Quantum autoencoders for short depth quantum circuit synthesis

Jhonathan Romero, Jonathan Olson, Alan Aspuru-Guzik

Background: Initial state preparation can be a troublesome task for near-term quantum computers where circuit depth is the primary hurdle for useful quantum protocols. Because quantum gates are inherently noisy, improvements in near-term algorithms are typically associated with simply reducing the gate depth of a desired operation or a variational ansatz. For instance, algorithms like VQE [1] try to prepare approximate ground states of a molecular electronic Hamiltonian, but the size of the system they can simulate and the accuracy of the simulation is directly tied to the depth of the corresponding circuit ansatz. The goal of the method in this proposal is to apply a general technique that targets a circuit (or family of circuits) used for state preparation and attempts to find a reduced-depth circuit for preparing the same states.

We will subsequently refer to this procedure as Compressed Unsupervised State Preparation (CUSP). CUSP is intended to incrementally constrain a state or set of a states to a target manifold. CUSP uses an initial state preparation, together with a quantum autoencoder [2], to produce a mapping from the original manifold to a compressed latent space. The decoder circuit from the autoencoder is then used as a generative model to approximately reproduce the manifold. A final step attempts to refine the circuit parameters to improve the state preparation. The CUSP protocol can be performed on a quantum simulator as well, particularly when the original state preparation circuit is already too deep to be implemented on hardware, potentially enabling an experiment that could not have otherwise taken place.

In summary, a quantum autoencoder can be used in a way specified by this protocol to create shorter quantum circuits whose fidelity is greater or depth lower than the circuit on which it was trained.

CUSP Protocol: Formally, the CUSP protocol takes as input a set of k training states $\{|\psi_i\rangle\}_{i=1}^{k}$ from some parameterized family $\mathcal{F}_{\vec{\phi}}$ and returns a circuit $U_{AE}^{\dagger}G(\vec{\theta})$ which, when applied to $|0\rangle^{\otimes n}$, prepares states from the family $\mathcal{F}_{\vec{\phi}}$. The parameter vector $\vec{\phi}$ could be seen as either a manifold of physical states (e.g. derived from the set of ground states of a Hamiltonian, as in VQE) or directly as the parameters corresponding to the quantum circuit which prepares the state. To be precise, we will denote the circuit as $S(\vec{\phi}_i)$. Similarly, the vector $\vec{\theta}$ corresponds to the quantum circuit parameters in the latent space.

Note that if the family $\mathcal{F}_{\vec{\phi}}$ is zero-dimensional (i.e. $\vec{\phi}$ is just a single parameter setting), or equivalently when $k = 1$, the protocol simplifies to a problem that is similar to that of circuit compilation (though CUSP differs fundamentally in the execution of solving the problem). For simplicity in this text, we will still refer to these circuits as "parameterized" with the understanding that they are only trivially so, and will only bring up the distinction when there are notable considerations.

For clarity, we will often use the example of ground states prepared using VQE [1] throughout our description of CUSP as a motivating example. However, note that the protocol is an entirely general one, and could be applied to other families of quantum states, including ones that represent the output states of arbitrary parameterized circuits.

The general CUSP protocol (see Fig. 1 at the end of this document) is broken into three or four phases:

(1) - Training set preparation
(2) - Autoencoder training
(3) - Generative model search
(4) - (Optional) Refinement

Stage 1: In this stage, the choice of the target parameterized circuit and the desired k training states is made. This is effectively just choosing the input to CUSP. However, it is important to consider the fidelity or quality of these states and how that metric is measured, as these quantities will be needed for comparison throughout the protocol. For example, VQE uses expectation values that correspond to a ground state energy as a quality metric. The circuit parameters for preparation of these states are saved for use in the next stage.

Stage 2: This stage consists of training the QAE circuit using "trash state" training [2], with some exceptions and caveats. The first is that an appropriate circuit should be chosen that is conducive to running on the architecture of the target hardware's quantum processor. Because this protocol does not require any specific type of gate sequence, it can be utilized on any gate-model quantum architecture, including but not limited to superconducting xmon or transmon computers, optical quantum computers, ion-traps, and so on. This circuit will ideally have gates utilizing the same connectivity of the target hardware.

One caveat is related to the difficulty of training circuit parameters in a variational circuit [3]. There are several ways to avoid this pitfall; we enumerate some of them here:

Iteratively decrease the latent space: Rather than attempting to train the entire circuit at once, we can train a subset of circuit elements to reduce the latent space by e.g. a single qubit at a time. If we denote the circuit that disentangles the i-th qubit by U_i, then the overall circuit reducing the latent space by n qubits simply becomes $U_{AE} = U_n ... U_2 U_1$.

Informed initial guess: The initial settings of the autoencoder circuit U_{AE} could have some known setting which is close enough to the desired state that optimization from the initial setting is practicable.

Cost function design: If enough is known about the target state or manifold, another cost function could be substituted that has properties amenable to the optimization task.

Finally, the classical algorithm which governs the optimization of the circuit can be done with any number of numerical optimization methods. The success of

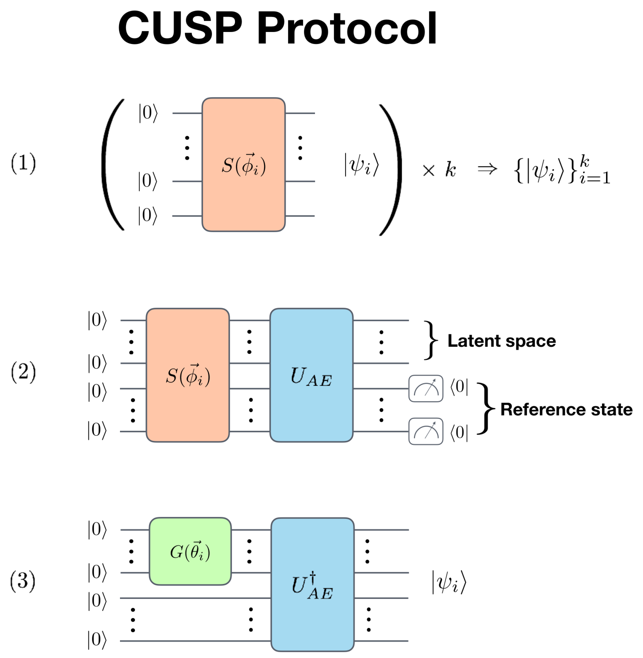

Figure 1: Outline for the CUSP algorithm. This algorithm consists of three major steps: (1) application of k different circuits to prepare k trial ground states, (2) training of the quantum autoencoder with the k trial states, producing a fixed circuit U_{AE} which compresses the input to a latent space, and (3) generation of the trial ground states using the latent space representations and unitaries $G(\vec{\theta})$. A fourth step can be added which allows $U^{\dagger}_{AE}$ to be additionally refined if there exists a method for measuring the quality of state preparation outside of using the original circuit itself as the measure of fidelity (for instance, minimizing the ground state energy in VQE).

these methods, however, may vary wildly across different manifolds and circuits, which one should take into account.

The output of stage (2) should be an encoder U_{AE} (which immediately implies a complementary decoder unitary $U^{\dagger}_{AE}$) that optimize the average fidelity of the k training states through the autoencoder network.

Stage 3: The goal of this stage is to find a parameterization of the quantum circuit in the latent space, i.e. $G(\vec{\theta})$, which is sufficient to prepare states in the training set, ideally generalizing to the entire linear span of the training set. In other words, we must optimize the parameters $\vec{\theta}$ relative to the cost function which determines the quality of the states that are prepared.

Note that for a particular circuit, there may be multiple ways to determine the quality of the prepared states. The available methods will largely depend on the algorithmic purpose of the states in question. For instance, in VQE, a series of measurements corresponds to some electronic energy which is then minimized. In general, one could use fidelity or state tomography to measure the quality. At this stage, one may also consider metrics that average over new examples of states not in the original training set, but which correspond to other states on the desired manifold (for example, see Figure 6 in [2]).

Stage 4: Because the QAE-decoded state preparation should by design have shorter gate depth than the original circuit, it may be possible in some instances to tune the parameters of the decoding unitary $U_{AE}^{\dagger}$ as well as the latent space unitary $G(\vec{\theta})$ to optimize the precision of the output state to the target state. This is possible because the noise inherent in applying the original state preparation circuit $S(\vec{\phi}_i)$ may have introduced errors in the parameters of the autoencoder circuit during training. Hence, if the metric for measuring the quality of the state does not re-use $S(\vec{\phi}_i)$, these errors might be removed by a final refinement of the autoencoder and latent space circuit parameters.

References

[1] Ian D Kivlichan, Jarrod McClean, Nathan Wiebe, Craig Gidney, Alán Aspuru-Guzik, Garnet Kin-Lic Chan, and Ryan Babbush. Quantum simulation of electronic structure with linear depth and connectivity. *Physical Review Letters*, 120(11):110501, 2018.

[2] Jonathan Romero, Jonathan P Olson, and Alan Aspuru-Guzik. Quantum autoencoders for efficient compression of quantum data. *Quantum Science and Technology*, 2(4):045001, 2017.

[3] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *arXiv:1803.11173 [quant-ph]*, 2018.