Dear Editor(s),

   We would like to thank you for considering our submission titled "NISQ Algorithm for Hamiltonian Simulation via Truncated Taylor Series" for publication in Scipost. We would also like to extend our gratitude to the referees for their valuable feedback and detailed reports. We believe we are able to provide a response to the critiques and comments raised by the referees. Please find attached our revised manuscript and, below, our response to the referees. The main comments we would like to address are summarized below:

1. We state explicitly the improvements our work makes on the existing published work in the field, and explain how we expect our work will be of benefit to the community. We believe our work provides a significant improvement to other NISQ algorithms for simulating dynamics in the literature as it uses a different framework to approaching the problem of time evolution. This allows it to bypass problems such as the barren plateau that have been shown to be one of the major problems in such approaches. Our algorithm also avoid a quantum-classical feedback loop, which cuts down on the number of circuits we need to run on the quantum computer by orders of magnitude. Also, our algorithm introduces a systematic way to generate a more expressible ansatz, using the structure of the Hamiltonian, which is absent in the existing NISQ algorithms for simulating dynamics.

2. We have elaborated on the scaling of our algorithm, both in our reply to the reviewers and in our main manuscript's appendix. The scaling of our algorithm is dependent on the number of moments we calculate. In the worst case, the number of moments scales linearly with the rank of the Hamiltonian. This is advantageous to other variational NISQ algorithms that rely on a parametric quantum circuit, especially since it is an open question how the other algorithms perform when scaled up. This scaling is fundamentally a result from complexity theoretic statements about the difficulty of obtaining an appropriate ansatz that is expressible enough and hence cannot be bypassed by any variational quantum simulation algorithm.

3. We explicitly list down the conditions, which if fulfilled, we expect our algorithm will have the capability to outperform classical methods. We also demonstrate an example of this with product states. While the example with product states is classically trivial to solve, we expect that with slightly better and larger quantum computers we can repeat the same examples with highly entangled states, which correspond to classically intractable dynamics. We also establish the advantage that our algorithm already outperforms most of the other NISQ dynamics algorithms.

4. Based on reviewers comments, we have worked on the presentation of our manuscript, taking the reviewers comments into consideration.

We hope that the improved manuscript meets the requirements for publication in Scipost.

Yours sincerely,
J.W.Z. Lau, T.Haug, and L. C. Kwek, K. Bharti

Response to Reviewer 1

Reviewer 1: One of the main motivations of the authors seems to be to get rid of the quantum-classical feedback loop of standard variational approaches for time evolution. However, in doing so it seems to me that an exponential overhead has been introduced. In my opinion, the question the authors have to answer is the following: How does the number of required basis states $|\chi\rangle$ grow as a function of time, and at which point does it reach the dimension of the Hilbert space?

Authors: We split the response to this question into 2 parts: Firstly, the issues leading us to want to find alternatives to the standard variational approaches, and secondly, the scaling in our algorithm.

1.Other variational algorithms like VQS[1,2,3], VFF[4,5], and their derivatives SVQS[6] and fsVFF[7] all rely on a quantum-classical feedback loop optimizing the parameters of a parametric quantum circuit, similar to the celebrated VQE algorithm for finding the ground state[8,9] Due to the fact that for a variatonal ansatz that relies on parameterized gates on the quantum computer, every time the circuit changes by changing the variational parameters, the quantum computer has to recalculate the result. Furthermore, in most of these variational algorithms, we also need more circuits to calculate gradients or related quantities, for the optimization to be performed. Thus, these methods introduce too many circuits that need to be evaluated on a quantum computer, with the number of circuits being a function of the number of variational parameters. In a previous work, we reported that just to simulate a small 2 qubit system with VQS on the IBM cloud quantum computer platform, we required around 500 circuits to be evaluated, which took us 3 days, mainly due to the waiting time in the queue .[10] From our own experience, this scaling is too much to be of practical use in the NISQ era. This is not to mention that the way most quantum computing platforms are being distributed right now (cloud services, which you may not have dedicated access too), this feedback loop could cause your classical computer to spend a large amount of time waiting for your circuits to be evaluated by the quantum computer, due to queueing.

This is not to mention the other problems in those algorithms, for example the potential to still be subject to the barren plateau problem[11,12] inherent in VQE, and the optimization being over a non-convex landscape that causes it to be extremely hard for the classical computer to optimize over once the system size grows past a few qubits. It has been shown that in variational algorithms that rely on a parametric quantum circuit, there will always be a tradeoff between trainability and expressibility.[13] This is not to mention that the current ansatz that have been used in such algorithms so far, hardware efficient ansatz, have not been shown to be capable of improving systematically the expressibility of the ansatz as the system size grows larger.

Our algorithm aims to deal with these problems. We aim to make a more practical algorithm, and thus mainly aim to compare our algorithm against these other algorithms. Although this is

[1]Ying Li and Simon C Benjamin. "Efficient variational quantum simulator incorporating active error minimization". In: *Physical Review X* 7.2 (2017), p. 021050.

[2]Xiao Yuan et al. "Theory of variational quantum simulation". In: *Quantum* 3 (2019), p. 191.

[3]Marcello Benedetti, Mattia Fiorentini, and Michael Lubasch. "Hardware-efficient variational quantum algorithms for time evolution". In: *arXiv preprint arXiv:2009.12361* (2020).

[4]Cristina Cirstoiu et al. "Variational fast forwarding for quantum simulation beyond the coherence time". In: *npj Quantum Information* 6.1 (2020), pp. 1–10.

[5]Benjamin Commeau et al. "Variational hamiltonian diagonalization for dynamical quantum simulation". In: *arXiv preprint arXiv:2009.02559* (2020).

[6]Kentaro Heya et al. "Subspace variational quantum simulator". In: *arXiv preprint arXiv:1904.08566* (2019).

[7]Joe Gibbs et al. "Long-time simulations with high fidelity on quantum hardware". In: *arXiv preprint arXiv:2102.04313* (2021).

[8]Alberto Peruzzo et al. "A variational eigenvalue solver on a photonic quantum processor". In: *Nature communications* 5 (2014), p. 4213.

[9]Jarrod R McClean et al. "The theory of variational hybrid quantum-classical algorithms". In: *New Journal of Physics* 18.2 (2016), p. 023023.

[10]Jonathan Wei Zhong Lau et al. "Quantum assisted simulation of time dependent Hamiltonians". In: *arXiv preprint arXiv:2101.07677* (2021).

[11]Jarrod R McClean et al. "Barren plateaus in quantum neural network training landscapes". In: *Nature communications* 9.1 (2018), p. 4812.

[12]Edward Grant et al. "An initialization strategy for addressing barren plateaus in parametrized quantum circuits". In: *Quantum* 3 (2019), p. 214.

[13]Zoë Holmes et al. "Connecting ansatz expressibility to gradient magnitudes and barren plateaus". In: *arXiv preprint arXiv:2101.02138* (2021).

a heuristic approach, similar to the other approaches, we believe that its practical benefits will be of interest to the community. As we mention in more detail in our manuscript, the problem of exponential overhead in the number of $|\chi\rangle$ states needed is related to a far more fundamental question of expressibility, which all variational NISQ algorithms will struggle with.

2. We have expanded the appendix to elaborate on the scaling of our argument. To summarize, in the worst case the number of states in the ansatz scales linearly with the rank of the Hamiltonian. We believe that this scaling is superior to other variational NISQ algorithms that require a parametric quantum circuit which have been previously published. In fact, it is an open question how variational quantum algorithms perform when scaling up and in general they also require an exponential number of parameters. This is now further discussed in our manuscript, where we added the connection to Krylov subspace for our method in the appendix. We provide a short summary here:

In the worst case, the size of the set CS_k scales as $O(r^K)$ for r terms in H. This is because our algorithm essentially relies on being able to calculate expectation values of powers of the Hamiltonian, $\langle\psi|H^k|\psi\rangle$ in an NISQ friendly manner. If we look at the Pauli string level (break our Hamiltonian into linear sums of Pauli strings), the number of Pauli terms in H^k grows exponentially in k. Right now, for current implementation of our algorithm on available quantum computers, this breaking into Pauli strings is necessary due to the imperfections in said quantum computers. However, we want to point out that if we allow for more complex quantum operations that are too difficult for NISQ era computers, such as complex controlled multi-qubit unitaries, the resources needed to measure $\langle\psi|H^k|\psi\rangle$ might scale more favourably.[14]

In general, this issue is fundamentally about obtaining an appropriate ansatz that is expressible enough. In other words, this is the problem of being able to generate an ansatz (whether it being one with parameterized gates or one that is a linear combination of quantum states) that has good enough parameters/quantum states to fully express the target quantum state. It is known that to prepare an arbitrary state on an n qubit quantum computer, we require a circuit depth of at least $\frac{1}{n}2^n$[15,16,17].[18] This tells us that it is very hard to produce an expressible enough ansatz in general. This is a complexity theoretic statement that cannot be bypassed by any quantum simulation algorithm based on parametric quantum circuits or linear combination of quantum states.

Thus, all current NISQ algorithms (with the exception of Trotter) currently face this problem of obtaining an expressible enough ansatz. In most variational algorithms like VQS, we create the ansatz with parameterized gates. To make it fully expressible, we would need an ansatz that has an exponential number of parameterized gates. In so doing, the variational algorithms faces with the problem of optimizing over an exponential number of parameters or solving a matrix differential equation, of which the size of the matrix also grows exponentially. Furthermore, the other algorithms also usually require an overparameterized ansatz with an exponential number of parameters to successfully simulate quantum dynamics. For the small examples where we compare our algorithm to pVQD (which is a similar algorithm that instead relies on a parametric quantum circuit), we note that our algorithm does not need an overparameterized ansatz, while usually other NISQ algorithms require one. Thus, heuristically, we have evidence that our

[14]Kazuhiro Seki and Seiji Yunoki. "Quantum power method by a superposition of time-evolved states". In: *PRX Quantum* 2.1 (2021), p. 010333.

[15]Emanuel Knill. "Approximation by quantum circuits". In: *arXiv preprint quant-ph/9508006* (1995).

[16]Mikko Möttönen et al. "Quantum circuits for general multiqubit gates". In: *Physical review letters* 93.13 (2004), p. 130502.

[17]Juha J Vartiainen, Mikko Möttönen, and Martti M Salomaa. "Efficient decomposition of quantum gates". In: *Physical review letters* 92.17 (2004), p. 177902.

[18]Martin Plesch and Časlav Brukner. "Quantum-state preparation with universal gate decompositions". In: *Physical Review A* 83.3 (2011), p. 032302.

algorithm requires less variational parameters.

One of the other major contributions of the TQS algorithm is that it also provides a systematic way to obtain a more and more expressible ansatz without suffering from the trainability issue. It has been shown that it is incredibly difficult to perform the training in such variational algorithms that rely on a changing quantum state .[19] We avoid this trainability problem as our training is done on a classical computer, with a quantum state that is not changing. Furthermore, the other variational algorithms still do not have a systematic method to generate an expressible enough trainable ansatz.

Reviewer 1: I heavily criticize the way the cumulative K-moment states are introduced. The authors do not connect them to the Taylor expansion of equation 3. It should be pointed out explicitly that if the Taylor expansion is applied to a state $|\psi\rangle$ the result is exactly a linear combination of cumulative K-moment states. At this point it becomes pretty clear that after long times it would be very difficult to generate a sufficiently expressible ansatz. Furthermore, the notation in equation (5) is sloppy. It should be made clear that the indices $i_K, ..., i_2, i_1$ all run from 1 to r. It is not mentioned that the number of basis states in this set is exponential in K

Authors: We thank the reviewer for the comment. We have updated the manuscript and appendix to better show the connection between the K-moment states and expressibility with the Taylor expansion, and changed the manuscript to further improve the notation. However, we would like to mention that this is a feature of the algorithm. Our method is essentially generating the Krylov subspace of H on the support of the initial state $|\psi\rangle$. When we choose a K, we are essentially constructing our ansatz with the Krylov subspace of our initial state up to H^K. And as the reviewer mentioned, this is taking advantage of the higher order terms in the Taylor expansion. By generating the ansatz in such a manner, we guarantee that the expressibility of our ansatz increases. This is in contrast to the other methods that other variational algorithms use to generate their ansatz, like the hardware efficient ansatz, which contain no guarantees that by adding in more layers, the expressibility will increase.

We would also like to mention again here that the problem of finding an expressible enough ansatz is fundamentally a complexity theoretic problem, present in all NISQ variational algorithms, be it based on linear combination of states or those based on parametric quantum circuits.

Reviewer 1: The authors test their approach on a couple of trivial examples: 2-qubit models that can be solved with pen and paper and a "classical" 4-spin example.

Authors: All examples of simulating quantum dynamics on current quantum computers so far have been on relatively trivial examples that can be solved with pen and paper. Shown below are the largest systems that other algorithms have simulated on current quantum hardware:

- Trotterization, largest system tested approximately 10 qubits, well known to have serious problems when simulating for anything more than a few time steps.[20,21]

- Variational Quantum Simulation, largest system tested is 2 qubits, no detailed study done so far on how it scales.[22]

[19]Eric R Anschuetz. "Critical Points in Hamiltonian Agnostic Variational Quantum Algorithms". In: *arXiv preprint arXiv:2109.06957* (2021).

[20]Esteban A Martinez et al. "Real-time dynamics of lattice gauge theories with a few-qubit quantum computer". In: *Nature* 534.7608 (2016), pp. 516–519.

[21]Rami Barends et al. "Digitized adiabatic quantum computing with a superconducting circuit". In: *Nature* 534.7606 (2016), pp. 222–226.

[22]Chee-Kong Lee et al. "Simulating Energy Transfer in Molecular Systems with Digital Quantum Computers". In:

- Variational Fast Forwarding, largest system tested is 2 qubits, no detailed study done so far on how it scales.[23]

Attempts to scale up the sizes of the systems have proved difficult due to various problems, such as the noise inherent in current quantum computers limiting the lengths of the circuits that can be run on current quantum computers. The question we want to answer if it is possible to develop an algorithm that circumvents some of these problems. In our manuscript, we proposed a new algorithm that avoids long circuits, complicated multi-qubit controlled-unitary gates, and the barren plateau problem. This provides hope that some non trivial examples can be solved in the near future, once better NISQ quantum computers are developed in the next few years. In our manuscript we now give an explicit example where our algorithm performs superior to other NISQ algorithms. In figure 6, we calculate the evolution under a Hamiltonian consisting of 7 random Pauli strings for thousands of qubits. We require only a limited number of overlaps in this case. By using highly entangled states instead of product states for the initial state, the evolution is intractable to simulate with classical methods such as tensor networks. Our algorithm provides hope that with better quantum computers that can be developed in the near future, we can outperform classical approaches along this line, as our algorithm does not require deep circuits, highly connected qubit architecture, or complicated controlled multi-qubit unitary gates, and only relies on sampling expectation values of Pauli strings on an initial state.

In general, we believe that under the following conditions, our algorithm can provide quantum advantage for these problems:

- The basis states which are used to represent the initial quantum state are highly entangled. This will make calculation of corresponding overlaps classically hard, as it boils down to a circuit sampling task. Note that the Quantum Threshold Assumption (QUATH) by Aaronson and Chen[24] says that there is no polynomial-time classical algorithm which takes as input a random circuit C and can decide with success probability at least $\frac{1}{2} + \Omega\left(\frac{1}{2^n}\right)$ whether $|\langle 0^n|C|0^n\rangle|^2$ is greater than the median of $|\langle 0^n|C|x^n\rangle|^2$ taken over all bit strings x^n. In other words, the circuit sampling task is difficult and hence classical algorithms should not be able to compete with algorithms based on circuit sampling as system size scales. The quantum part of TQS is based on circuit sampling which is classically difficult.

- The Hamiltonian should be a linear combination of small number of unitaries or the set of basis states corresponding to the terms in the Hamiltonian should close fast or the Hamiltonian is a low rank matrix. As an example, for a system size with a multiple of 3 qubits, if we consider the Hamiltonian of the form $H = XYZXYZ...XYZ + YZXYZX...YZX + ZXYZXY...ZXY + XXXX...XXX$, the set spanned by the terms in the Hamiltonian is only size 8, implying we can only generate 8 terms in our Ansatz no matter how high our K is or our system size is. Our algorithm can easily simulate these Hamiltonians, requiring only a limited number of overlaps. However, evolution by Trotterized methods or other variational methods on a quantum computer would be very challenging as it requires deep circuits and many variational parameters. It would even be a challenge to come up with an expressible enough Ansatz for variational methods. As mentioned above and in our new appendix, we actually simulated our algorithm with up to 1000 qubits for cases constructed in this way, and show that it is still able

arXiv:2101.06879 (2021). URL: https://arxiv.com/abs/2101.06879; Stefano Barison, Filippo Vicentini, and Giuseppe Carleo. "An efficient quantum algorithm for the time evolution of parameterized circuits". In: *Quantum* 5 (July 2021), p. 512. ISSN: 2521-327X. DOI: 10.22331/q-2021-07-28-512. URL: https://doi.org/10.22331/q-2021-07-28-512.

[23]Gibbs et al., "Long-time simulations with high fidelity on quantum hardware".

[24]Scott Aaronson and Lijie Chen. "Complexity-theoretic foundations of quantum supremacy experiments". In: *arXiv preprint arXiv:1612.05903* (2016).

to obtain an expressible ansatz and capture the full dynamics. A further important example are the simulation of many-body scars, which can have a low-rank Krylov subspace and thus the ansatz space closes fast .[25] Simulating these highly interesting systems are well suited for our method.

- •The system size of interest should be beyond the reach of classical simulation methods, for example beyond a few dozen qubits.

Reviewer 1: Furthermore they investigate time evolution of an 8-spin Ising chain. According to Figure 3, for $K = 1$ and $K = 2$ the fidelity drops quickly to zero (as expected) while the $K = 3$ moment expansion is able to capture the time evolution exactly. The Hamiltonian in equation (21) consists of 15 Pauli strings. If I count correctly, the number of basis states in the $K = 3$ moment expansion is given by $15^3 + 15^2 + 15 + 1 = 3616$, which by far exceeds the Hilbert space dimension of $2^8 = 256$. So it seems to be no surprise at all that this is able to capture the dynamics. None of this is mentioned by the authors.

Authors: We would like to clarify that we only take unique Pauli strings within the set. For example, if we consider the Hamiltonian $H = X_1Y_2Z_3 + Y_1Z_2X_3 + Z_1X_2Y_3 + X_1X_2X_3$, we will get $4 + 1$ states for K=1. However, for K=2, instead of getting $4^2 + 4 + 1$ states, we realize that only 8 unique Pauli strings can be generated ($X_1Y_2Z_3, Y_1Z_2X_3, Z_1X_2Y_3, X_1X_2X_3, I_1Z_2Y_3, Z_1Y_2I_3, Y_1I_2Z_3, I_1I_2I_3$). Thus, we end up with only 8 states in our K=2 ansatz for this case.

In the specific case the author mentioned that we used for 8 qubits, the number of states we end up with is given in table I in our appendix. For the $K = 3$ case, instead of ending up with 3616 states, we only consider 137 states. While this number is of the same order as the Hilbert space dimension, we note that it does not exceed the dimension.

Reviewer 1: What is exactly the role of the initial state $|\psi\rangle$? The authors talk about an "efficiently preparable quantum state". What happens if this a product state in the computational basis? In this case, it seems to me, the matrices E and D can just be calculated with pen and paper. At this point, there is no need for a quantum computer/simulator. So apparently, the algorithm proposed by the authors is only meaningful if the initial state $|\psi\rangle$ is a nontrivial (possibly highly entangled) state. Do the authors have any specific application in mind? Again, none of this is discussed.

Authors: We thank the reviewer for the comment. We agree that if the initial state $|\psi\rangle$ is a product state in the computational basis, the matrices are trivial and can be easily written down. The idea is that we want to choose an initial state that is highly entangled that cannot be easily written down, allowing us to study the dynamics of such highly entangled states with our algorithm. Examples of such efficiently preparable initial states using hardware efficient circuits are shown in our manuscript, in figure 7 and 8 in our appendix. These circuits were used in the manuscript. If we randomly choose the parameters Θ in such circuits, these circuits produce highly random states where the calculation of the E and D matrices is difficult for classical computers. In Fig.6, we demonstrate how our algorithm can simulate dynamics even for thousands of qubits. To facilitate classical computations we use a product state as demonstration, however using future quantum computers with more qubits one could calculate the dynamics for highly entangled states which are intractable for classical methods.

Reviewer 1: Is the algorithm as outlined on the right column of page 4 actually correct as its written down? In point 2, the authors say that E and D are measured for a fixed $K > 0$ and the job of the quantum computer is done. At stage 3 it is said, that if the fidelity is not satisfying, K has to be increased.

[25]Maksym Serbyn, Dmitry A Abanin, and Zlatko Papić. "Quantum many-body scars and weak breaking of ergodicity". In: *Nature Physics* 17.6 (2021), pp. 675–685.

Authors: We thank the referee for the comment. Once we choose a K, it is akin to choosing a particular ansatz. Once that is done, in our algorithm, we only require the usage of the quantum computer at the start, unlike other variational algorithms that require the usage of it through the simulation. However, if after we run the algorithm, we are not satisfied with the results fidelity, we can choose a more expressible ansatz by going to a higher K. To clarify, we removed the step about increasing K from the algorithm itself.

Authors: We hope that through our clarifications and the additions to our manuscript, the reviewer can see the value we hope to bring to the community with our paper. Above and through addition to our manuscript, we explain how this algorithm is an improvement over existing algorithms in the NISQ era and what examples we expect a possible advantage might come from. We would also like to state that we never claimed that a hybrid quantum-classical feedback loop is inefficient since the quantum computer must wait for the output of the classical computer. Indeed, most of the inefficiency of such a loop is in the classical computer waiting for the output of a quantum computer, especially if the quantum computer is delivered through a cloud computing platform where one might not have dedicated usage of the computer. As mentioned above in our responses to other questions, our algorithm removes the quantum-classical feedback loop and reduces the number of circuits needed to be evaluated by orders of magnitude (at least heuristically).

Response to Reviewer 2

Authors: We thank the reviewer for the comments. We have updated the manuscript and the appendix to further elaborate on these points. For convenience, we reproduce the main points here.

Our method is essentially generating the Krylov subspace of H. The Krylov subspace idea is a powerful method used in many classical algorithms to find approximate solutions to high-dimensional linear algebra problems, two examples being the Lanczos method for solving closed systems and the Generalized minimal residual method for solving the Lindblad master equation. Our approach is motivated by this observation from classical theory. It is known that the the Krylov subspace spans the entire space when you exponentiate the Hamiltonian H to the power of $K - 1$, where $K - 1 = rank(H)$. Thus, the number of states that we require in our Ansatz scales linearly with the rank of the Hamiltonian in the worst case.

We believe that this scales favourably compared to other NISQ algorithms previously mentioned in this paper, like VQS, VFF, and pVQD, which depend on parameterized variational quantum circuits and which do not contain any bounds on their scaling. The statement that this algorithm is exceptionally faster than the feedback loop is a practical statement and not a mathematically rigorous statement at this stage. Our belief in this statement stems from the fact that the other algorithms require a quantum-classical feedback loop and generate their ansatz in a non systematic way, which usually leads to a heavily over parameterized ansatz. Both these factors cause the number of circuits needed to be evaluated on the quantum computer in those algorithms to be far more than what we require, and for a few simple examples we show the evidence in our paper (comparing our algorithm to pVQD). From our testing where we compare our algorithm to a similar NISQ algorithm that requires a feedback loop (pVQD) for a simple 2 qubit system, the TQS algorithm only required ≈ 30 circuits to be run, while the pVQD algorithm required well over 4000 circuits to be run. Considering the time overhead involved in a quantum-classical feedback loop, especially when accessing the quantum computer through a cloud computing service (classical computer having to wait for the results of the quantum computer, having to queue up when we do not have dedicated access to the quantum computer), we believe that this is a practical statement to make. We also demonstrate running our algorithm for larger systems on actual quantum computers compared to other NISQ algorithms.

Also, if we consider the evolution of an arbitrary state by the time evolution operator $\exp{(-\iota H \Delta t)}$, by observing that the Taylor expansion involves powers of the Hamiltonian H $\left(\sum_{n=0}^{\infty} \frac{(-i \Delta t)^n}{n} H^n \right)$, it is clear that choosing the ansatz in such a way is suitable, as for a given k, the $\{|\chi_k\rangle\}_k$ states are essentially states in the Hilbert space of the space of $\frac{H^k |\psi\rangle}{\langle \psi | \psi \rangle}$.

We can further see that at longer times we would struggle with finding an expressible enough ansatz in the general case, as we need to keep considering higher powers of H. This is fundamentally an expressibility problem, present in all NISQ variational algorithms, be it based on linear combination of states or those based on parametric quantum circuits. It is known that to prepare an arbitrary state on an n qubit quantum computer, we require a circuit depth of at least $\frac{1}{n} 2^n$[26,27,28,][29] This tells us that it is very hard to produce an expressible enough ansatz to obtain unit fidelity for an arbitrarily long evolution time in general. This is a complexity theoretic statement that cannot be bypassed by all quantum simulation algorithm based on parametric quantum circuits or linear combination of quantum states.

In the worst case, the size of the set CS_k scales as $O(r^K)$ for r terms in H. This is because our algorithm essentially relies on being able to calculate expectation values of powers of the Hamiltonian, $\langle \psi | H^k | \psi \rangle$ in a NISQ friendly manner. If we look at the Pauli string level (break our Hamiltonian into linear sums of Pauli strings), the number of Pauli terms in H^k grows exponentially in k. Right now, for current implementation of our algorithm on available quantum computers, this breaking into Pauli strings is necessary due to the imperfections in said quantum computers. However, we want to point out that if we allow more complex operations that cannot be performed very well right now, such as complex controlled multi-qubit unitaries, the resources needed to measure such $\langle \psi | H^k | \psi \rangle$ values might scale more favourably .[30]

Reviewer 2: In the thirds step of the description of the algorithm △t is chosen based on knowledge of the eigenvalues of H, however diagonalizing a many-body Hamiltonian is a challenging task in its own

[26]Knill, "Approximation by quantum circuits".

[27]Möttönen et al., "Quantum circuits for general multiqubit gates".

[28]Vartiainen, Möttönen, and Salomaa, "Efficient decomposition of quantum gates".

[29]Plesch and Brukner, "Quantum-state preparation with universal gate decompositions".

[30]Seki and Yunoki, "Quantum power method by a superposition of time-evolved states".

Authors: We do not actually need to choose $\triangle t$ based on knowledge of the eigenvalues of H. We can just choose an extremely small value to start off with. The value of $\triangle t$ does not affect the number of circuits needed to be evaluated on the quantum computer and for all practical purpose can be taken to be very small, it is just to help us in our mathematical analysis.

Reviewer 2: A proposal to include higher orders in the power expansion of $U(\triangle t)$ should be accompanied with a warning that more quantum resources are needed, if not a full discussion about the scaling of such an increase. The inability to simulate long timescales is mentioned in the text as one of the drawbacks of currently available algorithms, how does the proposed algorithm compare?

Authors: We thank the referee for the suggestion. In our manuscripts and its appendix we now provide a more detailed discussion of the scaling. We also provide a simulation comparison between Trotterization and our algorithm, run on a noise model taken from a real noisy quantum computer provided through Qiskit, in figure 5. As can be seen, even for a small system, Trotterization already shows an inability to simulate for long timescales/many time steps on a noisy quantum computer, and our method compares favourably.

Reviewer 2: Although error mitigating techniques are mentioned for the data obtained through IBM, no error bars are shown in any of the plots, is it because they are too small to be shown? If so, this should be explicitly mentioned.

Authors: We used a error mitigation technique called Measurement Error Mitigation that involved characterizing the readout error of the quantum computer before we run our algorithm. This helps us obtain an error correction matrix that helps to mitigate the effect of imperfect readouts on our circuits. This method is well established and more details can be found in the Qiskit documentation here. This is more like a form of very restrictive tomography. If we are to obtain error bars, we would need to run our algorithm many times to obtain a large enough data set. However, this is quite expensive at this stage of quantum computation, and most other NISQ algorithm papers also do not run their algorithms many times to obtain error bars on the evolution. We ran our algorithm a few times, and did not see any noticeable difference in our results. Thus, we did not put in any error bars.

Reviewer 2: The data in Table 1 is very confusing. According to the notation in equation (5), for $K = 1, S_K$ has r elements (as many as Pauli strings in H). Are the values of K wrongly labelled, or is equation (5) wrong? Further, how can increasing K from 3 to 4 for the 4 quit case increase the number of basis states only by one?

Authors: We would like to clarify that we only take unique Pauli strings. For example, if we consider the Hamiltonian $H = X_1Y_2Z_3 + Y_1Z_2X_3 + Z_1X_2Y_3 + X_1X_2X_3$, we will get $4 + 1$ states for K=1. However, for K=2, instead of getting $4^2 + 4 + 1$ states, we realize that only 8 unique Pauli strings can be generated $(X_1Y_2Z_3, Y_1Z_2X_3, Z_1X_2Y_3, X_1X_2X_3, I_1Z_2Y_3, Z_1Y_2I_3, Y_1I_2Z_3, I_1I_2I_3)$. Thus, we end up with only 8 states in our K=2 ansatz for this case.

NISQ Algorithm for Hamiltonian Simulation via Truncated Taylor Series

Jonathan Wei Zhong Lau,[1, *] Tobias Haug,[2] Leong Chuan Kwek,[1, 3, 4] and Kishor Bharti[1, †]

[1] *Centre for Quantum Technologies, National University of Singapore 117543, Singapore*
[2] *QOLS, Blackett Laboratory, Imperial College London SW7 2AZ, UK*
[3] *National Institute of Education, Nanyang Technological University, 1 Nanyang Walk, Singapore 637616*
[4] *MajuLab, CNRS-UNS-NUS-NTU International Joint Research Unit, UMI 3654, Singapore*

Simulating the dynamics of many-body quantum systems is believed to be one of the first fields that quantum computers can show a quantum advantage over classical computers. Noisy intermediate-scale quantum (NISQ) algorithms aim at effectively using the currently available quantum hardware. For quantum simulation, various types of NISQ algorithms have been proposed with individual advantages as well as challenges. In this work, we propose a new algorithm, truncated Taylor quantum simulator (TQS), that shares the advantages of existing algorithms and alleviates some of the shortcomings. Our algorithm does not have any classical-quantum feedback loop and bypasses the barren plateau problem by construction. The classical part in our hybrid quantum-classical algorithm corresponds to a quadratically constrained quadratic program (QCQP) with a single quadratic equality constraint, which admits a semidefinite relaxation. The QCQP based classical optimization was recently introduced as the classical step in quantum assisted eigensolver (QAE), a NISQ algorithm for the Hamiltonian ground state problem. Thus, our work provides a conceptual unification between the NISQ algorithms for the Hamiltonian ground state problem and the Hamiltonian simulation. We recover differential equation-based NISQ algorithms for Hamiltonian simulation such as quantum assisted simulator (QAS) and variational quantum simulator (VQS) as particular cases of our algorithm. We test our algorithm on some toy examples on current cloud quantum computers. We also provide a systematic approach to improve the accuracy of our algorithm.

Digital quantum computers have made immense progress in recent years, advancing to solving problems considered to take an unreasonable time to compute for classical computers [1, 2]. Further, quantum computers are reaching the stage where quantum chemistry problems such as finding the ground state of certain molecules can be achieved within chemical accuracy [3]. In short, we are now in the Noisy Intermediate-Scale Quantum (NISQ) era [4, 5], which is characterized by quantum computers with up to a few hundred noisy qubits and lacking full scale quantum error correction. Thus, noise will limit the usefulness of the computations carried out by these computers [4], preventing algorithms that offer quantum advantage for practical problems, such as Shor's algorithm for prime factorization [6], from being implemented.

However, just because such algorithms cannot be implemented on NISQ devices does not mean that quantum advantage for practical problems cannot be found with NISQ devices. There is currently great interest in the quantum computing and quantum information community to develop algorithms that can be run on NISQ devices but yet deal with problems that are practical [5, 7, 8]. Some of the most promising avenues deal with the problems in many-body physics and quantum chemistry. One major problem in this field is to develop algorithms capable of estimating the ground state and energy of many-body Hamiltonians. To such ends, algorithms like variational quantum eigensolver (VQE) [9, 10] and quantum assisted eigensolver (QAE) [11, 12] have been proposed.

The other major problem is to be able to simulate the dynamics of these many-body Hamiltonians. This task can be extremely challenging for classical computers, and Feynman proposed that this would be one of the areas where quantum computers could exhibit clear advantages over classical computers [13]. Powerful methods to simulate quantum dynamics on fault-tolerant quantum computers have been proposed, such as the concept of using truncated Taylor series by Berry et al [14].

On NISQ devices, a standard approach in simulating quantum dynamics is to utilize the Trotter-Suzuki decomposition of the unitary time evolution operator into small discrete steps. Each step is made up of efficiently implementable quantum gates, which can be run on the quantum computer [15–21]. However, the depth of the quantum circuit increases linearly with evolution time and the desired target accuracy. On NISQ devices, fidelity rapidly decreases after a few Trotter steps [22], implying long time scales will be unfeasible to simulate with this method. Alternatives to Trotterization have been proposed, such as VQS [23–25], subspace variational quantum simulator (SVQS) [26], variational fast forwarding (VFF) [27, 28], fixed state variational fast forwarding (fsVFF) [29], quantum assisted simulator [30, 31] and generalized quantum assisted simulator (GQAS) [32] to name a few.

Recently, Otten, Cortes and Gray have proposed the idea of restarting the dynamics after every timestep by approximating the wavefunction with a variational

---

* e0032323@u.nus.edu
† kishor.bharti1@gmail.com

ansatz [33]. Building on that, Barison, Vicentini and Carleo have proposed a new algorithm [34] for simulating quantum dynamics. Their algorithm, named projected variational quantum dynamics (pVQD) combines the Trotterization and VQS approaches [23, 24]. They replace the differential equation with an optimization problem, although not well characterized, and require much simpler circuits compared to VQS. However, pVQD requires a quantum-classical feedback loop and might suffer from the barren plateau problem [35] as well the optimization problem may be computationally hard [36]. Further, the feedback loop mandates that one has to wait for each computation to finish before the next computation is run, which can be a major bottleneck on cloud-based quantum computers that are accessed via a queue.

Here, we propose the truncated Taylor quantum simulator (TQS) as new algorithm to simulate quantum dynamics. Our algorithm is building on the ideas of pVQD [33, 34] combined with the ansatz generation of QAS [30], which we further enhance by applying the concept of truncated Taylor series by Berry et al [14]. Our contributions and our algorithm are as following:

1. We recast the simulation of the quantum dynamics as a quadratically constrained quadratic program (QCQP). This optimization problem, unlike the optimization problem in pVQD, is well characterized and invites rigorous analysis. The QCQP in our algorithm admits a semidefinite relaxation [11]. Moreover, based on ideas from [11], one can provide a sufficient condition for a local minimum to be a global minimum, which a solver can further use as a stopping criterion. Since the classical optimization program in QAE is also a QCQP, it helps us achieve a conceptual unification of TQS with QAE.

2. The differential equations which form the classical part of QAS and VQS can be recovered in our framework. Since VQS is already a particular case of QAS [30], our approach yields both VQS and QAS as special cases of TQS.

3. We remove the need for the classical-quantum feedback loop in pVQD. The absence of the feedback loop yields our algorithm to be exceptionally faster than the feedback loop based NISQ algorithms for simulating quantum dynamics such as [23, 26–29].

4. Our algorithm avoids the trainability issues that plague other variational quantum algorithms. The choice of a problem-aware ansatz and the structure of the TQS algorithm helps bypass the barren plateau problem. It is known that in variational quantum algorithms that rely on a parametric quantum circuit, there will always be a trade-off between trainability and expressibility, implying that a highly expressible ansatz cannot be easily trainable [37]. In our case, we do not rely on parametric quantum circuits, thus we bypass this problem. Furthermore, our algorithm provides a systematic way to obtain a more expressible ansatz, which is missing in other algorithms.

TQS Approach— Let us first assume that the Hamiltonian H is expressed as a linear combination of r tensored Pauli matrices

$$H = \sum_{i=1}^{r} \beta_i P_i \,, \tag{1}$$

with coefficients $\beta_i \in \mathbb{C}$. The unitary evolution under the action of the Hamiltonian H for time Δt is given by

$$U(\Delta t) = \exp\left(-\iota H \Delta t\right) = \exp\left(-\iota \Delta t \sum_{j=1}^{r} \beta_j P_j\right) \tag{2}$$

$$= I - \iota \Delta t \left(\sum_{j=1}^{r} \beta_j P_j\right) - \frac{\Delta t^2}{2} \left(\sum_{j=1}^{r} \beta_j P_j\right)^2 + \mathcal{O}\left(\Delta t^3\right). \tag{3}$$

We do not need to implement the action of the unitary evolution in such a way. However, for purposes of describing the algorithm, we will use this power series expansion first, and talk more about alternatives later. We will now truncate the series, similar to [14]. If we choose small values of Δt with respect to the eigen energies of H, we can approximate the unitary evolution with $V(\Delta t)$

$$U(\Delta t) \approx I - \iota \Delta t \left(\sum_{j=1}^{r} \beta_j P_j\right) \equiv V(\Delta t). \tag{4}$$

Let us next choose the ansatz at time t as linear combination of elements from cumulative K-moment states, $\mathbb{CS}_K$ (refer to [30] for the formal definition). These states are defined in the same way as in [30] and will be constructed with the help of the given Hamiltonian, by essentially considering powers of the Hamiltonian. In terms of Pauli matrices, given a set of r tensored Pauli unitary matrices obtained from the unitary terms of the Hamiltonian $\mathcal{P} \equiv \{P_i\}_{i=1}^{r}$ and a positive integer K and some efficiently preparable quantum state $|\psi\rangle$, the K-moment states are the set of quantum states of the form

$$\{|\chi_i\rangle\}_i = \{P_{iK} \ldots P_{i2} P_{i1} |\psi\rangle\}_{iK=1,\ldots,i2=1,i1=1}^{r,\ldots,r,r}, \tag{5}$$

for $P_{il} \in \mathcal{P}$, where the indices i all run from 1 to r. We note that we only include unique states within the set $\{|\chi_i\rangle\}_i$. This corresponds to removing any repeated Pauli unitary in $\mathcal{P}$. It should also be mentioned that the way the K-moment states are being generated is closely related to the Taylor expansion of the time evolution operator. If we consider the evolution of an arbitrary state by the time evolution operator, by observing that the Taylor expansion involves powers of the Hamiltonian H, it is clear that choosing the ansatz in such a way is suitable, as the $\{|\chi_i\rangle\}_i$ states are essentially states in the

Hilbert space of $H^i |\psi\rangle$. This set is denoted by $\mathbb{S}_K$. The cumulative K-moment states $\mathbb{CS}_K$ are also defined in [30] as $\mathbb{CS}_K \equiv \cup_{j=0}^{K} \mathbb{S}_j$.

Now the ansatz is expressed as

$$|\psi\left(\alpha\left(t\right)\right)\rangle_K = \sum_{|\chi_i\rangle \in \mathbb{CS}_K} \alpha_i(t)|\chi_i\rangle, \qquad (6)$$

with some $\alpha_i \in \mathbb{C}$. For small values of Δt, the ansatz at time $t + \Delta t$ is given by

$$|\psi\left(\alpha\left(t + \Delta t\right)\right)\rangle_K = \frac{V\left(\Delta t\right)|\psi\left(\alpha\left(t\right)\right)\rangle_K}{\left(\langle\psi\left(\alpha\left(t\right)\right)|_K V^\dagger\left(\Delta t\right) V\left(\Delta t\right)|\psi\left(\alpha\left(t\right)\right)\rangle_K\right)^{\frac{1}{2}}}. \qquad (7)$$

Using the ideas in [34], our goal now is to variationally approximate the time evolution of the system by adjusting our variational parameters. The crucial difference in our case is that our variational parameters α are coefficients which do not change the basis quantum states $|\chi_i\rangle$. Thus, they can be solely updated via a classical computer and do not require a quantum-classical feedback loop. To evolve by time Δt, we update the α_i parameters to α_i' such that the following fidelity measure is maximized

$$F\left(\alpha'\right) = \frac{\left|\langle\psi\left(\alpha'\right)|_K V\left(\Delta t\right)|\psi\left(\alpha\right)\rangle_K\right|^2}{\langle\psi\left(\alpha\right)|_K V^\dagger\left(\Delta t\right) V\left(\Delta t\right)|\psi\left(\alpha\right)\rangle_K} \qquad (8)$$

Using the notation $|\phi\rangle = V\left(\Delta t\right)|\psi\left(\alpha\right)\rangle_K$, the expression for fidelity becomes

$$F\left(\alpha'\right) = \frac{\langle\psi\left(\alpha'\right)|\phi\rangle_K \langle\phi|\psi\left(\alpha'\right)\rangle_K}{\langle\phi|\phi\rangle}. \qquad (9)$$

Using the notation $W_\phi \equiv \frac{|\phi\rangle\langle\phi|}{\langle\phi|\phi\rangle}$, the above expression further simplifies to

$$F\left(\alpha'\right) = \langle\psi\left(\alpha'\right)|_K W_\phi|\psi\left(\alpha'\right)\rangle_K. \qquad (10)$$

The goal is to maximize the fidelity subject to the constraint that $\langle\psi\left(\alpha'\right)|\psi\left(\alpha'\right)\rangle = 1$. Thus, the optimization program at timestep t is given by

$$\max_{\alpha'} \ \langle\psi\left(\alpha'\right)|_K W_\phi|\psi\left(\alpha'\right)\rangle_K$$
$$\text{s.t. } \langle\psi\left(\alpha'\right)|\psi\left(\alpha'\right)\rangle_K = 1. \qquad (11)$$

Using the elements from $\mathbb{CS}_K$ and the Hamiltonian H, we define the overlap matrices $\mathcal{E}$ and $\mathcal{D}$ as the following

$$\mathcal{E}_{m,n} = \langle\chi_m|\chi_n\rangle, \qquad (12)$$
$$\mathcal{D}_{m,n} = \sum_j \beta_j \langle\chi_m|P_j|\chi_n\rangle. \qquad (13)$$

Because of the way the $|\chi_n\rangle$ states are constructed, these values can be easily computed on a quantum computer, as they simplify to the expectation values of Pauli strings acting on the original quantum state $|\psi\rangle$. The constraint

in the optimization program 11 can written in terms of α' as

$$\alpha'^\dagger \mathcal{E}\alpha' = 1. \qquad (14)$$

We proceed to write the objective in the optimization program 11 in terms of the overlap matrices $\mathcal{E}$ and $\mathcal{D}$. In first order, we can simplify the expression

$$\langle\phi|\phi\rangle = \langle\psi(\alpha)|_K \left(I + (\Delta t)^2 H^2\right)|\psi(\alpha)\rangle_K$$
$$= \alpha^\dagger \mathcal{E}\alpha + O((\Delta t)^2) \approx \alpha^\dagger \mathcal{E}\alpha. \qquad (15)$$

Further, using the notation $G \equiv (\mathcal{E} - \iota\Delta t\mathcal{D})$ we find

$$\langle\psi\left(\alpha'\right)|\phi\rangle_K \langle\phi|\psi\left(\alpha'\right)\rangle_K = \alpha'^\dagger G\alpha\alpha^\dagger G^\dagger\alpha'. \qquad (16)$$

Using Eq.14,15,16 and the notation $W_\alpha \equiv \frac{G\alpha\alpha^\dagger G^\dagger}{\alpha^\dagger \mathcal{E}\alpha}$, the optimization program in 11 can be re-expressed in terms of overlap matrices as

$$\max_{\alpha'} \ \alpha'^\dagger W_\alpha\alpha' \qquad (17)$$
$$\text{s.t } \alpha'^\dagger \mathcal{E}\alpha' = 1. \qquad (18)$$

The aforementioned optimization program is a quadratically constrained quadratic program with a single equality constraint. As described in [11], this QCQP admits a direct convex SDP relaxation. Moreover, the results from [11] provide a sufficient condition for a local minimum to be a global minimum, which a solver can further use as a stopping criterion. Alternatively, the problem can be solved with the classic Rayleigh-Ritz procedure by finding the eigenvector associated with the smallest eigenvalue λ of the generalized eigenvalue problem $-W_\alpha\alpha' = \lambda E\alpha'$.

It can be shown that in the limit of small Δt, TQS reduces to QAS (see Appendix C). This could potentially give us a way to obtain systematic higher-order corrections to the QAS matrix differential equation. Interestingly, this is a conceptual unification of the ground state problem (QAE) with the dynamics problem (QAS) in the quantum assisted framework. In QAE, finding the ground state and ground state energy of a Hamiltonian was formulated to become a QCQP. In TQS, the problem of simulating the dynamics is also given as a QCQP. This is conceptually satisfying as the problem of finding the dynamics is expressed as $e^{-itH}|\psi\rangle$, which is mathematically similar to using imaginary time evolution to finding the ground state via $e^{-\tau H}|\psi\rangle$. The aforementioned connection is also one of the primary justifications for ansatz selection in [12]. We note that as alternative it is possible implement the unitary evolution operator $U(\Delta t)$ directly instead of the Taylor series expansion of Eq.7, however this would require the usage of Hadamard tests (see Appendix D).

We want to emphasize again that the quantum computer is only required to measure the overlap matrices $\mathcal{E}$ and $\mathcal{D}$ at the start of the algorithm. No quantum-classical feedback loop for optimization is required. The
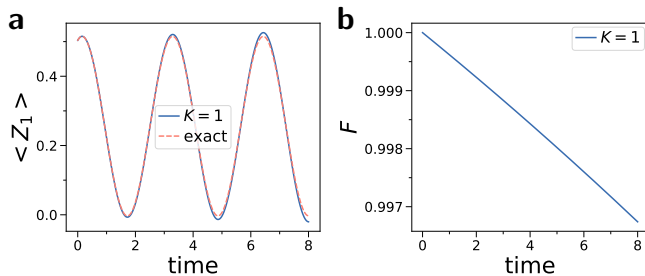
FIG. 1. Time evolution of TQS on a 2 qubit state, with Hamiltonian H_2, simulated on the IBM quantum processor *ibmq_rome*. **a)** Expectation value of $\langle Z_1 \rangle$ **b)** Fidelity of the state.

only optimization steps required are performed solely on the classical computer with knowledge of the overlap matrices. The algorithm is as follows:

1. Choose an efficiently implementable initial state $|\psi\rangle$, then choose some K>0 and form the unique K-moment states $|\chi_i\rangle$ to construct the ansatz.

2. With knowledge of the Hamiltonian H, calculate the overlap matrices $\mathcal{E}$ and $\mathcal{D}$ on the quantum computer. The job of the quantum computer is now done.

3. Choose a small Δt with respect to the eigenvalues of H and evolve the state forward in time using a classical computer, by solving the optimization program 17 subject to the constraint 18.

If a higher fidelity for the simulation is desired, one can increase K to acquire an ansatz with a higher expressibility. The timestep Δt could be increased by including higher order terms in the power series expansion of $U(\Delta t)$ in our calculations (Described in Appendix E).

Results— We first use TQS to simulate a 2 qubit Heisenberg model

$$H_2 = \frac{1}{2}X_1X_2 + \frac{1}{2}Y_1Y_2 + \frac{1}{2}Z_1Z_2. \qquad (19)$$

We apply it to evolve an initial randomized 2 qubit state $|\psi_2\rangle$. This initial state is generated by 5 layers of U_3 rotations and CNOT gates on the 2 qubits (see Appendix A). We ran the TQS algorithm on the 5-qubit quantum computer *ibmq_rome*, available through IBM Quantum Experience. We used error mitigation by calibrating the measurement errors and applying a filter obtained from that calibration on our data with the toolbox provided in Qiskit [38]. The results are shown in Fig.1. The evolution of the state under TQS reproduces the exact behavior very well for an ansatz with $K = 1$ moment states, even in the presence of the noise of a real quantum computer.

Next, we apply TQS to simulate a 4 qubit XX chain model on a quantum computer. Although this Hamiltonian is analytically solvable, we simulate this as a proof

of principle.

$$H_4 = \frac{1}{2}X_1X_2 + \frac{1}{2}X_2X_3 + \frac{1}{2}X_3X_4. \qquad (20)$$

In Fig.2, we simulate this Hamiltonian on *ibmq_rome* with an initial randomized 4 qubit state, generated by 5 layers of U_3 rotations and CNOT gates (see Appendix A). We run it for the $K = 1$ to $K = 3$ moment states. The evolution of the state under TQS again reproduces the exact behavior very well for the $K = 3$ case.
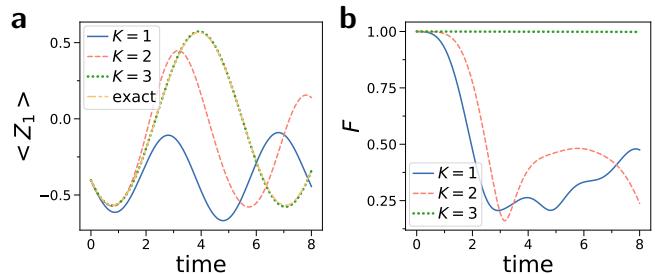
FIG. 2. Time evolution of TQS on a 4 qubit state with Hamiltonian H_4 simulated on the IBM quantum processor *ibmq_rome*. **a)** Expectation value of $\langle Z_1 \rangle$ **b)** Fidelity with exact solution.

Next, we investigate in Fig.3 the transverse Ising model with 8 qubits by simulating TQS on a classical computer.

$$H_8 = \sum_{i=0}^{6} \frac{1}{2}Z_iZ_{i+1} + \sum_{j=0}^{7} X_j. \qquad (21)$$

With an initial random state, we find that the evolution of the state reproduces the exact dynamics for the case of $K = 3$ moment expansion.

Lastly, we compare TQS to pVQD for a 2 qubit transverse Ising model on a simulation. We consider the 2
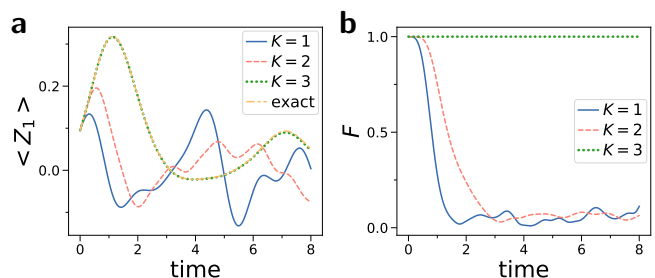
FIG. 3. Time evolution of TQS on a 8 qubit state, with Hamiltonian H_8, simulated on a classical computer, with a random initial state. The initial state was generated with 3 successive layers of U_3 rotations with randomized parameters on each qubit, followed by CNOT/entangling gates. This is further described in Appendix A. **a)** Expectation value of $\langle Z_1 \rangle$. **b)** Fidelity of the state.

qubit transverse Ising Hamiltonian:

$$H_{TFI,2} = \sum_{i=0}^{2} \frac{1}{2} Z_i Z_{i+1} + \sum_{j=0}^{2} X_j. \quad (22)$$

We compared them with noisy simulators, with the noise models taken from the IBM Quantum Experience provider, which is meant to mimic the noise on their actual quantum computers. The results are shown in Fig.4. As can be seen, while both TQS and pVQD do have errors when trying to simulate this Hamiltonian in the presence of noise, the results for the expectation values of the state for TQS are closer to the classical results most of the time. This is especially so for the expectation value of $\langle Z_1 \rangle$. However, while the results might be argued to be somewhat similar, the resource needs of both algorithms on the quantum computer are quite different. The TQS algorithm in our case required ≈ 30 circuits to be run, while the pVQD simulator required well over 4000 circuits to be run, which is already a little challenging for us to run on the IBM Quantum Experience. It should be mentioned that if we wanted to increase the simulation time for this example, since the algebra has already closed, we could do that with no extra circuits with TQS, while the number of circuits in pVQD scales linearly with the number of steps required. Furthermore, we only needed to use 4 variational parameters in the TQS case, while for the pVQD case we required 8 variational paramters. This behavior of TQS requiring less variational parameters to get a similar result seems to be consistent for the small models we tested, as the other variational algorithms usually need an over-parameterized ansatz when using hardware efficient ansatz. Maybe add a citation here

We also compare TQS to Trotterization on a noisy simulator, for a 2 qubit transverse Ising Hamiltonian.

$$H_{2,TFI} = Z_1 Z_2 + X_1 + X_2. \quad (23)$$

A simple Trotterization of the time evolution operator for this case is decomposed as:

$$e^{-i\tau H_{2,TFI}} \approx \prod_{i=1}^{N} \left(\left(e^{-i\delta t_i Z_1 Z_2} \right) \left(e^{-i\delta t_i X_1} e^{-i\delta t_i X_2} \right) \right), \quad (24)$$

with $\sum_{i=1}^{N} \delta t_i = \tau$. The results are shown in Fig.5. As can be seen, even for a simple case such as this, due to the circuit lengths in Trotter increasing linearly with the time, the circuit lengths rapidly grow too long to obtain any meaningful results from the quantum computer. This is in contrast to TQS, which is able to capture the dynamics faithfully for a high enough K (representing an expressible enough ansatz).

In Fig.6, we study our algorithm for up to thousands of qubits N. We use a Hamiltonian $H = \sum_{i=1}^{r} P_i$ that consists of r randomly chosen N-body Pauli strings
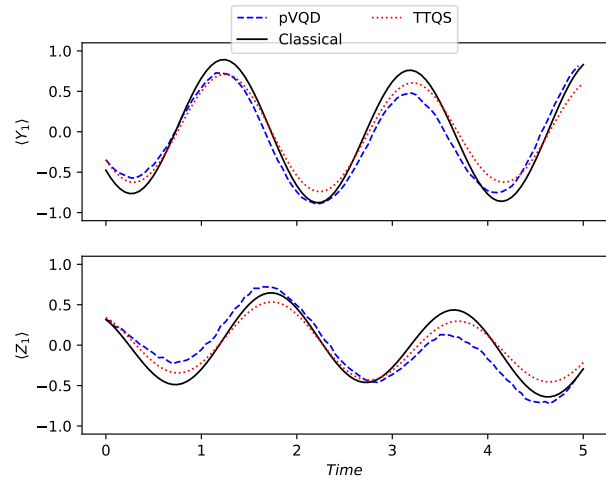
FIG. 4. Time evolution of TQS and pVQD on a 2 qubit state, with Hamiltonian $H_{TFI,2}$, simulated with a noisy simulator. The noise model was taken from the IBM Quantum Experience provider, mimicking the noise of the real quantum processor *ibmq_bogota*. pVQD was run for 100 optimization steps, and made use of a parametric quantum circuit with 8 parameters, made out of sucessive layers of single qubit X rotations and 2-qubit ZZ rotations. The expectation values of $\langle Y_1 \rangle$ and $\langle Z_1 \rangle$ are plotted.
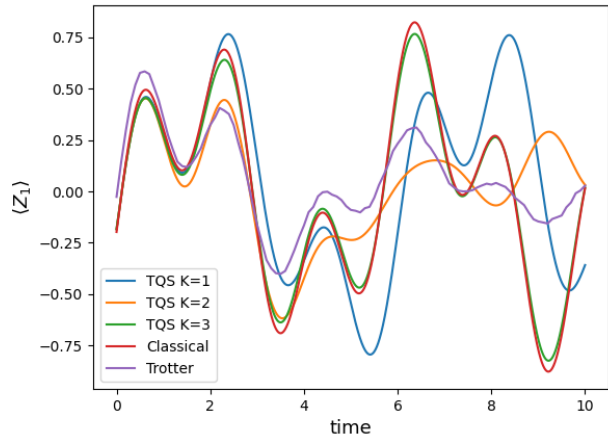
FIG. 5. Dynamics of $H_{2,TFI}$ compared between Trotterization and TQS. The noise model was taken from the IBM Quantum Experience provider, mimicking the noise of the real quantum processor *ibmq_bogota*. We used a total of 100 steps for the Trotterized run. The expectation value of $\langle Z_1 \rangle$ is shown.

$P_i = \otimes_{j=1}^{N} \boldsymbol{\sigma}_j$, where $\boldsymbol{\sigma}_j \in \{I, X, Y, Z\}$. The cumulative K-moment states close at order $K = r$ and yield the full ansatz space necessary to describe the dynamics exactly. We use the product state $|0\rangle^N$ as initial state for the dynamics. This choice makes the dynamics tractable for classical computation. However, choos-
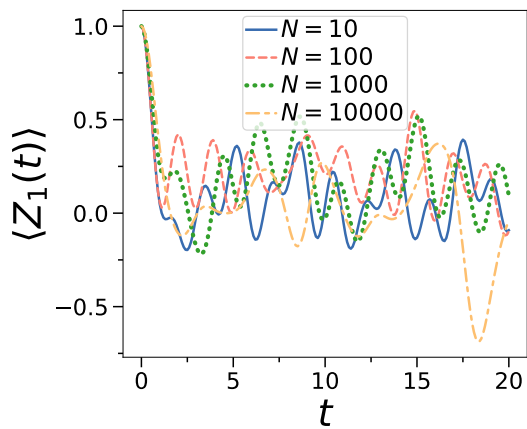
FIG. 6. Dynamics of Hamiltonians consisting of multi-body Pauli strings for varying number of qubits N. Hamiltonians are composed of r different random Pauli strings $H = \sum_{i=1}^{r} P_i$, where the Pauli strings $P_i = \otimes_{j=1}^{N} \boldsymbol{\sigma}_j$ consist of N tensored Pauli operators $\boldsymbol{\sigma}_j \in \{I, X, Y, Z\}$. The initial state $|\psi\rangle = |0\rangle^{\otimes N}$ is the N-qubit product state with all zeros. The cumulative K-moment states consists of $2^r = 128$ ansatz states and exactly captures the full dynamics.

ing an highly entangled initial state $|\psi\rangle$ would require a quantum computer to evaluate the overlaps. For such intractable states, our method provides a possible quantum advantage.

Discussion and Conclusion— The currently proposed NISQ algorithms face problems in scaling up to system sizes where classical computers cannot simulate the same systems, or in other words, to the point where we would see quantum advantage. For example, VQS/SVQS/pVQD require the use of a quantum-classical feedback loop, usually require complicated circuits, share similar problems as VQE like the barren plateau problem, and lack a systematic way to generate a parameterized ansatz. VFF/fsVFF also suffers from lacking a systematic way to generate the ansatz, usually requires complicated circuits and has to run a quantum-classical feedback loop it at the start. Further, the no fast-forwarding theorem suggests that not all Hamiltonians will be able to be accurately diagonalized with a reasonable amount of gates and circuit length, and the optimization step of the cost function in VFF might be too difficult to carry out efficiently. However, the barren plateau problem and ansatz state generation could be improved upon by applying various techniques [37, 39–42].

One problem that VQS and QAS share is that they require solving a differential equation which includes the pseudo-inverse of a matrix, whose elements are measured on a quantum computer. This matrix can be ill-conditioned. This procedure, via singular value decomposition, can be numerically unstable and sensitive to noise, especially as the system increases in size [43]. However, the sensitivity of these matrices has not been rigorously analyzed and more work has to be done to understand the scaling of the sensitivity.

In this work, we develop TQS for simulating quantum dynamics on digital quantum computers. TQS recasts the dynamical problem as a QCQP optimization program, which is well characterized unlike the optimization program in pVQD, allowing us to avoid the aforementioned problem in VQS and QAS.

At the same time, TQS retains the advantages of QAS, namely providing us a systematic method to select the ansatz, avoiding complicated Hadamard tests and controlled unitaries, avoiding the barren plateau problem, and only requiring usage of the quantum computer at the start, all of which are problems that are present in pVQD.

However, there are still many problems to tackle in our approach. One problem is an inherited problem from QAS. As the Hamiltonian size and complexity increase, large K values may be needed to generate enough states for a sufficiently expressible ansatz to produce accurate results. It is clear from the connection between the Taylor expansion of the time evolution operator and our K-moment states that in the general case, the further in time we want to simulate, the exponentially larger our ansatz should be and the harder the difficulty of generating that ansatz. However, this is fundamentally a complexity theoretic statement which can not be bypassed in the general case by any quantum simulation algorithm based on parametric quantum circuits (variational quantum algorithms) or linear combination of quantum states (our algorithm). Though our algorithm uses a problem aware ansatz, more information from the problem such as the combination coefficients β_i and symmetries of the Hamiltonian could be employed to further tame the complexity. A further discussion and analysis on the number of states needed is given in appendix B.

As the system size increases, it may be required to reduce Δt to preserve accuracy in the post-processing part of the algorithm. This will increase the computational cost of the classical computer. The number of optimization steps to be carried out increases linearly with the number of discretizations steps of the evolution time. Determining whether this poses a bottleneck for TQS when applied to large systems requires further studies.

Furthermore, in the presence of noise, the calculated fidelity of our states can go above one. A possible origin are small eigenvalues in the $\mathcal{E}$ overlap matrix, which can give the procedure of optimizing or solving the generalized eigenvalue problem numerical instability. As we scale up the system and consider more ansatz states, this issue can become more prevalent.

We expect our algorithm not to provide quantum advantage in the general case. However, we believe our algorithm is capable of providing quantum advantage over classical methods for certain cases. The conditions where we believe our algorithm will do so are:

- The basis states which are used to represent the initial quantum state are highly entangled. This

will render the calculation of corresponding overlaps classically hard, as it boils down to a circuit sampling task. Note that the Quantum Threshold Assumption (QUATH) by Aaronson and Chen [44] says that there is no polynomial-time classical algorithm which takes as input a random circuit C and can decide with success probability at least $\frac{1}{2} + \Omega\left(\frac{1}{2^n}\right)$ whether $|\langle 0^n|C|0^n\rangle|^2$ is greater than the median of $|\langle 0^n|C|x^n\rangle|^2$ taken over all bit strings x^n. In other words, the circuit sampling task is difficult and hence classical algorithms will not be able to compete with algorithms based on circuit sampling as system size scales. The quantum part of TQS is based on circuit sampling which is classically difficult.

- The Hamiltonian possesses a particular structure. For example, the Hamiltonian consists of a small number of unitaries, the Krylov subspace closes fast, or the Hamiltonian is a low-rank matrix. We demonstrated such an example for a Hamiltonian consisting of a limited amount of multi-body Pauli strings where our method can simulate the dynamics of thousands of qubits. These Hamiltonians would be challenging for other methods such as Trotter or variational quantum algorithms. First, simulating the multi-body interaction would require deep circuits to implement. Further, for the number of qubits we studied it is very difficult to find an expressible ansatz that captures the dynamics accurately. A further prominent example that has been of major interest recently are quantum many-body scars. They can arise when the Krylov subspace closes fast at a low order K [45]. Our algorithm is thus very capable to simulate these fundamental effect of many-body physics using NISQ computers.

- The system size of interest should be beyond the reach of classical simulation methods, for example beyond a few dozen qubits.

In the future, the NISQ community should investigate these challenges, so that we can successfully run NISQ algorithms for larger qubit numbers.

Acknowledgements— We are grateful to the National Research Foundation and the Ministry of Education, Singapore for financial support. The authors acknowledge the use of the IBM Quantum Experience devices for this work. This work is supported by a Samsung GRC project and the UK Hub in Quantum Computing and Simulation, part of the UK National Quantum Technologies Programme with funding from UKRI EPSRC grant EP/T001062/1.

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et al.*, Nature **574**, 505 (2019).

[2] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, *et al.*, Science **370**, 1460 (2020).

[3] G. A. Quantum *et al.*, Science **369**, 1084 (2020).

[4] J. Preskill, Quantum **2**, 79 (2018).

[5] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, *et al.*, arXiv preprint arXiv:2101.08448 (2021).

[6] P. W. Shor, in *Proceedings 35th annual symposium on foundations of computer science* (Ieee, 1994) pp. 124–134.

[7] I. H. Deutsch, arXiv preprint arXiv:2010.10283 (2020).

[8] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, *et al.*, arXiv preprint arXiv:2012.09265 (2020).

[9] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. Obrien, Nature communications **5**, 4213 (2014).

[10] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, New Journal of Physics **18**, 023023 (2016).

[11] K. Bharti, arXiv preprint arXiv:2009.11001 (2020).

[12] K. Bharti and T. Haug, arXiv preprint arXiv:2010.05638 (2020).

[13] R. P. Feynman, Int. J. Theor. Phys **21** (1982).

[14] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Physical Review Letters **114**, 090502 (2015).

[15] S. Lloyd, Science , 1073 (1996).

[16] B. P. Lanyon, C. Hempel, D. Nigg, M. Müller, R. Gerritsma, F. Zähringer, P. Schindler, J. T. Barreiro, M. Rambach, G. Kirchmair, *et al.*, Science **334**, 57 (2011).

[17] X. Peng, J. Du, and D. Suter, Physical review A **71**, 012307 (2005).

[18] R. Barends, A. Shabani, L. Lamata, J. Kelly, A. Mezzacapo, U. Las Heras, R. Babbush, A. G. Fowler, B. Campbell, Y. Chen, *et al.*, Nature **534**, 222 (2016).

[19] R. Barends, L. Lamata, J. Kelly, L. García-Álvarez, A. Fowler, A. Megrant, E. Jeffrey, T. White, D. Sank, J. Mutus, *et al.*, Nature communications **6**, 1 (2015).

[20] E. A. Martinez, C. A. Muschik, P. Schindler, D. Nigg, A. Erhard, M. Heyl, P. Hauke, M. Dalmonte, T. Monz, P. Zoller, *et al.*, Nature **534**, 516 (2016).

[21] L. M. Sieberer, T. Olsacher, A. Elben, M. Heyl, P. Hauke, F. Haake, and P. Zoller, npj Quantum Information **5**, 1 (2019).

[22] D. Poulin, M. B. Hastings, D. Wecker, N. Wiebe, A. C. Doherty, and M. Troyer, arXiv:1406.4920 (2014).

[23] Y. Li and S. C. Benjamin, Physical Review X **7**, 021050 (2017).

[24] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, Quantum **3**, 191 (2019).

[25] M. Benedetti, M. Fiorentini, and M. Lubasch, arXiv preprint arXiv:2009.12361 (2020).

[26] K. Heya, K. M. Nakanishi, K. Mitarai, and K. Fujii, arXiv preprint arXiv:1904.08566 (2019).

[27] C. Cirstoiu, Z. Holmes, J. Iosue, L. Cincio, P. J. Coles, and A. Sornborger, npj Quantum Information **6**, 1 (2020).

[28] B. Commeau, M. Cerezo, Z. Holmes, L. Cincio, P. J. Coles, and A. Sornborger, arXiv preprint arXiv:2009.02559 (2020).

[29] J. Gibbs, K. Gili, Z. Holmes, B. Commeau, A. Arrasmith, L. Cincio, P. J. Coles, and A. Sornborger, arXiv preprint arXiv:2102.04313 (2021).

[30] K. Bharti and T. Haug, arXiv preprint arXiv:2011.06911 (2020).

[31] J. W. Z. Lau, K. Bharti, T. Haug, and L. C. Kwek, arXiv preprint arXiv:2101.07677 (2021).

[32] T. Haug and K. Bharti, arXiv preprint arXiv:2011.14737 (2020).

[33] M. Otten, C. L. Cortes, and S. K. Gray, arXiv preprint arXiv:1910.06284 (2019).

[34] S. Barison, F. Vicentini, and G. Carleo, Quantum **5**, 512 (2021).

[35] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Nature communications **9**, 4812 (2018).

[36] L. Bittel and M. Kliesch, arXiv:2101.07267 (2021).

[37] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, arXiv preprint arXiv:2101.02138 (2021).

[38] H. Abraham *et al.*, "Qiskit: An open-source framework for quantum computing," (2019).

[39] T. Haug, K. Bharti, and M. Kim, arXiv preprint arXiv:2102.01659 (2021).

[40] K. Nakaji and N. Yamamoto, arXiv:2005.12537 (2020).

[41] T. Volkoff and P. J. Coles, Quantum Science and Technology **6**, 025008 (2021).

[42] H.-Y. Huang, K. Bharti, and P. Rebentrost, arXiv preprint arXiv:1909.07344 (2019).

[43] J. W. Demmel, Numerische Mathematik **51**, 251 (1987).

[44] S. Aaronson and L. Chen, arXiv preprint arXiv:1612.05903 (2016).

[45] M. Serbyn, D. A. Abanin, and Z. Papić, Nature Physics **17**, 675 (2021).

[46] E. Knill, arXiv preprint quant-ph/9508006 (1995).

[47] M. Möttönen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, Physical review letters **93**, 130502 (2004).

[48] J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, Physical review letters **92**, 177902 (2004).

[49] M. Plesch and Č. Brukner, Physical Review A **83**, 032302 (2011).

[50] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, Quantum **3**, 214 (2019).

[51] K. Seki and S. Yunoki, PRX Quantum **2**, 010333 (2021).

[52] A. McLachlan, Molecular Physics **8**, 39 (1964).

Appendix A: Details on running circuits on the IBM quantum computer

For the runs on the real quantum computer, we generated an initial state with randomized parameters to evolve with the following circuit. It comprised 5 layers of successive U_3 rotation with randomized parameters on each qubit, followed by a CNOT/entangling gate. We (see Fig.7 and 8). We sampled from each circuit 8192 shots.
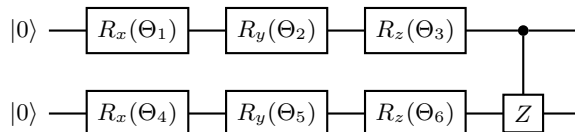
FIG. 7. Circuit for two qubits that generate one set of U_3 rotation with randomized parameters, followed by a CNOT gate between the 2 qubits. 5 successive layers of this circuit were used to generate the initial starting state for the 2 qubit case on the IBM quantum computer for our runs of TQS. The Θs were randomly generated.
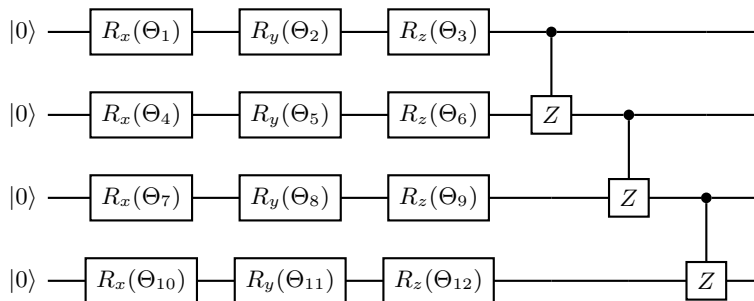
FIG. 8. Circuit for four qubits that generate one set of U_3 rotation with randomized parameters, followed by a series of CNOT gates between the adjacent qubits. 5 successive layers of this circuit were used to generate the initial starting state for the 4 qubit case on the IBM quantum computer for our runs of TQS. The Θs were randomly generated.

	$K=1$	$K=2$	$K=3$	$K=4$
2 Qubit Case	1	4		
4 Qubit Case	1	4	7	8
8 Qubit Case	1	17	137	

TABLE I. Comparison of the number of basis states used to construct the hybrid ansatz for each K for each Hamiltonian. For example, the $K=2$ expansion for the 4 qubit case, using the Hamiltonian H_4, requires 4 quantum states to construct the hybrid ansatz. We only considered unique states, which correspond to only taking unique Pauli strings. For example, in the 8 qubit case, while the number of Pauli strings in the Hamiltonian is 15, which might suggest the $K=3$ expansion generates $15^3 + 15^2 + 15 + 1 = 3616$ Pauli strings and thus 3616 states, many of them are repeated and only 137 of those strings are unique. Thus, we only end up having 137 states in our ansatz.

Appendix B: Number of basis states considered for each K, and discussion on scaling

The number of basis states that was used to construct the hybrid ansatz, for each K moment expansion, for each Hamiltonian, is given in Table I.

Given a scalar τ, an $N \times N$ matrix A and an $N \times 1$ vector v, the action of the matrix exponential operator $\exp(\tau A)$ on v can be approximated as

$$\exp(\tau A) v \approx p_{K-1}(\tau A) v, \tag{B1}$$

where p_{K-1} is a $K-1$ degree polynomial. The approximation in equation B1 is an element of the Krylov subspace,

$$Kr_{K-1} \equiv span\left\{v, Av, \cdots, A^{K-1}v\right\}. \tag{B2}$$

Thus, the problem of approximating $\exp(\tau A) v$ can be recast as finding an element from Kr_K. Note that the approximation in equation B1 becomes exact when $K-1 = rank(A)$. In our case, we can identify v with the initial state $|\psi\rangle$, τ with $-\iota t$ and A with the Hamiltonian H.

In the worst case, the number of overlaps scales as $O(r^K)$ for r terms in H. By observing the Taylor expansion of the time evolution operator $\exp(-iH\Delta t)$, we can see that at longer times we would struggle with finding an expressible enough ansatz in the general case, as we need to keep considering higher powers of H. This is fundamentally an expressibility problem, present in all NISQ variational algorithms, be it based on linear combination of states or those based on parametric quantum circuits. It is known that to prepare an arbitrary state on an n qubit quantum computer, we require a circuit depth of at least $\frac{1}{n}2^n$ [46–49]. This suggests that it is very hard to produce an expressible enough Ansatz to reproduce an arbitrary quantum state in the Hilbert space.

It is known that the the Krylov subspace spans the entire space when you exponentiate the Hamiltonian H to the power of $K-1$, where $K-1 = rank(H)$. Thus, the number of states that we require in our Ansatz scales linearly with the rank of the Hamiltonian. We believe that this scales favourably compared to other NISQ algorithms such as VQS and VFF.

Furthermore, one of the major contributions of the TQS algorithm is that, by using this problem-aware Ansatz, it provides a systematic way to obtain a more and more expressible Ansatz. The other variational algorithms like VQS and VFF still do not have a systematic method to generate an expressible enough Ansatz, or to improve on an Ansatz in a efficient way. Also, it has been shown that if we use a hardware efficient Ansatz, we would in general expect to encounter the barren plateau problem, which makes it very hard for the algorithm to train and optimize [35, 50]. Furthermore, the usual technique of using more and more layers of hardware efficient Ansatz circuits gives no guarantee that it will become more and more expressible in an efficient manner, when compared to the number of variational parameters that we are adding. There is also no guarantee that this will indeed improve the appropriateness of the Ansatz. This is especially true for larger systems. In TQS, with the way we generate the Ansatz with K moment states, we can see that at worst, we get an Ansatz with as many states as the size of the Hilbert space, which is fully expressible. This is due to the group of Pauli strings closing on itself eventually. Also, we can see that as we increase the K, we will definitely improve our Ansatz and get to a point where it is eventually expressible enough. In future, using the coefficients of the terms in the Hamiltonian, we expect to be able to slow down the growth of the number of states.

Our algorithm also relies on being able to calculate expectation values of powers of the Hamiltonian, $\langle\psi|H^k|\psi\rangle$ in an efficient manner. If we look at the Pauli string level (break our Hamiltonian into linear sums of Pauli strings), the number of Pauli terms in H^k grows exponentially in k. Right now, for current implementation of our algorithm on available quantum computers, this breaking into Pauli strings is necessary due to the imperfections in said quantum computers. However, if we allow more complex operations that cannot be performed very well right now, such as complex controlled unitaries, the resources needed to measure such $\langle\psi|H^k|\psi\rangle$ values might scale less [51].

We would also like to mention that depending on the Lie algebra of the Pauli terms in the Hamiltonian and the rank of the Hamiltonian, the number of required overlaps can be a lot smaller compared to the upper bound. By considering specific kinds of Hamiltonians, the number of states needed will be manageable. As an example, for a system size with a multiple of 3 qubits, if we consider the Hamiltonian of the form $H = XYZXYZ...XYZ + YZXYZX...YZX + ZXYZXY...ZXY + XXXXX...XXX$, the set of K-moment states is maximally size 8, implying that 8 ansatz states are sufficient to simulate the dynamics with our algorithm .

Appendix C: QAS and VQS as special cases of TQS

In this appendix, we show that in the limit of choosing a very small Δt, one obtains QAS from TQS. Since VQS is a special case of QAS [30], we get VQS also as special case of TQS. We start out with the series expansion of $|\psi(\vec{\alpha} + \delta\vec{\alpha})\rangle$

$$|\psi(\vec{\alpha} + \delta\vec{\alpha})\rangle = |\psi(\vec{\alpha})\rangle + \sum_j \frac{\partial}{\partial\alpha_j} |\psi(\vec{\alpha})\rangle \, \delta\alpha_j. \tag{C1}$$

Now in TQS we want to maximize the overlap of $U(\Delta t) |\psi(\vec{\alpha})\rangle$ and $|\psi(\vec{\alpha} + \delta\vec{\alpha})\rangle$, which is essentially the fidelity measure in equation 8

$$|\langle\psi(\vec{\alpha})| U^\dagger(\Delta t) |\psi(\vec{\alpha} + \delta\vec{\alpha})\rangle|^2 = \left[\langle\psi(\vec{\alpha})| U^\dagger(\Delta t) |\psi(\vec{\alpha})\rangle + \sum_j |\psi(\vec{\alpha})\rangle U^\dagger(\Delta t)\frac{\partial |\psi(\vec{\alpha})\rangle}{\partial\alpha_j}\delta\alpha_j\right] \times [\text{C. C.}]$$

$$\overset{|\psi(\vec{\alpha})\rangle = \sum_j \alpha_j |\chi_j\rangle}{=} \left[\langle\psi(\vec{\alpha})| U^\dagger(\Delta t) |\psi(\vec{\alpha})\rangle + \sum_j |\psi(\vec{\alpha})\rangle U^\dagger(\Delta t) |\chi_j\rangle \delta\alpha_j\right] \times [\text{C. C.}]$$

$$= |\langle\psi(\vec{\alpha})| U^\dagger(\Delta t) |\psi(\vec{\alpha})\rangle|^2 + \sum_j \langle\psi(\vec{\alpha})| U^\dagger(\Delta t) |\chi_j\rangle \langle\psi(\vec{\alpha})| U(\Delta t) |\psi(\vec{\alpha})\rangle \delta\alpha_j$$

$$+ \sum_j \langle\chi_j| U(\Delta t) |\psi(\vec{\alpha})\rangle \langle\psi(\vec{\alpha})| U^\dagger(\Delta t) |\psi(\vec{\alpha})\rangle \delta\alpha_j^* + \sum_{j,k} \langle\psi(\vec{\alpha})|U^\dagger(\Delta t)|\chi_j\rangle \langle\chi_k|U(\Delta t)|\psi(\vec{\alpha})\rangle \delta\alpha_j\delta\alpha_k^*. \tag{C2}$$

Now in the same manner as QAS, using the Mclachlan's variational principle [24, 30, 31, 52], we demand that the variation of this fidelity is equal to 0 with respect to α_j:

$$\implies \langle\psi(\vec{\alpha})| U^\dagger(\Delta t) |\chi_j\rangle \langle\psi(\vec{\alpha})| U(\Delta t) |\psi(\vec{\alpha})\rangle + \sum_k \langle\psi(\vec{\alpha})|U^\dagger(\Delta t)|\chi_j\rangle \langle\chi_k|U(\Delta t)|\psi(\vec{\alpha})\rangle \delta\alpha_k^* = 0$$

$$\implies \langle\psi(\vec{\alpha})| U(\Delta t) |\psi(\vec{\alpha})\rangle + \sum_k \langle\chi_k|U(\Delta t)|\psi(\vec{\alpha})\rangle \delta\alpha_k^* = 0. \tag{C3}$$

Now we substitute in $U(\delta t) = I - i\Delta t H$:

$$\implies \langle\psi(\vec{\alpha})|\psi(\vec{\alpha})\rangle - i\Delta t \langle\psi(\vec{\alpha})| H |\psi(\vec{\alpha})\rangle + \sum_k \langle\chi_k|\psi(\vec{\alpha})\rangle \delta\alpha_k^* - i\Delta t \sum_k \langle\chi_k|H|\psi(\vec{\alpha})\rangle \delta\alpha_k^* = 0. \tag{C4}$$

Now we take the derivative of this equation with respect to Δt. Note that $\frac{d}{d\Delta t}\delta\alpha_k^* = \delta\dot{\alpha}_k^*$. We then discard any terms remaining that are linear in Δt or in $\delta\alpha$ (implying we have chosen such a small Δt that $\delta\alpha$ is also very small).

$$\implies -i \langle\psi(\vec{\alpha})| H |\psi(\vec{\alpha})\rangle + \sum_k \delta\dot{\alpha}_k^* \langle\chi_k|\psi(\vec{\alpha})\rangle \delta\alpha_k^* = 0. \tag{C5}$$

Using the above definition of the $\mathcal{E}$ and $\mathcal{D}$ matrices in equation 12 and 13, this simplifies to:

$$\implies -i\vec{\alpha}^\dagger \mathcal{D}\vec{\alpha} + \vec{\alpha}^\dagger \mathcal{E}\vec{\alpha} = 0$$

$$\implies \mathcal{E}\vec{\alpha} = -i\mathcal{D}\vec{\alpha}. \tag{C6}$$

This is exactly the same differential equation that we aim to solve in QAS. If we do not ignore the higher order terms, we could obtain systematic higher order corrections to the QAS matrix differential equation using such a method.

Appendix D: Unitary implementation

As alternative, we could implement the unitary evolution operator $U(\Delta t)$ directly instead of the Taylor series expansion of Eq.7

$$|\psi\left(\alpha\left(t+\Delta t\right)\right)\rangle_K = U\left(\Delta t\right)|\psi\left(\alpha\left(t\right)\right)\rangle_K. \tag{D1}$$

and defining the matrix $\mathcal{R}_{m,n} = \langle\chi_m|U(\Delta t)|\chi_n\rangle$ to solve the program

$$\max_{\alpha'}\ \alpha'^\dagger\mathcal{R}\alpha\alpha^\dagger\mathcal{R}^\dagger\alpha' \tag{D2}$$

$$\text{s.t}\ \alpha'^\dagger\mathcal{E}\alpha' = 1. \tag{D3}$$

$U(\Delta t)$ could be implemented with a Trotter decomposition or with an oracle. However, this complicates the circuits needed to calculate the $\mathcal{R}$ matrix, requiring the usage of Hadamard tests.

Appendix E: Higher order approximations

We investigate higher order expansion for the evolution operator in this section. First, we define the overlap matrix $\mathcal{J}$

$$\mathcal{J}_{m,n} = \sum_{i,j}\beta_i\beta_j\langle\chi_m|P_iP_j|\chi_n\rangle. \tag{E1}$$

Considering the next highest power expansion of $U(\Delta t)$:

$$U(\Delta t) \approx I - \iota\Delta t\left(\sum_{j=1}^{r}\beta_jP_j\right) - \frac{\Delta t^2}{2}\left(\sum_{j=1}^{r}\beta_jP_j\right)^2 \equiv V_2\left(\Delta t\right), \tag{E2}$$

and defining $|\phi\rangle = V_2(\Delta t)|\psi(\alpha)\rangle_K$, the constraint in the optimization program 11 turns out to be still the same as equation 14:

$$\langle\psi(\alpha'^\dagger)|\psi(\alpha'^\dagger)\rangle = \alpha'^\dagger\mathcal{E}\alpha'. \tag{E3}$$

It turns out that $\langle\phi|\phi\rangle$ is actually exactly equal to $\alpha^\dagger E\alpha$, which is the result we used earlier in equation 15, as all the 2nd order terms nicely cancel out.

Now, using the notation $G_2 \equiv \left(\mathcal{E} - \iota\Delta t\mathcal{D} - \frac{\Delta t^2}{2}\mathcal{J}\right)$,

$$\langle\psi\left(\alpha'\right)|\phi\rangle_K\langle\phi|\psi\left(\alpha'\right)\rangle_K = \alpha'^\dagger G_2\alpha\alpha^\dagger G_2^\dagger\alpha'. \tag{E4}$$

Now the optimization program in 11 can be re-expression in this higher order approximation as

$$\max_{\alpha'}\ \alpha'^\dagger\left(\frac{G_2\alpha\alpha^\dagger G_2^\dagger}{\alpha^\dagger\mathcal{E}\alpha}\right)\alpha' \tag{E5}$$

$$\text{s.t}\ \alpha'^\dagger\mathcal{E}\alpha' = 1.$$

And using the notation $W_{2,\alpha} \equiv \frac{G_2\alpha\alpha^\dagger G_2^\dagger}{\alpha'\mathcal{E}\alpha}$, we further condense the above optimization program as

$$\max_{\alpha'}\ \alpha'^\dagger W_{2,\alpha}\alpha' \tag{E6}$$

$$\text{s.t}\ \alpha'^\dagger\mathcal{E}\alpha' = 1. \tag{E7}$$

Once again, the only work that the quantum computer need to do is to calculate overlap matrices in the start, in this case having to calculate $\mathcal{E}$, $\mathcal{D}$ and $\mathcal{J}$. In fact, when going from lower order approximations to higher order approximations, you can reuse the saved matrices and only calculate the new ones needed. In this case, in the original TQS, which uses a first order approximation for $U(\Delta t)$, we already have the $\mathcal{E}$ and $\mathcal{D}$ matrices, so if we deem the results not up to our desired accuracy, we can easily go to the second order approximation showed here, and only require calculation of one additional matrix $\mathcal{J}$.