RADICAL LAB END OF SUMMER INTERNSHIP REPORT

Esraa Abdelmotteleb

I spent the first few weeks familiarizing myself with high performance computing and the work done by RADICAL lab. Some of the books I read include “Introduction to High-Performance Scientific Computing” by Victor Eijkhout and “High Performance Computing” by Charles Severance and Kevin Dowd. Addiotionally, I read the major papers produced by RADICAL such as the building blocks series.

After this, I learned about RADICAL Cybertools including RADICAL-Pilot,Ensemble Toolkit, and RADICAL-SAGA. I wrote a few applications that use EnTK and ran them on high performance computing machines suck as Comet and Stampede 2. My first set of EnTK applications aimed to understand EnTK properties. Figures 1 and 2 depict the results of an experiment to verify EnTK’s ability to run tasks sequentially and in parallel. Figure 1 shows the results of running 1-5 tasks, each sleeps for 100 seconds, concurrently on comet supercomputer while figure 2 shows the results of running those tasks sequentially. The two graphs show that running tasks sequentially

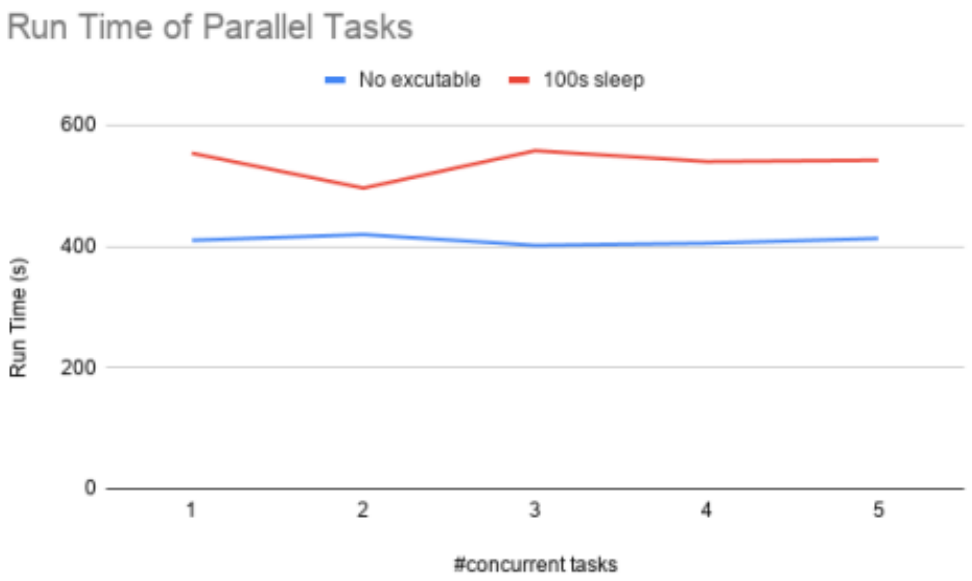

Figure 1: Run times of 1-5 tasks executing concurrently on comet, each sleeps for 100 seconds

For all of the jobs, the overhead for using EnTK and running on comet supercomputer was ~300s-550s. The time to terminate the application manager on average was 185.8s.

One of EnTK’s properties that I investigated was strong and weak scaling. Strong scaling concerns the speedup for a fixed problem size with respect to the number of processors. Weak scaling concerns the speedup for a scaled problem size with respect to the number of processors.

To test the weak scaling properties of EnTK, I performed 3 experiments each consisting of a varying number of pipelines. Each pipeline contains 12 stages each containing 1 task(100s

stress-ng). The number of pipelines = 4* the number of cores. The results in figure 3 show a gradual increase in TTX followed by a more steep increase when going from 8 to 16 cores. These results are consistent among the three experiments.

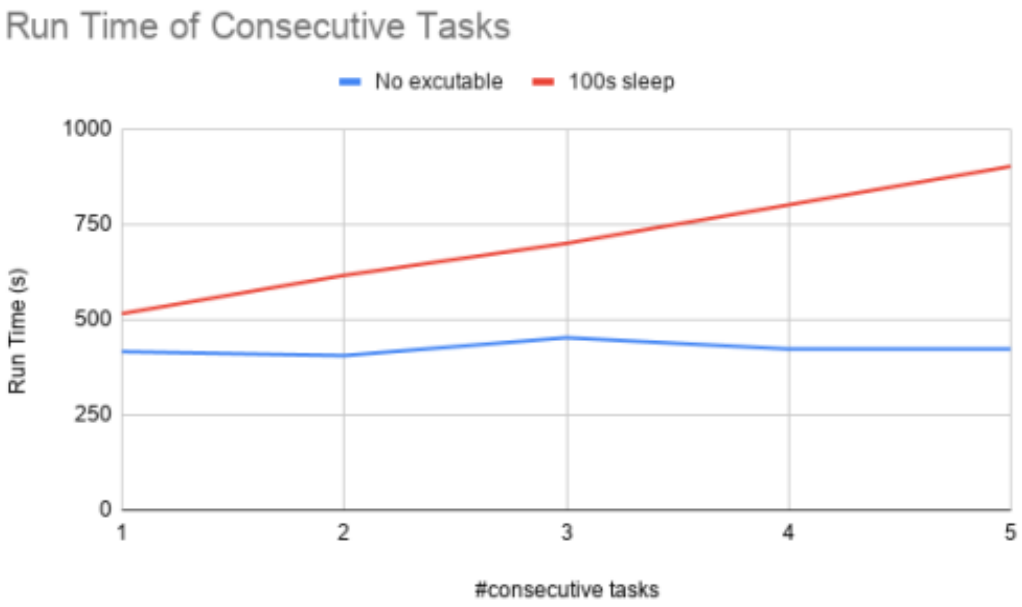

Figure 2: Run times of 1-5 tasks executing sequentially on comet, each sleeps for 100 seconds

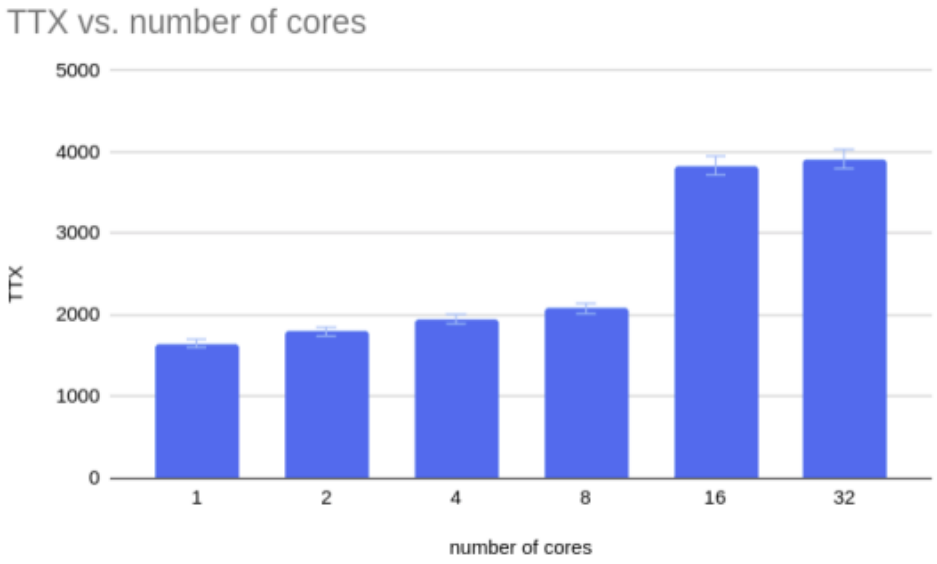

Figure 3: EnTK's week scaling experiment results

To understand EnTK's strong scaling properties, I performed 4 experiments each repeated 3 times. The structure and executables are shown in table 1.

Experiment number	Number of pipelines	stages/pipeline *	Executable
1	10	12	100-second stress
2	30	4	100-second stress
3	30	4	Generates random numbers for 100 seconds
4	10	12	Generates random numbers for 100 seconds

Table 1: Summary of EnTK's strong scaling experiment structure and executables
*each stage consists of 1 task

I expected the time needed to execute an application to decrease in half (almost) with the doubling of the number of cores.

In experiments 1 and 4, TTX remained the same despite the increase in the number of cores. This is different from my expectation. Since I use two different executables in the 2 experiments, it's unlikely that a problem with the executables is the reason for the observed result. The results suggest that no significant parallelization was happening with the increase in the number of cores; ie, maximal parallelization is happening with only 1 core. This is unexpected since each core has only 1 thread. To enable more parallelization, I increased the number of pipelines while keeping the total number of tasks the same as in experiment 1. The results of experiments 2 and 3 were consistent with my expectations.

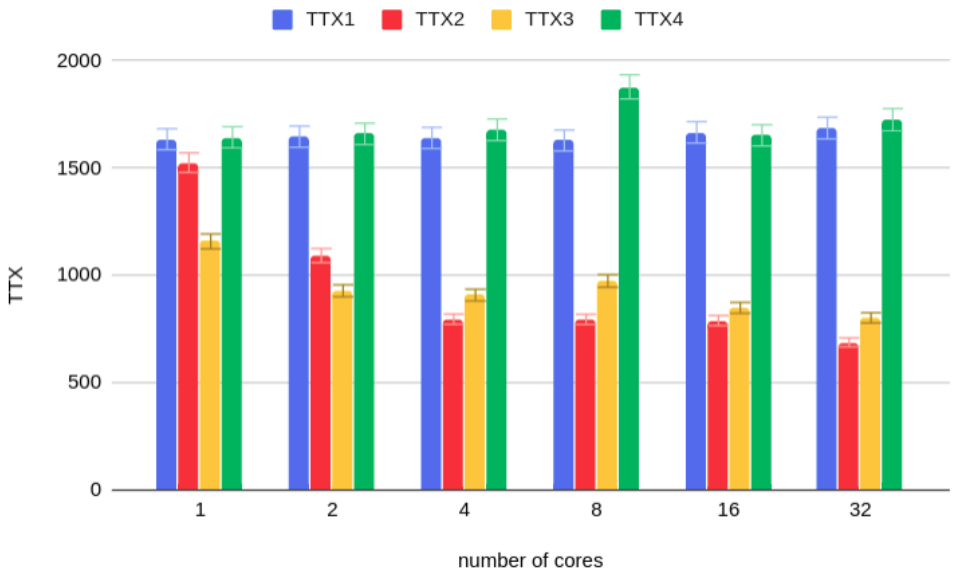

Figure 3: EnTK's strong scaling experiment results

An important part of my work this summer involved studying and classifying smart surrogates and classified them. In order to classify surrogates, I investigated some taxonomy development methodologies [1-3] and decided to use Bailey's three-level indicator model and combine both the empirical to deductive approach and the deductive to empirical approach [2]. Figure 4 shows a summary of this method. Taxonomy development is an iterative active process. The taxonomy below is the result of the 3rd iteration. Note that the classification consists of multiple dimensions; each dimension encompasses related characteristics.

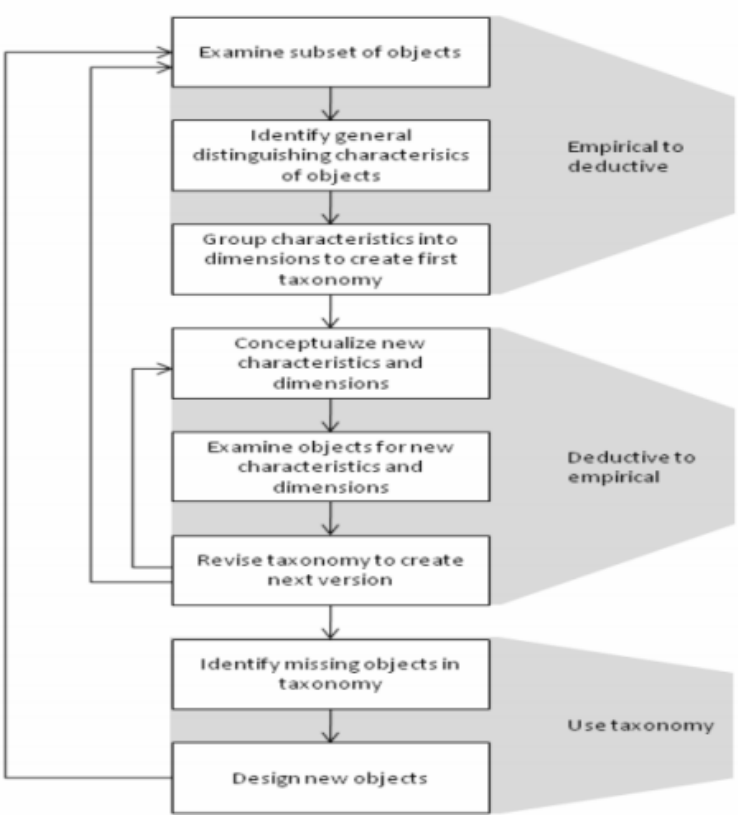

Figure 4: classification method summary

Surrogates Classification:

1- Functionality Dimension: concerned with the functionality of the surrogate and the link between ML and HPC:

* *Link between ML and HPC*: HPCforML uses HPC to execute and enhance ML performance, or uses HPC simulations to train ML algorithms, which are then used to understand experimental data or simulations. MLforHPC uses ML to enhance HPC applications and systems [10].

**Functionality*: Smart surrogates can fulfill different functionalities including 1) Use HPC simulations to train surrogates. This can be done by engaging supercomputing resources to take advantage of high-dimensional training data produced from simulations and/or experimental data to train a neural network to predict an event of interest (For example, predict disruptive instabilities in controlled fusion plasmas) [23], 2) replace an internal simulation component (for example, learn an accurate and transferable potential for organic molecules) [24], 3) to guide computations (for example, choose the optimal training data) [26]. and 4)to extract patterns.
Note: Surrogates that require the use of summary statistics cannot serve as universal approximators . Thus, having the ability to learn the underlying distributions of data and

emulate the observable without imposing the choice of summary statistics, as in the traditional approach to emulators, is valuable.

2 - Problem Dimension: concerned with the algorithmic area and the dimentionality of the problems.

* *Algorithmic area*: The scientific problem the surrogate is investigating. For example: partial differential equations, particle dynamics.

3- Algorithmic dimension: concerned with the machine learning model used and the resulting architecture and its properties.

* *Machine learning model*: most work to date in building surrogates uses random forests[4], Gaussian Processes[5], CNN[14], FCN[20], RNN[20], or other machine learning models [6,7]. Non- Neural Network models have limited ability to model high-dimensional inputs/outputs. Dimensionality reduction techniques can enable higher dimensional inputs/outputs but the improvement is limited [21,22]. Thus, Neural Network models are more suitable for processing natural n-dimensional signals. However, most existing deep active learning methods are concerned with classification tasks while Neural surrogates are generally concerned with regression tasks.

* *Machine Learning Architecture*: Neural surrogate can use different architectures to handle different tasks. Most neural surrogates have their own unique architecture. Recurrent neural networks (RNNs) powerfully handle sequential data by maintaining information in an internal state that is passed between successive time steps, in addition to taking into account new input data at every time step. Meanwhile, convolutional neural networks (CNNs) can learn salient, low-dimensional representations from high-dimensional data by successively applying convolutional and downsampling operations. Some neural surrogates use already existing architecture while others use new architectures.

Note: finding the “correct” architecture is instrumental in the successful learning of surrogate models. For example, CNN priors inherently rely on their architectures, one has to find an architecture that gives the suitable prior of a given problem. However, required optimal architecture is likely to change when training data changes. Therefore, it is important to have a variable architecture that can adjust with associated uncertainty

* *Variability of the architecture*: Most surrogates used in scientific workflow use a fixed architecture [4-7]. However, a recent paper [14] successfully used NAS (Neural Architecture Search) in order to infer the architecture of the CNN-based neural surrogate. Further work needs to be done to use NAS with other NN models.

4- Data dimension: is a sub-dimension of the algorithmic dimension. Concerned with the generation, labeling and selection of the data used to train the ML model.

* *Data set generation and labeling*: in supervised (sequential) learning, the labeled data set is given beforehand and learning is performed afterwards (offline)[4]. In concurrent machine learning, the data set is generated and labeled as model training proceeds (Online) [8]. In active learning, we are given the unlabeled data, and we decide which ones should be labeled during the training process[9].

* *Data selection method*: Most surrogates models require human intervention for the selection and adjustment of training data [4-7]. However, an active approach to surrogates learning that enables automatic training data selection has emerged in GP surrogates[15].

* *Data Volume*: Since the training data generation typically involves numerical solutions of the underlying micro-scale model, a process that is often quite expensive, learning an efficient surrogate is a small data problem. Datasets vary in size from of a few hundred[18] or a few thousand[16,17,19] individual simulations. The dataset that one uses to construct the model should be a good representation of all the practical situations that the model is intended for.

* *Data span*: training and the testing examples could be drawn uniformly across all experimental runs (thus requiring low generalizability). On the other hand, training and testing data could be drawn from slightly different distributions (for example, using data from one reactor to simulate the performance of another reactor) [24].

To further my understanding of the classification, I studied and summarized 8 surrogates then classified them according to the classification scheme above. Table 2 shows the results.

Surrogate Paper Name	**Functionality Dimension**	**Problem Dimension**	**Algorithmic Dimension: Machine learning model**	**Algorithmic Dimension: Machine learning model**
Predicting disruptive instabilities in controlled fusion plasmas through deep learning	HPC simulations to train surrogates	Nuclear reactors	FRNN	Fixed

ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost	Replace simulation component	Organic molecules	Fully-connected neural network with modified HDNN method of Behler and Parrinello	Fixed
CosmoGAN: creating high-fidelity weak lensing convergence maps using Generative Adversarial Networks	Replace a simulation	Convergence maps	Generative Adversarial Network	Fixed
Up to two billion times acceleration of scientific simulations with deep neural architecture search	Wide range of possible functionalities	N/A	Convolutional Neural Network	Variable
Machine Learning for Crystal Identification and Discovery	Replace part of a simulation	Crystal Identification	Artificial Neural Network (Not specified)	Fixed
Hidden physics models: Machine learning of nonlinear partial differential equations	Extract patterns	Differential equations	Gaussian processes	Fixed
Machine learning surrogates for molecular dynamics simulations of soft materials	Replace a simulation	Molecular dynamics	Artificial Neural Network (Not specified)	Fixed
Active learning of constitutive relation from mesoscopic dynamics for macroscopic modeling of non-Newtonian flows	Replace a simulation	Fluid dynamics`	Gaussian process regression	Fixed

Table 2: classification of eight surrogates

RESOURCES:

[1] D. E. Taylor. Survey and Taxonomy of Packet Classification Techniques. Technical Report WUCSE-2004-24, Washington Univ., St. Louis, May 2004.

[2] T. M. Mengistu and D. Che, "Survey and taxonomy of volunteer computing", ACM Comput. Surveys, vol. 52, no. 3, pp. 1-35, Jul. 2019..

[3] Nickerson, R., Muntermann, J., Varshney, U., & Isaac, H. (2009). Taxonomy Development in Information Systems: Developing a Taxonomy of Mobile Applications.

[4] JL Peterson, KD Humbird, JE Field, ST Brandon, SH Langer, RC Nora, BK Spears, and PT Springer. Zonal flow generation in inertial confinement fusion implosions. Physics of Plasmas, 24(3):032702, 2017.

[5] Juliana Kwan, Katrin Heitmann, Salman Habib, Nikhil Padmanabhan, Earl Lawrence, Hal Finkel, Nicholas Frontiere, and Adrian Pope. Cosmic emulation: fast predictions for the galaxy power spectrum. The Astrophysical Journal, 810(1):35, 2015.

[6] Felix Brockherde, Leslie Vogt, Li Li, Mark E Tuckerman, Kieron Burke, and Klaus-Robert M¨uller. Bypassing the kohn-sham equations with machine learning. Nature communications, 8(1):872, 2017.

[7] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert M¨uller, and O Anatole Von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. Physical review letters, 108(5):058301, 2012

[8] Linfeng Zhang, Han Wang, and Weinan E. Reinforced dynamics for enhanced sampling in large atomic and molecular systems. The Journal of Chemical Physics, 148(12):124113, 2018.

[9] Linfeng Zhang, De-Ye Lin, Han Wang, Roberto Car, and Weinan E. Active learning of uniformly accurate interatomic potentials for materials simulation. Physical Review Materials, 3(2):023804, 2019.

[10] Geoffrey Fox, James A. Glazier, JCS Kadupitiya, Vikram Jadhao, Minje Kim, Judy Qiu, James P. Sluka, Endre Somogyi, Madhav Marathe, Abhijin Adiga, Jiangzhuo Chen, Oliver Beckstein, and Shantenu Jha, “Learning Everywhere: Pervasive Machine Learning for Effective HighPerformance Computation,” presented at the HPDC Workshop at IPDPS, Rio de Janeiro [Online]. Available: https://arxiv.org/abs/1902.10810, http://dsc.soic.indiana.edu/publications/Learning_Everywhere_Summar y.pdf

[11] Geoffrey Fox, James A. Glazier, JCS Kadupitiya, Vikram Jadhao, Minje Kim, Judy Qiu, James P. Sluka, Endre Somogyi, Madhav Marathe, Abhijin Adiga, Jiangzhuo Chen, Oliver Beckstein, and Shantenu Jha, “Learning Everywhere: Pervasive Machine Learning for Effective HighPerformance Computation: Application Background,” Feb. 2019 [Online]. Available: http://dsc.soic.indiana.edu/publications/Learning_Everywhere.pdf

[12] Geoffrey Fox, Shantenu Jha, “Understanding ML driven HPC: Applications and Infrastructure,” in IEEE eScience 2019 Conference, San Diego, California [Online]. Available: https://escience2019.sdsc.edu/

[13] Han, Jiequn, and Linfeng Zhang. "Integrating Machine Learning with Physics-Based Modeling." arXiv preprint arXiv:2006.02619 (2020).
[14] MF Kasim, D Watson-Parris, L Deaconu, S Oliver, P Hatfield, DH Froula, G Gregori, M Jarvis, S Khatiwala, J Korenaga, et al. Up to two billion times acceleration of scientific simulations with deep neural architecture search. ArXiv preprint arXiv:2001.08055, 2020
[15] Francisco Sahli Costabalab, Paris Perdikaris, Ellen Kuhle, and Daniel E. Hurtado. Multifidelity classification using gaussian processes: accelerating the prediction of large-scale computational models. Computer Methods in Applied Mechanics and Engineering, 357:112602, 2019
[16] Clark, Daniel S., Steven W. Haan, and Jay D. Salmonson. "Robustness studies of ignition targets for the National Ignition Facility in two dimensions." Physics of Plasmas 15.5 (2008): 056305.
[17] Kritcher, A. L., et al. "Metrics for long wavelength asymmetries in inertial confinement fusion implosions on the National Ignition Facility." Physics of Plasmas 21.4 (2014): 042708.
[18] J. Gu, Jianfa, et al. "A new metric of the low-mode asymmetry for ignition target designs." Physics of Plasmas 21.1 (2014): 012704.
[19] Spears, Brian K., et al. "Performance metrics for inertial confinement fusion implosions: Aspects of the technical framework for measuring progress in the national ignition campaign." Physics of Plasmas 19.5 (2012): 056316.
[20] M. W. Coughlin, T. Dietrich, Z. Doctor, D. Kasen, S. Coughlin, A. Jerkstrand, G. Leloudas, O. McBrien, B. D. Metzger, R. O'Shaughnessy, and S. J. Smartt. Constraints on the neutron star equation of state from AT2017gfo using radiative transfer simulations. , 480:3871–3878, November 2018
[21] I. Bilionis, N. Zabaras, B. A. Konomi, and G. Lin. Multi-output separable gaussian process: towards an efficient, fully bayesian paradigm for uncertainty quantification. Journal of Computational Physics, 241:212–239, 2013
[22] J. Dhamala, Pradeep Baracharya, Hermenegild J. Arevalo, J. L. Sapp, B. M. Horacek, Katherine C. Wu, Natalia A. Trayanova, and L.Wang. Embedding high-dimensional bayesian optimization via generative modeling: Parameter personalization of cardiac electrophysiological models. Medical Image Analysis, page accepted, 2020.
[23] Julian Kates-Harbeck, Alexey Svyatkovskiy, and William Tang. Predicting disruptive instabilities in controlled fusion plasmas through deep learning. Nature, 568(7753):526–531, 1 April 2019.
[24] Julian Kates-Harbeck, Alexey Svyatkovskiy, and William Tang. Predicting disruptive instabilities in controlled fusion plasmas through deep learning. Nature, 568(7753):526–531, 1 April 2019.
[25] Mustafa Mustafa, Deborah Bard, Wahid Bhimji, Zarija Lukic, Rami Al-Rfou, and Jan M Kratochvil. ´ Cosmogan: creating high-fidelity weak lensing convergence maps using generative adversarial networks. Computational Astrophysics and Cosmology, 6(1):1–13, 2019

[26] Justin S Smith, Ben Nebgen, Nicholas Lubbers, Olexandr Isayev, and Adrian E Roitberg. Less is more: Sampling chemical space with active learning. The Journal of chemical physics, 148(24):241733, 2018
[27] Maziar Raissi and George Em Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. Journal of computational physics, 357:125–141, 15 March 2018
[28] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Multistep neural networks for data-driven discovery of nonlinear dynamical systems. 4 January 2018.
[29] Zarija Lukić, Rami Al-Rfou, Wahid Bhimji, Deborah Bard, Mustafa Mustafa. Creating Virtual Universes Using Generative Adversarial Networks. 2017
[30] Julian Kates-Harbeck, Alexey Svyatkovskiy, and William Tang. Predicting disruptive instabilities in controlled fusion plasmas through deep learning. Nature, 568(7753):526–531, 1 April 2019.
[31] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. Chemical science, 8(4):3192–3203, 2017.
[32] Mustafa Mustafa, Deborah Bard, Wahid Bhimji, Zarija Lukic, Rami Al-Rfou, and Jan M Kratochvil. ´ Cosmogan: creating high-fidelity weak lensing convergence maps using generative adversarial networks. Computational Astrophysics and Cosmology, 6(1):1–13, 2019.
[33] MF Kasim, D Watson-Parris, L Deaconu, S Oliver, P Hatfield, DH Froula, G Gregori, M Jarvis, S Khatiwala, J Korenaga, et al. Up to two billion times acceleration of scientific simulations with deep neural architecture search. arXiv preprint arXiv:2001.08055, 2020.
[34] Matthew Spellings and Sharon C Glotzer. Machine learning for crystal identification and discovery. AIChE journal. American Institute of Chemical Engineers, 64(6):2198–2206, 30 June 2018.
[35] Maziar Raissi and George Em Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. Journal of computational physics, 357:125–141, 15 March 2018.
[36] J C S Kadupitiya, Fanbo Sun, Geoffrey Fox, and Vikram Jadhao. Machine learning surrogates for molecular dynamics simulations of soft materials. Journal of computational science, 42:101107, 1 April 2020
[37] Lifei Zhao, Zhen Li, Bruce Caswell, Jie Ouyang, and George Em Karniadakis. Active learning of constitutive relation from mesoscopic dynamics for macroscopic modeling of non-newtonian flows. Journal of Computational Physics, 363:116–127, 2018.
[38] Hyungro Lee, Heng Ma, Matteo Turilli, Debsindhu Bhowmik, Shantenu Jha, Arvind Ramanathan (2019), GitHub repository,https://github.com/radical-collaboration/DeepDriveMD