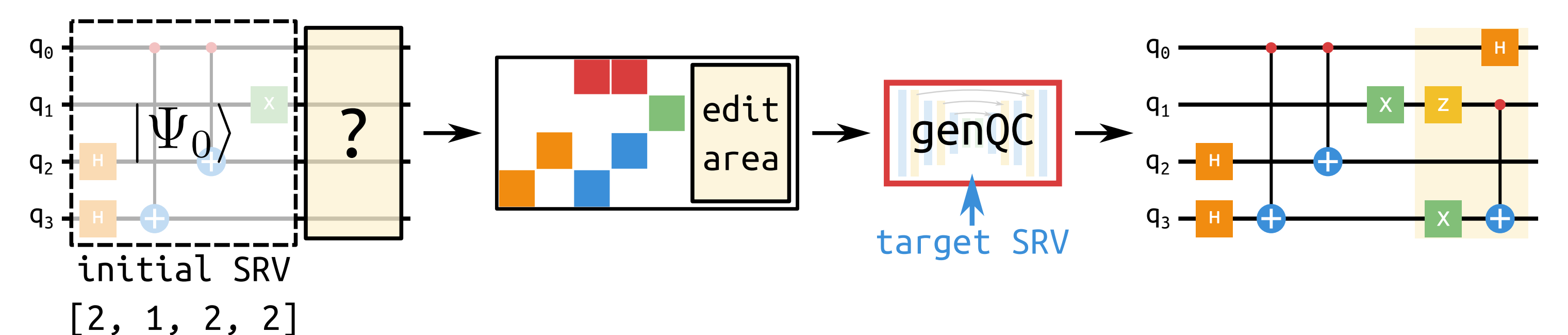
## Masking

Input tensor can be *masked* to prevent the model placing gates to, e.g. account for QPU connectivity:



## Editing

Similarly, we can prefix some gates and *edit* the rest to, e.g. consider input states and perform operations on them:



## Outlooks

- **Other platforms:** photons, qudits, MBQC,... Same model, just different tensor encodings!
- **Scaling:** bigger models for bigger circuits, trained on multiple tasks at the same time.
- **Interpretability:** attention mechanism gives a tool to understand the model's decisions.
- **Better conditioning:** either learnt (embeddings) or physically inspired (Hamiltonians).

## Resources:



## Acknowledgements:

