2018 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
Modeling & Simulation, Testing and Validation (MSTV) Technical Session
August 7-9, 2018 – Novi, Michigan

QUANTUM ANNEALING FOR MOBILITY STUDIES: GO/NO-GO MAPS VIA QUANTUM-ASSISTED MACHINE LEARNING

Radu Serban
University of Wisconsin – Madison
Madison, WI

Max Wilson, Marcello Benedetti, John Realpe-Gómez, Alejandro Perdomo-Ortiz, Andre Petukhov
NASA Ames Research Center
Moffett Field, CA

Paramsothy Jayakumar
U.S. Army RDECOM-TARDEC
Warren, MI

ABSTRACT

We present the results of an exploratory investigation of applying a hybrid quantum-classical architecture to an off-road vehicle mobility problem, namely the generation of GO/NO-GO maps posed as a machine learning problem.

The premise of this work rests on two observations. First, quantum computing allows in principle for algorithms that provide a speedup over the best known classical counterparts. However, as it is to be expected of such novel and complex tools (both hardware and algorithmic) at this early developmental stage, current quantum algorithms do not always perform well on real-world problems. Second, complex physics-based vehicle and terramechanics models and simulations, currently advocated for high-fidelity high-accuracy ground vehicle–terrain interaction analyses, pose significant computational burden, especially when applied to mobility studies which may require numerous simulation runs.

We describe the Quantum-Assisted Helmholtz Machine formulation, suitable to be implemented on a quantum annealer such as the D-Wave 2000Q machine, discuss the high-performance classical computing framework used to generate through simulation the training and test sets, and provide the results of our investigations and analysis into the performance of the machine learning model and its predictive capabilities for generating GO/NO-GO mobility maps.

This work represents a contribution to an ongoing effort of exploring the applicability of the emerging field of quantum computing to challenging engineering and scientific problems.

1 INTRODUCTION

The work presented in this paper is anchored in two main observations. First, quantum computing allows for algorithms that provide a speedup over the best known classical algorithms for the same task. However, current implementations of quantum algorithms do not always perform well on real-world-scale problems, as is to be expected of novel and complex tools at such an early stage of development. Demonstrating the implementation and applicability of quantum-assisted algorithms on problems of engineering and scientific interest and scale serves two purposes: (i) guiding the development of the hardware; and (ii) establishing areas of application for future devices. While predictions from various experts vary, it is expected that over the next decade quantum computers will be at a stage where hybrid architectures (classical algorithms with quantum-accelerated subroutines) will be useful in a variety of real-world problems. Machine Learning (ML) is a particular area in computational science that is likely to benefit from these developments.

The second remark relates to high-fidelity modeling and simulation of ground vehicle – terrain interaction. Multibody system dynamics is nowadays a mature and well-established field, widely adopted for predictive simulations of mechanical systems in general and ground vehicles on hard surface in particular. Simultaneously, terrain and soil mechanics is at a stage where complex vehicle–terrain interactions can be resolved at high-fidelity and increasing accuracy. This resulted in a renewed push to adopt these simulation techniques and include them in operational tools for assessing mobility of off-road ground vehicles. As an example, we mention the efforts for the development of an updated NATO Reference Mobility Model (NRMM) [1]; namely the Next Generation NRMM (NG-NRMM), which will rely on full 3-D multibody vehicle models and physics-based terramechanics models, while drawing on more powerful and varied computer architectures [2].

The goal of this work was therefore an exploratory investigation on the applicability of quantum-assisted computing to the generation of GO/NO-GO maps, cast as a machine learning problem. Concretely, we investigate the applicability of Quantum Machine Learning (QML), using an algorithm possible to be implemented on a particular instance of quantum computing hardware (quantum annealing as implemented on the D-Wave 2000Q platform [3]), to a problem of significant engineering and operational interest: off-road ground vehicle mobility. The definition of the particular problem considered here and the overall solution approach is delineated next. The ML approach, the simulation infrastructure, and results are described in the subsequent sections, with further details available in [4].

While there is no general agreement on the exact definition of ground vehicle mobility, the consensus is that the maximal *speed-made-good* is a good measure for mobility [1]. Herein, speed-made-good is defined as the ratio of the straight-line distance between two points and the time required to travel between them, regardless of the actual path. This measure of mobility depends in a highly-nonlinear fashion on many parameters, among which we mention: terrain topology, profile, and roughness; soil type, properties, and moisture content; meteorological conditions; and, finally, vehicle type, configuration, and characteristics.

Under the definition adopted here, a GO/NO-GO map can thus be obtained by color-coding a particular geographical area as function of the corresponding mobility measure (either a continuous variable such as the speed-made-good or a binary output indicating whether or not the destination point can be reached). Since terrain variability and heterogeneity can be very large even for relatively confined geographical regions, a pragmatic approach consists of discretizing the area in a grid,

not necessarily uniform, such that input parameters (e.g., terrain characteristics and soil properties) can be considered constant within a grid cell. Then, the GO/NO-GO map is obtained by estimating a mobility measure for each cell and appropriately color-coding the grid.

Generation of such a GO/NO-GO map can be based on empirical data or on simulation results of varying fidelity. On the one hand, empirical data alone will lack accuracy and predictive power. On the other hand, a simulation-based approach can be prohibitively intensive in terms of computational cost depending on the level of fidelity adopted for modeling soil and/or vehicles, rendering this approach unfeasible for operational purposes.

The machine learning aspect. A solution to the potentially prohibitive computational cost of mobility analysis based on simulations involving high-fidelity vehicle multibody systems and complex terramechanics is framing the generation of GO/NO-GO maps as a machine learning problem. This can take the form of either a classification problem; i.e., the problem of deciding to which label – GO or NO-GO – a given set of observations (input data; e.g., terrain profile, soil properties, vehicle type) belongs, or even more generally, as an unsupervised learning problem, where the goal is to infer/learn correlations among all variables involved, including both input data and labels.

The quantum computing aspect. Quantum computers are designed to manipulate vectors and tensor products in high-dimensional spaces. As a consequence, algorithms for QML can potentially provide exponential speed-ups for a wide range of machine learning tasks [5]. Much of the record-breaking performance of classical machine-learning algorithms regularly reported in the literature pertains to task-specific supervised learning algorithms. On the other hand, unsupervised learning algorithms are more general and more 'human-like', but their development has been lagging behind due to their intractability. This intractability may come from the combinatorial nature of the problem (e.g. in clustering), or it may come from the need to sample complex probability distributions (e.g. the Boltzmann distribution). Quantum annealing holds the potential to sample more efficiently than classical Markov Chain Monte Carlo methods, and this could in turn significantly advance the field of unsupervised learning.

The hybrid computing aspect. The overall methodology envisioned and adopted for exploration herein was therefore a hybrid approach in which quantum computing was used to train a machine learning model using classical computing-based simulation to generate the training and test data.

This paper is organized as follows. In Section 2 we provide an overview of the machine learning model used in this study, namely the Quantum-Assisted Helmholtz Machine. Section 3 provides a brief description of the vehicle, terrain, and vehicle-ground interaction models and simulation setup. Various algorithmic investigations, focusing on the characteristics of the problem at hand and their impact on model performance and predictive capabilities are discussed in section 4. We conclude the paper with final remarks and suggestions for future work in section 5.

2 QUANTUM-ASSISTED MACHINE LEARNING

The Quantum-Assisted Helmholtz Machine (QAHM) [6] is a concrete proposal that can exploit the sampling power of quantum annealing to learn a complex probability distribution over continuous variables.

2.1 Model and learning algorithm

Consider a dataset $\mathcal{S} = \{\mathbf{v}^1, \ldots, \mathbf{v}^d\}$ with empirical distribution $Q_{\mathcal{S}}(\mathbf{v})$. We seek a generative model $P(\mathbf{v}) = \sum_{\mathbf{u}} P(\mathbf{u}, \mathbf{v})$, where $P(\mathbf{u}, \mathbf{v}) = P(\mathbf{v}|\mathbf{u})P_{QC}(\mathbf{u})$. The prior distribution $P_{QC}(\mathbf{u}) = \langle \mathbf{u}|\rho|\mathbf{u}\rangle$ describes samples obtained from a quantum device, and we assume it can be described by a quantum Gibbs distribution $\rho = e^{-\mathcal{H}}/\mathcal{Z}$, where $\mathcal{H}$ is the Hamiltonian implemented in quantum hardware and $\mathcal{Z}$ is the partition function. In the particular case of quantum annealing hardware, we have

$$\mathcal{H} = \sum_{i<j} J_{ij}\hat{Z}_i\hat{Z}_j + \sum_i h_i\hat{Z}_i + \Gamma\sum_i \hat{X}_i, \qquad (1)$$

where $\hat{Z}_i$ and $\hat{X}_i$ denote Pauli matrices in the z and x direction, respectively, while J_{ij}, h_i, and Γ are controllable parameters. The conditional distribution $P(\mathbf{v}|\mathbf{u})$ stochastically translates samples from the quantum computer into samples on the domain of the data. Because $\mathbf{v}$ is sampled indirectly from hardware, it can be any type of data. In our case, it corresponds to a vector of continuous variables.

Ideally, an unsupervised learning algorithm for this generative model would maximize the average log-likelihood of the data $\mathcal{L} = \sum_{\mathbf{v}} Q_{\mathcal{S}}(\mathbf{v}) \ln P(\mathbf{v})$, with respect to the controllable parameters. Unfortunately this quantity is intractable because it requires computation of the posterior $P(\mathbf{u}|\mathbf{v})$ for all possible $\mathbf{u}$. The Helmholtz machine overcomes this by introducing a lower bound

$$\sum_{\mathbf{v}} Q_{\mathcal{S}}(\mathbf{v}) \ln P(\mathbf{v}) \geq \sum_{\mathbf{v},\mathbf{u}} Q_{\mathcal{S}}(\mathbf{v})Q(\mathbf{u}|\mathbf{v}) \ln \frac{P(\mathbf{v}|\mathbf{u})P_{QC}(\mathbf{u})}{Q(\mathbf{u}|\mathbf{v})}, \qquad (2)$$

where $Q(\mathbf{u}|\mathbf{v})$ is an auxiliary model that approximates the intractable true posterior $P(\mathbf{u}|\mathbf{v})$. In practice, the newly introduced $Q(\mathbf{u}|\mathbf{v})$ and the conditional $P(\mathbf{v}|\mathbf{u})$ are implemented by classical artificial neural networks. From now on, we call probability distribution P the *generator network*, and probability distribution Q the *recognition network*.

Note that $P_{QC}(\mathbf{u}) = \ln\langle \mathbf{u}|\rho|\mathbf{u}\rangle$ is intractable due to the projection of the Gibbs distribution on the states $|\mathbf{u}\rangle$. However, this term can be further bounded as

$$\ln\langle \mathbf{u}|\rho|\mathbf{u}\rangle \geq \langle \mathbf{u}|\ln\rho|\mathbf{u}.\rangle \qquad (3)$$

Combining Eqs. (2) and (3), we get a tractable lower bound for the generator to maximize

$$\mathcal{G}(\theta_G, \theta_{QC}) = \sum_{\mathbf{v}} Q_{\mathcal{S}}(\mathbf{v}) \sum_{\mathbf{u}} Q(\mathbf{u}|\mathbf{v}) \left[\ln P(\mathbf{v}|\mathbf{u}) + \langle \mathbf{u}|\ln\rho|\mathbf{u}\rangle\right], \qquad (4)$$

where θ_G and θ_{QC} denote the parameters of generator network P and quantum state ρ, respectively, and where. In Eq. (4) we neglected terms that do *not* depend on either θ_G or θ_{QC}, as they vanish when computing the gradient of $\mathcal{G}$.

Now we consider the recognition network $Q(\mathbf{u}|\mathbf{v})$, which has to closely track the true posterior during learning. The maximization of Eq. (2) with respect to the parameters of the recognition network is also intractable. Reference [7] introduced the *wake-sleep* algorithm which minimizes the more tractable Kullback-Leibler divergence

$$D_{KL}\left[P(\mathbf{u}|\mathbf{v})||Q(\mathbf{u}|\mathbf{v})\right] = \sum_{\mathbf{u}} P(\mathbf{u}|\mathbf{v}) \ln \frac{P(\mathbf{u}|\mathbf{v})}{Q(\mathbf{u}|\mathbf{v})},$$

averaged over the marginal $P(\mathbf{v})$ to take into account the relevance of each configuration $\mathbf{v}$. In other words, the reconstruction network maximizes function

$$\mathcal{R}(\theta_R) = \sum_{\mathbf{u}} P_{QC}(\mathbf{u}) \sum_{\mathbf{v}} P(\mathbf{v}|\mathbf{u}) \ln Q(\mathbf{u}|\mathbf{v}), \qquad (5)$$

where θ_R denotes, collectively, the parameters of the recognition network Q. In Eq. (5) we neglected

terms that do not depend on θ_R, as they vanish when computing the gradient of $\mathcal{R}$.

The gradient ascent equations have structure $\theta^{(t+1)} = (1-\eta\lambda)\theta^{(t)} + \eta\nabla_\theta \mathcal{F}$, where θ stands for the parameters being updated, η is the learning rate, λ is a regularization factor, and $\mathcal{F}$ stands for either $\mathcal{G}$ or $\mathcal{R}$, accordingly. The type of regularization used here is called weight decay and helps in preventing overfitting by keeping the weights small.

Since $\ln\rho = -\mathcal{H} - \ln\mathcal{Z}$, for the parameters of the quantum distribution $\theta_{QC} = (J_{ij}, h_i)$ we have

$$-\frac{\partial \mathcal{G}}{\partial J_{ij}} = \langle u_i u_j\rangle_Q - \langle u_i u_j\rangle_\rho, \quad (6)$$

$$-\frac{\partial \mathcal{G}}{\partial h_i} = \langle u_i\rangle_Q - \langle u_i\rangle_\rho, \quad (7)$$

where $\langle\,\rangle_Q$ and $\langle\,\rangle_\rho$ denote expectation values with respect to $Q(\mathbf{u}|\mathbf{v})Q_{\mathcal{S}}(\mathbf{v})$ and $P_{QC}(\mathbf{u}) = \langle\mathbf{u}|\rho|\mathbf{u}\rangle$, respectively. Here we have used the property $\hat{Z}_i|u_i\rangle = u_i|u_i\rangle$.

The generation and recognition networks can be written as deep learning architectures

$$P(\mathbf{v}|\mathbf{u}) = \sum_{\mathbf{u}^1,\ldots,\mathbf{u}^L} P_0(\mathbf{v}|\mathbf{u}^1)P_1(\mathbf{u}^1|\mathbf{u}^2)\cdots P_L(\mathbf{u}^L|\mathbf{u})$$

$$Q(\mathbf{u}|\mathbf{v}) = \sum_{\mathbf{u}^1,\ldots,\mathbf{u}^L} Q_L(\mathbf{u}|\mathbf{u}^L)\cdots Q_1(\mathbf{u}^2|\mathbf{u}^1)Q_0(\mathbf{u}^1|\mathbf{v}),$$

in terms of L additional sets of hidden variables $\mathbf{u}^1,\ldots,\mathbf{u}^L$ that connect the variables $\mathbf{v}\equiv\mathbf{u}^0$ in the visible layer with $\mathbf{u}\equiv\mathbf{u}^{L+1}$ in the last hidden layer. More specifically, when using Bernoulli variables $u_i^\ell \in \{-1,+1\}$, we have

$$P_\ell(\mathbf{u}^\ell|\mathbf{u}^{\ell+1}) = \prod_i \pi(u_i^\ell|\mathbf{u}^{\ell+1}; A^\ell, a^\ell), \quad (8)$$

$$Q_\ell(\mathbf{u}^\ell|\mathbf{u}^{\ell-1}) = \prod_i \pi(u_i^\ell|\mathbf{u}^{\ell-1}; B^\ell, b^\ell), \quad (9)$$

where $\pi(u_i|\mathbf{u}'; C, c) = \left[1 + e^{-2u_i\left(\sum_j C_{ij}u'_j + c_i\right)}\right]^{-1}$.

The gradients for the generative network are

$$\frac{\partial \mathcal{G}}{\partial A_{ij}^\ell} = \langle u_i^\ell u_j^{\ell+1}\rangle_Q - \langle u_i^\ell\rangle_P\langle u_j^{\ell+1}\rangle_Q,$$

$$\frac{\partial \mathcal{G}}{\partial a_i^\ell} = \langle u_i^\ell\rangle_Q - \langle u_i^\ell\rangle_P,$$

and similarly for the recognition network

$$\frac{\partial \mathcal{R}}{\partial B_{ij}^\ell} = \langle u_i^\ell u_j^{\ell-1}\rangle_P - \langle u_i^\ell\rangle_Q\langle u_j^{\ell-1}\rangle_P,$$

$$\frac{\partial \mathcal{R}}{\partial b_i^\ell} = \langle u_i^\ell\rangle_P - \langle u_i^\ell\rangle_Q.$$

2.2 Implementation

We now discuss our implementation of the QAHM for the D-Wave 2000Q quantum annealer. The annealer implements a noisy version of the programmed Hamiltonian in Eq. (1) defined on a sparse graph of qubit interactions. The device samples low energy states in the limit $\Gamma \to 0$, nontrivial non-equilibrium effects may make samples deviate from the corresponding classical Gibbs distribution. Several features of the algorithm are engineered to cope with these effects.

To overcome this we follow the work in [8]. By treating the annealer as grey-box, this approach allows updating the parameters without the need to estimate deviations from the Gibbs distribution. It also allows us to implement a fully connected prior distribution $P_{QC}(\mathbf{u})$, despite the actual physical connectivity be sparse.

For implementing continuous variables, when using the recognition network, each variable v_i in the dataset is rescaled to lie in $[-1,+1]$ and interpreted as the expected value of a binary variable in $\{-1,+1\}$, i.e. the average value of that variable if evaluated many times. Conversely, the probabilities in output from the generator network are used to compute the expected value of a binary variable, rather than to actually sample the binary variable.

Such a value can then be scaled back to the original range of the continuous variable. These techniques allow to use the model without further modifications and without the need of additional parameters.

Further details of the physical implementation can be found in [4].

3 SIMULATION FOR MOBILITY ASSESSMENT

Generation of the training and test sets were performed using the *Chrono* software suite [9]. *Chrono*'s strength lies in its ability to simulate the dynamics of large multibody systems [10], including discipline-specific support for vehicle modeling and simulation and soil/terrain modeling and simulation. In particular, deformable terrains can be simulated using a fully-resolved Discrete Element Method (DEM) approach, by modeling the soil as a large system of bodies interacting through contact, friction, and cohesion (albeit, not necessarily at the physical soil particle size). Alternatively, *Chrono* provides support for more expeditious deformable soil representation, such as the Soil Contact Model (SCM) [11].

The *Chrono*::Vehicle module provides support for template-based modeling and simulation of ground vehicles, both wheeled and tracked, that can be fully and implicitly coupled with the terrain/soil models mentioned above. As such, *Chrono* can be used to generate the input data sets required for the learning algorithms considered here: for a given set of parameters, a coupled vehicle-terrain interaction simulation is conducted and results processed to obtain the desired mobility measure (speed-made-good or simply a binary GO/NO-GO decision variable).

3.1 Vehicle modeling

Chrono::Vehicle [12] is a module of the open-source multi-physics simulation package *Chrono*, aimed at modeling, simulation, and visualization of wheeled and tracked ground vehicle multi-body systems. Its software architecture and design was dictated by the desire to provide an expeditious and user-friendly mechanism for assembling complex vehicle models, while leveraging the underlying *Chrono* modeling and simulation capabilities, allowing seamless interfacing to other optional *Chrono* modules (e.g., its granular dynamics and fluid-solid interaction capabilities), and providing a modular and expressive API to facilitate its use in third-party applications. Vehicle models are specified as a hierarchy of subsystems, each of which is an instantiation of a predefined subsystem template. Written in C++, *Chrono*::Vehicle is offered as a middleware library.

Chrono::Vehicle provides a comprehensive set of vehicle subsystem templates (tire, suspension, steering mechanism, driveline, sprocket, track shoe, etc.), templates for external systems (powertrain, driver, terrain), and additional utility classes and functions for vehicle visualization, monitoring, and collection of simulation results. Three different classes of tire models are supported: rigid (modeled as cylindrical shapes or else as non-deformable triangular meshes), semi-empirical (Pacejka, Fiala, Lugre, and TMeasy), and finite element (based on ANCF or Reissner shell elements). For additional flexibility and to allow integration of third-party software, *Chrono*::Vehicle is designed to permit either monolithic simulations or co-simulation where the vehicle, powertrain, tires, driver, and terrain/soil can be simulated independently and simultaneously.

For the studies conducted herein, we used a model of a four-wheel drive off-road vehicle with independent double-wishbone suspension and Pitman arm steering (see Fig. 1).

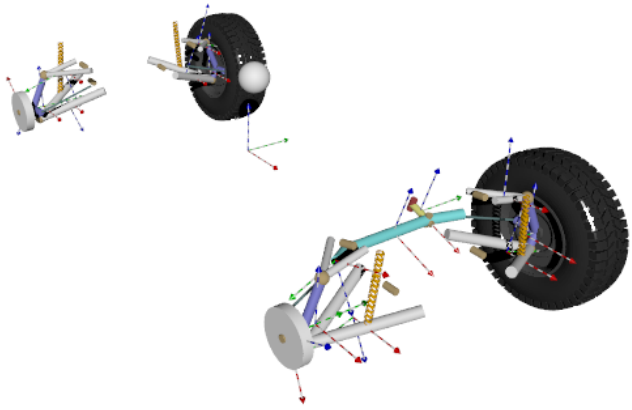

Figure 1: Wheeled vehicle with double wishbone suspensions and Pitman arm steering.

3.2 Granular terrain modeling

Chrono::Vehicle provides several classes of terrain and soil models, of different fidelity and computational complexity, ranging from rigid, to semi-empirical Bekker-Wong type models, to complex physics-based models based on either a granular or finite-element based soil representation. For simple terramechanics simulations, *Chrono*::Vehicle provides a customized implementation of the Soil Contact Model, based on Bekker theory, with extensions to allow non-structured triangular grids and adaptive mesh refinement. Second, *Chrono* provides an FEA continuum soil model based on multiplicative plasticity theory with Drucker-Prager failure criterion and a specialized 9-node brick element which alleviates locking issues with standard brick elements. Finally, leveraging the *Chrono*::Granular module and support for multi-core and distributed parallel computing in *Chrono*, off-road vehicle simulations can be conducted using fully-resolved, granular dynamics-based complex terramechanics, using a Discrete Element Method (DEM) approach. Such simulations can use either of the two methods supported in *Chrono*, namely a penalty-based, compliant-body approach, or a complementarity-based, rigid-body approach.

Two alternative approaches have emerged as viable solutions for large frictional contact problems in granular flow dynamics and quasi-static geomechanics applications, collectively termed herein DEM. The so-called complementarity method (DEM-C) is generally favored within the multi-body dynamics community. In this approach, individual particles in a bulk granular material are modeled as rigid bodies, and non-penetration conditions are written as complementarity equations which, in conjunction with a Coulomb friction law, lead to a Differential Variational Inequality (DVI) form of the Newton-Euler equations of motion. Not limited by stability considerations, DEM-C allows for much larger time integration steps than the alternative penalty-based (DEM-P) solutions, since the latter involve large contact stiffnesses that impose strict stability conditions on all explicit time integration algorithms. However, DEM-C involves a relatively complex and computationally costly solution sequence per time step, since it leads to a mathematical program with complementarity and equality constraints, which must be relaxed to obtain tractable linear complementarity or cone complementarity problems. More mature and widely adopted within the geomechanics community [13], DEM-P can be viewed either as a regularization (or smoothing) approach, which relies on a relaxation of the rigid-body assumption, or as a deformable-body approach localized to the points of contact between individual particles in a bulk granular material [14]. In this approach, normal and tangential contact forces are calculated using various laws [15], which are based on the local body deformation at the point of contact.

A more in depth comparison between the rigid-body (DEM-C) and soft-body (DEM-P) formulations is provided in [16]. All granular terrain simulations used herein relied on the DEM-C formulation. For details on the DEM-C formulation, derivation of the DVI problem, and *Chrono* imple-

mentation, the reader is directed to [17, 18, 10, 4]. Here, we only provide the resulting mixed differential algebraic–differential variational inequality problem in Eq. (10).

$$\dot{\mathbf{q}} = \mathbf{L}(\mathbf{q})\mathbf{v} \tag{10a}$$

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{v}} = \mathbf{f}\left(t, \mathbf{q}, \mathbf{v}\right) - \mathbf{g}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}, t)\hat{\lambda} \quad + \sum_{i \in \mathcal{A}(\mathbf{q}, \delta)} \underbrace{\left(\widehat{\gamma}_{i,n}\, \mathbf{D}_{i,n} + \widehat{\gamma}_{i,u}\, \mathbf{D}_{i,u} + \widehat{\gamma}_{i,w}\, \mathbf{D}_{i,w}\right)}_{i^{th}\text{frictional contact force}} \tag{10b}$$

$$\mathbf{0} = \mathbf{g}(\mathbf{q}, t) \tag{10c}$$

$$i \in \mathcal{A}(\mathbf{q}(t)) : \begin{cases} 0 \leq \Phi_i(\mathbf{q}) \;\perp\; \widehat{\gamma}_{i,n} \geq 0 \\ (\widehat{\gamma}_{i,u}, \widehat{\gamma}_{i,w}) = \underset{\sqrt{\bar{\gamma}_{i,u}^2 + \bar{\gamma}_{i,w}^2} \leq \mu_i \widehat{\gamma}_{i,n}}{\operatorname{argmin}} \quad \mathbf{v}^T \cdot \left(\bar{\gamma}_{i,u}\, \mathbf{D}_{i,u} + \bar{\gamma}_{i,w}\, \mathbf{D}_{i,w}\right) \;. \end{cases} \tag{10d}$$

These represent the equations of motion for a multibody system involving both bilateral (equality) constraints and unilateral (frictional contact) constraints. The differential equations in Eq. (10a) relate the time derivative of the generalized positions $\mathbf{q}$ and velocities $\mathbf{v}$ through a linear transformation defined by $\mathbf{L}(\mathbf{q})$. The force balance in Eq. (10b) ties the inertial forces to the applied and constraint forces, $\mathbf{f}\left(t, \mathbf{q}, \mathbf{v}\right)$ and $-\mathbf{g}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}, t)\hat{\lambda}$, respectively. The latter are impressed by the bilateral constraints of Eq. (10c) that restrict the relative motion of the rigid or flexible bodies present in the system. The non-penetration conditions in Eq. (10d) express the complementarity between the separation function Φ_i for contact i and the associated normal contact force. The last equation poses an optimization problem whose first order Karush-Kuhn-Tucker optimality conditions are equivalent to the Coulomb dry friction model. The frictional contact force associated with contact i leads to a set of generalized forces, shown in red in Eq. (10b), which are obtained using the projectors $\mathbf{D}_{i,n}$, $\mathbf{D}_{i,u}$, and $\mathbf{D}_{i,w}$ [18].

The problem in Eq. (10) is discretized to yield a mathematical program with complementarity and equality constraints. A relaxation of the complementarity conditions and further algebraic manipulations [17] yield a cone complementarity problem (CCP) whose solution provides the Lagrange multipliers λ and γ.

3.3 Computational and simulation setup

All simulations were conducted on a Cray XC30 system with 12-core Intel® Xeon® E5-2697 v2 processors and a dedicated Cray Aries high-speed network. Independent runs were launched in batches corresponding to the subsets of training points requested by the sequential approach described previously. Each separate run (corresponding to a particular set of the five independent design parameters considered, namely longitudinal terrain slope, particle radius, particle density, inter-particle coefficient of friction, and cohesion pressure) was queued on a single node and used 24 OpenMP threads.

To accelerate time to solution, simulations conducted for this study employed the moving patch feature provided within *Chrono*::Vehicle. In this approach, granular material is simulated only within a sliding window (of user-defined dimensions) centered around and moving with the ve-

hicle. The number of granular material particles remain constant (and relatively low). Particles falling outside the moving patch behind the vehicle are reused by relocating them in front of the vehicle in chunks of spatial dimensions controlled by the user. With this approach, depending on the current particle size, the simulations used herein employed between 48,000 and 480,000 particles.

Each simulation run represented a straight-line acceleration maneuver on longitudinally inclined terrain. The granular material was considered homogeneous and consisting of identical spheres. Particles were initialized in layers (with a number of layers determined dynamically, function of the particle radius) and using a uniform random distribution within each layer obtained with a Poisson disk sampling technique. Following a short granular material settling phase, the vehicle is created above the terrain and allowed to settle. Subsequently, throttle is increased from 0% to 100% over the span of 0.5 s, while the gravitational acceleration vector is rotated by the appropriate angle to model the incline plane. To maintain a straight-line, a path-follower steering controller is used which makes minute adjustments to the vehicle steering input. Several heuristics are implemented to decide completion of the simulated maneuver; tracking running averages of the vehicle forward velocity and acceleration, simulations are stopped when a steady-state maximum velocity is achieved or when the case of the vehicle sliding backward is identified.

During simulation, we record relevant vehicle states for each individual run. In a post-processing stage, information from these output files is collated to generate the incremental training set for the sequential QML algorithm, as well as statistics for estimating computational performance. Additional information pertaining to computational performance is provided in [4].

4 ALGORITHMIC INVESTIGATIONS

The hybrid quantum-classical computational architecture, described in Sections 2 and 3 and illustrated in Fig. 2, was used for generating and learning the distribution of the simulation data. We describe here the implementation of the QAHM on datasets provided through simulations and examine relevant characteristics of both the dataset and the model. We highlight and discuss two salient features of the model and draw a strong link between these features and the model effectiveness, using as a performance metric the mean squared error between the predictions of the model and an evaluation (test) set.

The training and test sets contain 320 and 40 points, respectively, sampled from the 6-D parameter space (5 inputs and 1 output). The majority of the networks used for these demonstrations consist of 2 hidden layers comprising 12 nodes each, and a visible 6-node layer. Several experiments were needed in order to choose the hyperparameters. A learning rate $\eta = 0.01$ and a regularization factor $\lambda = 0.0001$ were found to produce best results. The ranges used for the normalization of the input parameters were slope = [0.0, 20.0], radius = [8.0, 18.0], density = [1000.0, 2000.0], friction = [0.6, 1.0], cohesion = [500.0, 1500.0] and velocity = [-5.0, 40.0].

4.1 Training and test data

There are three key features of the data which are important to these investigations. First, the size of the initial training set is 320 points; although neural network machine learning algorithms are notoriously data intensive, expanding the dataset available was limited by the computational resource required. Second, the points in the training set and the test set were both distributed regularly, using Latin Hypercube Sampling (LHS), over the 5-D input parameter space. Lastly, the distribution of the velocities is roughly bimodal. As shown in

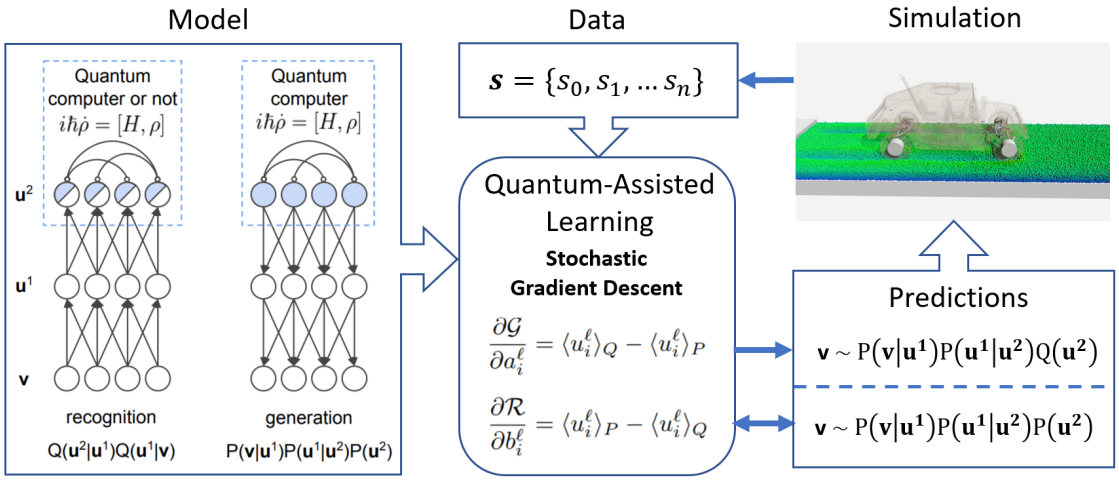

Figure 2: Pipeline for learning simulation datasets with quantum assisted models.

Fig. 3, the final velocity of the simulation tends to fall into one of two modes, high ($20 \ldots 34$ m/s) or low ($-5 \ldots 5$ m/s), and is skewed to higher velocities; this can be explained in part by the LHS sampling used to generate the training set.

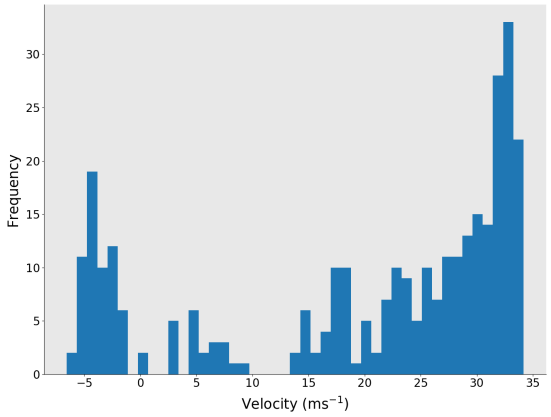

Figure 3: Distribution of final velocities in the training set (simulation results), showing its skew and bimodal nature.

In classical machine learning these characteristics of the training data can be addressed and accommodated by selecting tailored algorithms. However, at this early stage of quantum machine learning development, only few viable algorithmic options are available. As such, the dataset size limits the ability of the algorithm to effectively train models with larger number of nodes (and hence the number of parameters to be updated during training); this situation can result in underfitting.

Furthermore, sampling the test and training sets using the same method resulted in undesired correlations between the predictions on the two sets. However, LHS was a necessary compromise between the need for random sampling and the ability to only provide a relatively small dataset through simulation; indeed, random sampling with a small number of points could have resulted in small sample size bias.

4.2 Model fitting

Initial investigation into optimizing the model performance, primarily by varying the number of nodes in an attempt to correctly fit the model, indicated that the network complexity (number of hidden nodes) was not the main source of the observed systematic errors. As shown in Fig. 4, the model performance – as measured by the mean squared error between predictions and data – improves with increased network complexity up to

about 10 nodes per hidden layer, but then reaches a noise floor around 60 $(\mathrm{m/s})^2$. This is observed for both the training and test sets, whereas typically one expects a monotonic evolution as the model tends to overfitting.

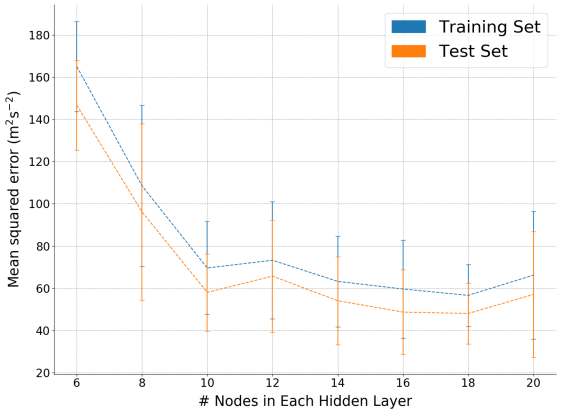

Figure 4: Prediction accuracy as function of network complexity (number of nodes per hidden layer).

The source of these prediction errors can be identified by looking at their distribution across given dataset. Figure 5 shows the mean squared errors at the different velocities in the training set and highlights a *squeezing* phenomenon, whereas the extremities of the velocity range in the dataset are outside the predictive bounds of the model; in other words, the model tends to generate a range of predictions that is squeezed relative to the data in the training set.

A second consequential effect of the larger errors at the extremities of velocity range, coupled with the mostly bimodal distribution of velocities in the training set, is a tendency of predictions to switch between these two modes. This effect, termed here *mode-switching*, negatively impacts the accuracy of samples from the model and contributed dominant anomalous errors to the resulting predictions.

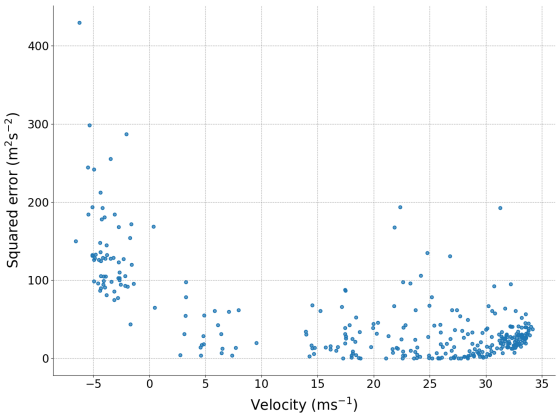

Figure 5: Distribution of the errors in predictions across the training set, showing clear dominant errors at lower velocities and upticks in error at both the lowest and highest velocities.

4.3 Model sampling

The model is sampled in order to make predictions about the velocity at a given point in the parameter space. In order to mitigate systematic errors discussed above, due to the *squeezing* and *mode-switching* phenomena, several sampling methods were investigated. these are succinctly listed here and described in more detail in [4].

Sampling 1: Markov chain sampling from the generative network. In this process, one step in the Markov chain evolution consists of a bottom–up pass of the recognition network (with given input parameters and randomized velocity), followed by a top–down pass of the generation network. While the system does reach an equilibrium, the variance of the produced distribution is so large that the results resemble noise; see Fig. 6.

Sampling 2: Markov chain sampling with averaging. Different from the previous case, at each step of the Markov chain, the model is sampled multiple times (e.g., 10 samples) and the results averaged; see Fig. 7. While mitigating the large variance observed with the pure Markov chain

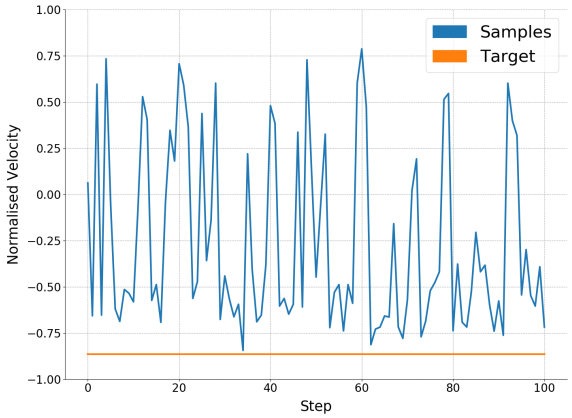

Figure 6: Markov chain evolution of the QAHM: samples from the device are used as measures of the predicted velocity.

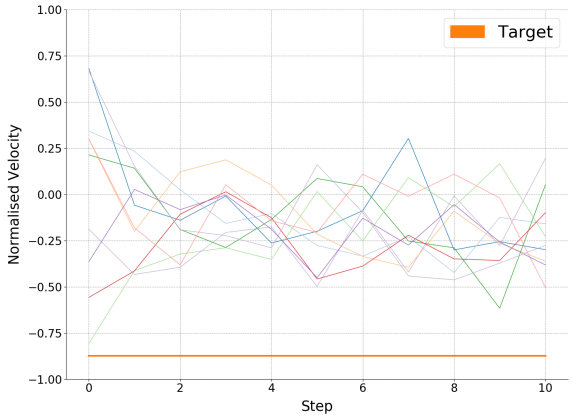

Figure 7: Markov chain sampling with averaging: at each step of the Markov chain several samples are taken and the average of these values used to calculate the value of the next step.

sampling above, this averaging has the disadvantage of further squeezing the results away from the extreme values.

Sampling 3: Peak frequency sampling. In this process, a number of samples (e.g.; 500) are taken at random velocities in the range given by the dataset, without evolving the system like in the previous sampling methods. A histogram is then generated (using 40–50 bins) and the prediction is set to be the velocity corresponding to peak frequency. This approach provides better results for predicting velocities from the lower and higher modes, but has difficulties in predicting intermediate values. Indeed, in the case illustrated in Fig. 8 the predicted normalized velocity is around -0.5, far from the training data, indicating bias toward the low mode.

4.4 Results

We implemented both a quantum-assisted and a classical versions of the QAHM model and use them to predict model output (final velocities) on a regular discretization of the 5-D input parameter space, generating a total of 10^5 predictions. In both cases a 6 x 12 x 12 deep neural network is trained on a 320 point dataset from the 6-D parameter space for 5000 epochs with learning rate $\eta = 0.01$ and regularization factor $\lambda = 0.0001$. We present here a comparison of the learning performance of the two versions of the algorithm and then discuss the predictions of the quantum-assisted generative model and their respective variations.

As demonstrated in Fig. 9, the quantum and classical models generate predictions of comparable accuracy (as measured against the training data). Furthermore, both sets of predictions have the same characteristics, namely non-uniform mean squared error distribution with a large range and squeezing, as explored in the algorithm investigations above.

The first conclusion drawn from the plots in Fig. 9 is that the quantum algorithm is successfully training a model. The second promising result is that the classical and quantum algorithms are comparable. Any slight difference can be attributed to the numerical differences incurred when translating between the two algorithms (e.g,

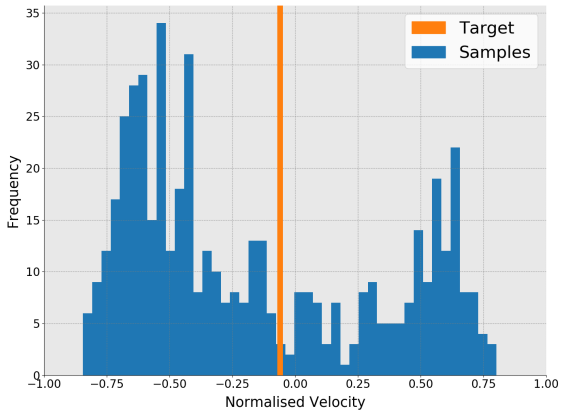

Figure 8: Peak frequency sampling: for a given input vector the QAHM is sampled repeatedly and the corresponding distribution generated.

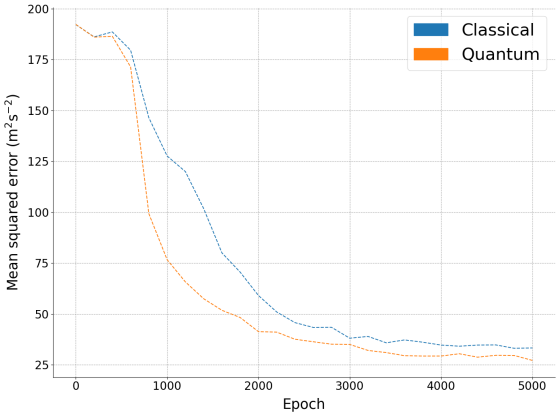

Figure 9: Comparison of the performance of the quantum and classical algorithms.

a learning rate of 1 in the classical algorithm may not equate to a learning rate of 1 for the quantum algorithm). These ideas are complementary and a first step in establishing the quantum-assisted algorithm and laying the groundwork for extensions, variations, and improvements.

Figure 10 confirms intuitive understanding of the data, thus providing evidence that the quantum-assisted model is learning the underlying distribution. The plot on the left represents a slice through the parameter space (for constant particle radius, material density, and inter-particle coefficient of friction) with final velocities obtained as model predictions at various combinations of slope and cohesion. The shape of the predicted surface across the considered range generally follows the expected trend. In particular, velocity is negatively correlated with slope and positively correlated with cohesion. The standard deviation of the predictions, provided on the right of Fig. 10, shows that the regions of high variance of the predictions tend to be in regions of steep change, namely between the two modes (large and small velocity), characterized by large gradient of the velocity.

4.5 Algorithm Investigation Discussion

The issues outlined in these investigations are *mode-switching* and *squeezing*. These were identified and described above, in conjunction with various attempts at mitigating their effect on prediction accuracy. Mode-switching results in dominant anomalous predictions, while squeezing leads to a small systematic error for the predictions at the extremities of the model.

Throughout the investigations, several sources of prediction squeezing have been highlighted. Although other sources may contribute to this effect, the evidence suggests that fundamental aspects of the training algorithm result in a model which squeezes predictions to some range smaller than that of the input data. It is possible to say with some confidence that mode-switching is the combined effect of the particular structure of the data considered here and some learning feature of the algorithm which results in samples with a high variance and with a bias toward regions trained with more data. Potential directions for devising algorithm adjustments to reduce variance and increase range of the resulting model are outlined in Section 5.

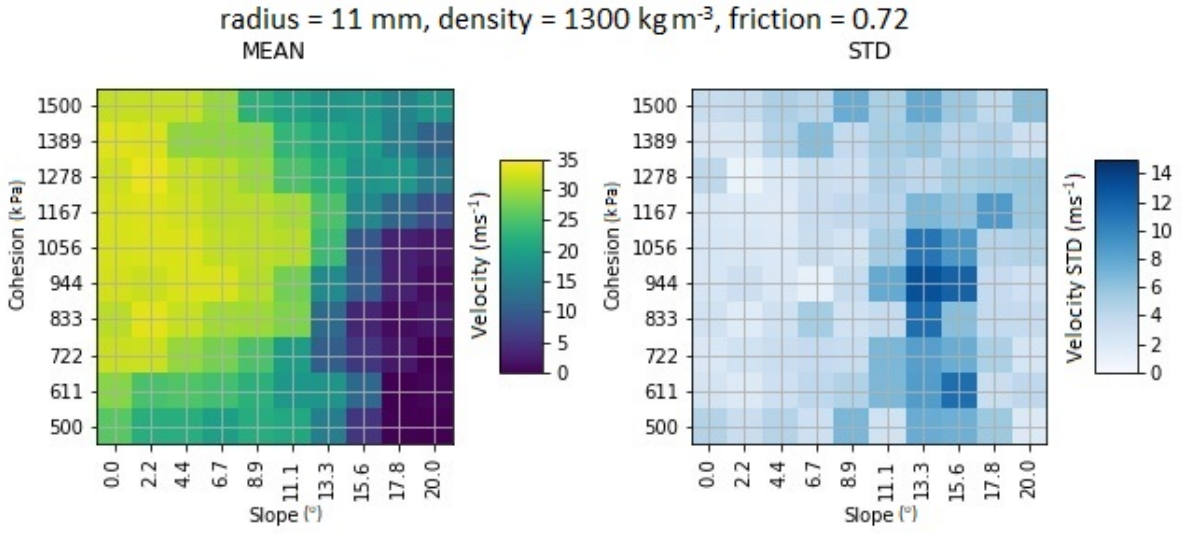

Figure 10: Slice of the predictions and corresponding variance.

5 CONCLUSIONS. FUTURE WORK

A main result of this interdisciplinary project was the identification of some of the constraints that come from integrating the different fields. Mobility simulations are computationally expensive and generate high-dimensional results. This means that a dataset of mobility simulations will contain (i) a relatively small number of simulations, and (ii) a large amount of information for each of the simulations. This regime is the exact opposite of the desired one for two reasons. First, some of the most successful machine learning algorithms are known to perform well when large amount of data is available, in contrast to (i). Second, currently available quantum computers have a small number of qubits and therefore can handle a limited number of variables, in contrast to (ii).

Among the available quantum machine learning proposals, we chose an algorithm that can in principle deal with some of the above constraints and, at the same time, be implemented on existing quantum hardware. In particular, the algorithm was designed for quantum annealing as implemented by the D-Wave 2000Q hardware. A simplified classical version was used for comparison purposes and to establish a baseline; the quantum-assisted version was used for the final results. Although the quantum algorithm did not excel, it did not perform worse than the classical version. This is promising in the sense that the algorithm was able to cope with noise and errors from the quantum annealer. As more general gate-model quantum computers become available and quantum algorithms evolve, we expect much better outcomes.

The machine learning model used here has a large number of hyperparameters: number of layers, number of units, learning rate, regularization factor, number of samples, and data encoding, to name a few. A thorough exploration of these would require a large number of executions and was beyond the scope and timeline of this work. Although we successfully proposed and implemented a D-wave-assisted alternative for the GO/NO-GO problem, more work is needed to determine whether there could be any advantage in this approach or if it would be better to consider quantum machine learning implementations of other alternative solutions to this problem; e.g., via Bayesian optimization. However, it is likely that implementation of

these alternative machine learning techniques will require quantum hardware resources beyond the state-of-the-art of currently available devices.

Future work. There are many routes open to expanding this work. As an early stage implementation of sampling from a quantum annealer to assist classical machine learning, it stands as one of the first efforts to address the problems facing near-term implementation of quantum devices. Lessons learned here inform the development of quantum algorithms that will eventually surpass their classical counterparts. We take it as a given that quantum devices will be used in some way for computation, with quantum-accelerated subroutines within hybrid classical-quantum architectures, like the one presented here, a likely scenario.

However, much remains to be done before quantum-assisted algorithms are mature enough to be robustly applied to complex engineering and scientific problems. We highlight here three key potential avenues for further exploration.

(i) First, other classical machine learning architectures can be explored for the opportunity of replacing core subroutines with quantum-assisted algorithms. As there are promising advantages in the implementation described in this paper, the lessons learned here can be applied to the development of novel and potentially improved hybrid algorithms. Alternative quantum computing architectures (notably gate-based) have passed small-scale, proof of principle, tests and are now coming to the second major hurdle: scalability. Algorithms such as the one described here must be tailored, implemented, and analyzed to map their associated range of applicability, advantages, and limitations.

(ii) A second route is further research into applications of the generative model learned by the QAHM. The generative model can potentially be applied to a wide range of scenarios, such as time-series analysis, compression, fast search of data through example generation, and assessing the effect of dataset size and complexity on the model.

(iii) Finally, we see great benefit for further research into potential QAHM variants, including placing the annealer at the top of the recognition network. This particular QAHM algorithm has several advantages: the classical recognition network can map continuous variables and large data vectors to a small quantum device, the quantum device does not need to be queried for each data point, and the temperature of the system does not need to be calculated, see Section 2. With these advantages in mind, there are significant incentives to look at QAHM variants, both to see if there can be improvements made to the model learned and to determine if alternative implementations can be applied to a wider class of applications.

While the above directions are focused on the QML aspects of further exploration, there are interesting adjustments that can be considered on the simulation side, to make it more amenable to a QML treatment and to fully tap into the potential advantages of using QML over classical machine learning. In particular, the type of mobility problems considered here invite a higher-dimensional input space; in other words, more input parameters, which would allow an increased number of nodes in the visible and hidden layers. However, more parameters to be learned will require significantly larger training datasets. This suggests the use of using lower-fidelity simulations, assumed to be more expeditious. The disadvantages of lower accuracy can potentially be offset by a deeper learning of the parameter space and the generative capabilities of the model. Furthermore, such a model may be able to capture the uncertainty in

these lower fidelity simulations.

Finally, it is worth noting that the nascent field of quantum computing requires the identification and exploration of new areas of scientific and engineering applications as candidates for quantum acceleration. It is the challenges posed by real-world problems that inform and direct the development of new algorithms and hardware.

Acknowledgments

This work was supported by a U.S. Army TARDEC project. Chrono development was supported in part by U.S. Army TARDEC Rapid Innovation Fund grant No. W911NF-13-R-0011, Topic No. 6a, "Maneuverability Prediction". Support for the development of Chrono::Vehicle was provided by U.S. Army TARDEC grant W56HZV-08-C-0236.

REFERENCES

[1] R. B. Ahlvin and P. W. Haley. *"NATO Reference Mobility Model: Edition II. NRMM User's Guide"*. U.S. Army Engineer Waterways Experiment Station, 1992.

[2] M. McCullough, P. Jayakumar, J. Dasch, and D. Gorsich. "The Next Generation NATO Reference Mobility Model Development". In *Proc. 8th Americas Regional Conference of the International Society for Terrain-Vehicle Systems*, 2016.

[3] D-Wave Systems Inc. "D-Wave 2000Q Technology Overview". `https://www.dwavesys.com/resources/publications`. Accessed: 2018-05-17.

[4] R. Serban, M. Wilson, M. Benedetti, J. Realpe-Gómez, and A. Perdomo-Ortiz. "Quantum Annealing for Mobility Studies: Generation of GO/NO-GO Maps via Quantum-Assisted Machine Learning". Technical Report TR-2018-03, Simulation-Based Engineering Laboratory, University of Wisconsin-Madison, 2018.

[5] S. Lloyd, M. Mohseni, and P. Rebentrost. "Quantum Algorithms for Supervised and Unsupervised Machine Learning". *arXiv:1307.0411*, 2013.

[6] M. Benedetti, J. Realpe Gómez, and A. Perdomo-Ortiz. "Quantum-Assisted Helmholtz Machines: A Quantum-Classical Deep Learning Framework for Industrial Datasets in Near-term Devices". *Quantum Science and Technology*, 3(3):034007, 2018.

[7] G.E. Hinton, P. Dayan, B.J. Frey, and R.M. Neal. "The Wake-Sleep Algorithm for Unsupervised Neural Networks". *Science*, 268(5214):1158, 1995.

[8] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz. "Quantum-Assisted Learning of Hardware-Embedded Probabilistic Graphical Models". *Phys. Rev. X*, 7:041052, Nov 2017.

[9] Project Chrono. "Chrono: An Open Source Framework for the Physics-Based Simulation of Dynamic Systems". `http://www.projectchrono.org`. Accessed: 2018-02-07.

[10] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut. "Chrono: An Open Source Multi-physics Dynamics Engine". In T. Kozubek, editor, *Lecture Notes in Computer Science, in print*, pages –. Springer, 2016.

[11] A. Tasora, D. Mangoni, D. Negrut, R. Serban, and P. Jayakumar. "Deformable Soil with Adaptive Level of Detail for Tracked and

Wheeled Vehicles". *Intl. J. Veh. Performance*, in press, 2018.

[12] R. Serban, M. Taylor, D. Negrut, and A. Tasora. "Chrono::Vehicle Template-Based Ground Vehicle Modeling and Simulation". *Intl. J. Veh. Performance*, in press, 2018.

[13] C. O'Sullivan. "Particle-Based Discrete Element Modeling: Geomechanics Perspective". *Int. J. Geomech.*, 11(6):449–464, 2011.

[14] P. Cundall and O. Strack. "A Discrete Element Model for Granular Assemblies". *Geotechnique*, 29:47–65, 1979.

[15] H. Kruggel-Emden, S. Wirtz, and V. Scherer. "A Study of Tangential Force Laws Applicable to the Discrete Element Method (DEM) for Materials with Viscoelastic or Plastic Behavior". *Chem. Eng. Sci.*, 63:1523–1541, 2008.

[16] A. Pazouki, M. Kwarta, K. Williams, W. Likos, R. Serban, J. Jayakumar, and D. Negrut. "Compliant Versus Rigid Contact: A Comparison in the Context of Granular Dynamics". *Phys. Rev. E*, 96, 2017.

[17] M. Anitescu. "Optimization-Based Simulation of Nonsmooth Rigid Multibody Dynamics". *Mathematical Programming*, 105(1):113–143, 2006.

[18] M. Anitescu and A. Tasora. "An Iterative Approach for Cone Complementarity Problems for Nonsmooth Dynamics". *Computational Optimization and Applications*, 47(2):207–235, 2010.