Adversarial Forces of Physical Models

Ekin D. Cubuk
Google Research, Brain Team

Samuel S. Schoenholz
Google Research, Brain Team

Abstract

While most systems are governed by quantum mechanics at the nanoscale, it is almost always prohibitively expensive to simulate these systems by exactly solving Schrödinger's equation. For this reason, a hierarchy of approximate models are commonly used in biology, chemistry, and materials science that allow practitioners to trade-off between accuracy and speed to simulate larger systems at longer time scales. Recently, significant attention has been devoted to leveraging machine learning to develop new and more accurate approximations. While these approximate models have typically been assessed based on their average-case performance, recent work in the adversarial example literature in other domains has offered ample evidence that this is often a poor indicator of worst-case performance. Here we show that there is a well defined sense of adversarial direction that governs the worst-case behavior for these approximate models of physical systems trained for energy prediction. Unlike in other contexts, where adversarial examples are scarce absent malicious intervention, in physical systems we show that the laws of physics naturally lead the system to move in adversarial directions. Surprisingly, we find that these adversarial directions exist even for traditional, analytic force fields such as the BKS potential. We verify our predictions by comparing a variety of hand-designed and machine learned models of quantum mechanical energies, including Behler-Parrinello and graph neural networks trained on energies or forces, and *ab initio* quantum mechanical calculations. We conclude by discussing strategies that can prevent a physical model from moving in its adversarial directions.

1 Introduction

Although quantum mechanics offers a framework to exactly solve for the dynamics of physical systems, it is intractable in all but the simplest cases. To overcome these computational difficulties, physicists have developed a hierarchy of approximate models that span a wide range of accuracy-to-performance trade-offs (e.g. Ref.[1–5], see Related Work in Appendix for more examples). Recently, neural networks have become essential building blocks of these approximate models; however, the validity of these approximate models is usually reported averaged over test data. In this paper we take inspiration from adversarial examples in computer vision and evaluate the worst-case performance of a wide range of approximate physical models.

We show that adversarial examples proliferate approximate energy models, including state-of-the-art neural networks and hand-designed polynomial fits that predate deep learning. Moreover, we show that physical systems can sometimes naturally be driven in these adversarial directions when using Newton's laws to simulate molecular systems. This is in sharp contrast to adversarial examples in computer vision where a malicious adversary is required to produce them. As such, although the relevance of adversarial examples in computer vision is debated [6–8], dealing with them seems to be an essential step in developing robust physical models.

Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020), Vancouver, Canada.

2 Approximate energy models in computational sciences

According to quantum mechanics, the properties of a system can be derived from the Hamiltonian operator, whose eigenvalues correspond to the energy. In less accurate classical approximations, the energy of a system still plays a central role. The positions of N atoms in 3-dimensions will be defined by a vector $\vec{R} \in \mathbb{R}^{N\times 3}$. Either via Density Functional Theory (DFT) or using an empirical potential we can assign an energy, $E(\vec{R})$, to a given configuration. With an energy in hand, the atoms move in the direction of the forces on them which is given by the gradient of potential energy with respect to atomic positions $\vec{F} = -\nabla_{\vec{R}} E(\vec{R})$, where $\vec{F}$ denotes the forces on N atoms, and $\vec{R}$ denotes the positions of atoms. The accuracy of materials property predictions (such as mechanical, dynamic, and catalytic) are dependent on the accuracy of $E(\vec{R})$ compared to the quantum mechanical energy.

It follows that, an important metric for measuring the accuracy of an approximate energy model is:

$$\mathcal{L} = \sum_j \left[E^{\mathrm{Acc}}(\vec{R}_j) - E^{\mathrm{App}}(\vec{R}_j) \right]^2, \tag{1}$$

where j is an index that runs over the atomistic configurations (samples) in a dataset, $E^{\mathrm{Acc}}(\vec{R})$ is the more accurate but slower energy model, and $E^{\mathrm{App}}(\vec{R})$ is the approximate energy model. In machine learning (ML) applications, DFT is often used to compute $E^{\mathrm{Acc}}(\vec{R})$ and a neural network is trained to predict $E^{\mathrm{App}}(\vec{R})$. The loss in Equation (1) is most commonly used loss for training ML models to speed up quantum mechanical calculations[9–13] (or it is commonly used as one of the terms in the loss [14, 15]). However, the analysis in this paper is not restricted to models that use the particular loss in Eq. 1. For ML models that use a different loss (e.g. one in terms of forces or other materials properties) or even for non-ML energy models, the value of $\mathcal{L}$ is still extremely relevant for quantifying the quality of the model. As described above, we will consider non-ML models that were not trained using Eq. (1) as well as various ML models.

3 Adversarial directions for approximate physics models

To generate adversarial examples for physical systems by analogy to previous work in vision, we will seek to maximize the discrepancy between the predictions of our model and the ground truth target. It follows that the adversarial direction for the j^{th} configuration in a dataset is defined by taking the gradient of Eq. 1 with respect to $\vec{R}_j$:

$$\vec{\mathcal{A}}(\vec{R}_j) \equiv \nabla_{\vec{R}_j} \mathcal{L} = \nabla_{\vec{R}_j} \left(E^{\mathrm{Acc}}(\vec{R}_j) - E^{\mathrm{App}}(\vec{R}_j) \right)^2 \tag{2}$$

$$= \underbrace{2 \left(E^{\mathrm{Acc}}(\vec{R}_j) - E^{\mathrm{App}}(\vec{R}_j) \right)}_{\text{scalar}} \left(\vec{F}^{\mathrm{App}}(\vec{R}_j) - \vec{F}^{\mathrm{Acc}}(\vec{R}_j) \right). \tag{3}$$

Thus the adversarial direction can be written as a scalar multiplying the difference between the forces on the atoms according to the accurate energy model and the forces according to the approximate model. Physical systems of interest are often close to their local minimum in the energy landscape. For systems in equilibrium, for instance, we know that the probability of observing a configuration decreases exponentially with its energy ($P(\vec{R}_j) \propto \exp(-\Delta E(\vec{R}_j)/kT)$). By construction, configurations that are close to a local minimum will feature small forces in the accurate model and so $||\vec{F}^{\mathrm{Acc}}|| \ll 1$. However, if the forces calculated by the approximate energy model ($\vec{F}^{\mathrm{App}}$) are not small, the adversarial direction can be approximated as (i.e. assuming $||\vec{F}^{\mathrm{App}}|| \gg ||\vec{F}^{\mathrm{Acc}}||$):

$$\vec{\mathcal{A}}(\vec{R}_j) \approx C \vec{F}^{\mathrm{App}}(\vec{R}_j), \tag{4}$$

where C is a scalar that can be positive or negative. For configurations where this approximation is valid, the approximate energy model will move the system in its own adversarial direction, and thus adversarial examples will likely be encountered. In the next section, we will measure the validity of our approximation for several model systems: including neural networks models, traditional analytic models, and quantum mechanical calculations.

	$\epsilon = 0$	$\epsilon = 0.1$	$\epsilon = 0.3$	$\epsilon = 0.5$
Random direction	5.1	5.3	5.9	6.8
Adversarial direction	5.1	15.4	31.9	42.0

Table 1: **Error grows faster due to distortions in the adversarial direction compared to random directions.** Error is measured in meV/atom, and ϵ is the L2-norm of the distortion, measured in Å^2. Typically atoms are separated by several Å, thus a total distortion size of $\epsilon = 0.5$ Å^2 is very small.

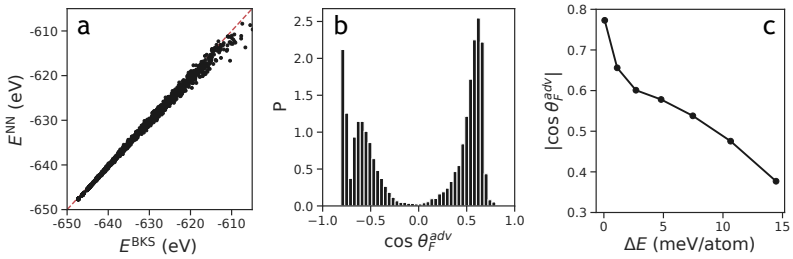

Figure 1: **BP-NN with accurate energy predictions has adversarial directions aligned with its forces.** NN is trained on BKS energy calculations of SiO_2 configurations. **a)** Points represent the energy prediction of the neural network (E^{NN}) vs. the labels (E^{BKS}) for test configurations. Red dashed line denotes perfect agreement. **b)**: Histogram density of $\cos\,\theta_F^{adv}$ for test configurations. **c)** Average of $|\cos\,\theta_F^{adv}|$ vs. ΔE, the energy (E^{Acc}) above the local minimum. We see that the magnitude of overlap between the adversarial direction and $\vec{F}^{\mathrm{App}}$ goes down with ΔE.

We now investigate the presence of adversarial directions, and their overlap with the force, for range of choices for E^{Acc} and E^{App}. After training the approximate energy model, E^{App}, we compute its adversarial direction $\vec{\mathcal{A}}(\vec{R}_j)$ using Eq. (3). To evaluate how aligned the adversarial direction is with the force, $\vec{F}^{\mathrm{App}}(\vec{R}_j)$, we compute the cosine-angle between the two vectors,

$$\cos\,\theta_F^{adv} = \frac{\vec{\mathcal{A}}(\vec{R}) \cdot \vec{F}^{\mathrm{App}}(\vec{R})}{||\vec{\mathcal{A}}(\vec{R})||\;||\vec{F}^{\mathrm{App}}(\vec{R})||}. \tag{5}$$

Note that both $\vec{\mathcal{A}}(\vec{R}_j)$ and $\vec{F}^{\mathrm{App}}$ are 3N-dimensional vectors where N is the number of atoms along with three spatial directions for each atom. Since our systems have 50 to 100 atoms, the cosine-angle between two random vectors is close to 0. For each system studied we evaluate the error of the model and the distribution of cosine-angles. We will see that even well-performing models typically have high alignment with adversarial directions. Moreover, we will find that the sign of the overlap is random; this agrees with the observation that the constant C in Eq. (4) will have a random sign. Additionally, we recall that Eq. (4) will be accurate when the system is near a minimum of the energy. To validate this approximation and to show that adversarial examples become particularly problematic near extrema, we also evaluate the average magnitude of the cosine-angle, $\cos\,\theta_F^{adv}$, as a function of the energy above the local minimum, ΔE.

3.1 Behler-Parrinello Neural Network (BP-NN) approximating BKS[1]

We train the BP-NN model to achieve per-atom root-mean-squared-error (RMSE) of 5.1 meV/atom, so that its accuracy is comparable to more realistic tasks, which is 4-7 meV/atom (for example, see [9, 16]). The training and test data is sampled by displacing atoms about their equilibrium positions, with the total displacement uniformly sampled between 0 and 1.0 Å^2. Fig. 1(a) is a scatter plot of labels and corresponding predicted energies for a test set of 2000 configurations. We then calculate the error after distorting the atoms, in all configurations, either randomly or in the direction of $\vec{\mathcal{A}}(\vec{R})$. Table 1 shows that while the error grows modestly when the atoms are moved in random directions, it grows significantly when they are moved in the adversarial direction by the same total distortion size. This measurement shows that atoms moving in the adversarial directions can be problematic for the fidelity of simulations.

The BP-NN model naturally moves in the adversarial direction, as argued in Section 3. To see this, we show in Fig. 1(b) that the magnitude of $\cos\,\theta_F^{adv}$ is large, which indicates that the adversarial

[1]BP-NN implementation within JAX-MD is available as a Colaboratory notebook at: github.com/google/jax-md/blob/master/notebooks/jax_md_cookbook.ipynb

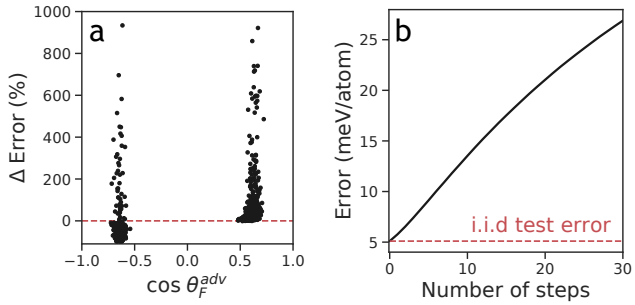

Figure 2: **Energy prediction error of the molecular physics model grows with every step. a)** Each point is the percentage increase in error after a single step. Red curve denotes no change. We see that even after a single step in the direction of $\vec{F}^{\text{App}}$, configurations for which $\cos\theta_F^{adv}(\vec{R}) > 0$ has significantly larger error. Interestingly, configurations for which $\cos\theta_F^{adv}(\vec{R}) < 0$ have lower error (naturally bounded by -100%). **b)** Energy prediction error as the system evolves under $\vec{F}^{\text{App}}$.

direction is approximately aligned (or anti-aligned) with $\vec{F}^{\text{App}}$ for most configurations. Note that the adversarial direction is in a $3N$ dimensional space, and two random vectors in this space have very small overlap. We see that the alignment is positive for some of the configurations and negative for the others (corresponding to C from Eq. 4 being negative or positive, respectively). Finally, we evaluate how the alignment depends on energy relative to the local minimum. In Section 3, we had argued that the approximation would only be valid for small $\vec{F}^{\text{Acc}}$, which corresponds to configurations near the local minimum. Fig. 1(c) shows that the magnitude of $\cos\theta_F^{adv}$ goes down with ΔE.

Estimating the error due to high $\cos\theta_{\mathbf{F}}^{\mathbf{adv}}$: In Fig. 2(a) we present the change in error after taking one step in the simulation (atoms are moved in the direction of $\vec{F}^{\text{App}}$ according to Newton's second law). We see that for configurations where the adversarial direction is pointing in the same direction as the neural network force (in other words, $\cos\theta_F^{adv}(\vec{R}) > 0$), the error on the next step is almost always larger, often by more than 100%. Interestingly, we see that for most of the configurations where the adversarial direction is pointing in the opposite direction of $\vec{F}^{\text{App}}$ (in other words, $\cos\theta_F^{adv}(\vec{R}) < 0$), the error on the next step of simulation goes down.

Next, we calculate the error as system takes multiple steps following $\vec{F}^{\text{App}}$. Molecular energy models are often evolved for many steps, up to thousands if the model is used for structural optimization, and up to 400 billion steps if the model is used for molecular dynamics [17]. In structural optimization, the goal is to find lower energy configurations for an atomistic systems (similar to finding lower-loss configurations of weights for neural networks). An indispensable component of structural optimization [18, 19] is gradient descent, where atoms are moved in the direction of forces for many iterations. In the case where the model force, $\vec{F}^{\text{App}}$, is aligned with the adversarial direction, it is plausible that the model error will grow with each step. To evaluate this, we apply gradient descent for 30 steps using $\vec{F}^{\text{App}}$. At every step, we also evaluate the correct energy, E^{Acc}, and report the error in Fig. 2(b). We see that before we take any steps, the error is same as the i.i.d test error. However, with each step, the error grows, and ends up more than 5 times larger after 30 steps. This suggests that for one of the most important applications of molecular physics models, the effective error of the model might be significantly larger than the error estimated on the i.i.d. test set.

3.2 Traditional analytic model approximating DFT

In the previous example, we analyzed a commonly used, machine learned energy model. As mentioned above, the loss defined by Eq. 1 is an important metric for evaluating the accuracy of any model, even if it is not trained with this loss function, or even if it is not an ML model. Following this intuition, we study the adversarial directions of the BKS model [20–22], a traditional analytic model that only has 12 free parameters (see Appendix 4 for details about the BKS potential).

For this particular experiment, we use DFT as (E^{Acc}), and the BKS potential (defined by Eq. 6), as E^{App}. Despite the small number of fitting parameters in the BKS potential and the fact that the parameters are not trained to reproduce DFT energies, E^{Acc} and E^{App} are in decent agreement for configurations we sampled by randomly displacing atoms about their equilibrium position, with a

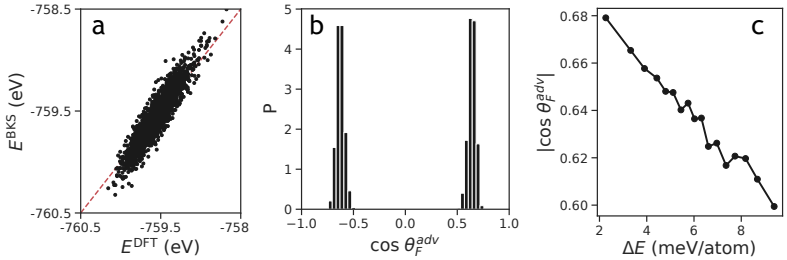

Figure 3: **A simple analytic model with 12 parameters still has adversarial directions aligned with its forces.** The BKS model was not trained using DFT energies. **a)** Points represent the energy prediction (E^{BKS}) vs. the labels (E^{DFT}). Red dashed line denotes perfect agreement. **b)**: Histogram density of $\cos\theta_F^{adv}$ for test configurations. **c)** Average magnitude of $\cos\theta_F^{adv}$ vs. ΔE.

total distortion size of 0.3 Å^2 (see Fig. 3(a)). Next, we measure $\cos\theta_F^{adv}$ for the BKS model. We see that for this analytic energy model, the adversarial directions are still aligned with $\vec{F}^{\mathrm{App}}$, but perhaps to a smaller extent than the machine-learned models. We emphasize that the BKS model is not an ML model, so the adversarial direction considered here is not the adversarial direction of the training objective. Nonetheless, Eq. 1 represents one of the most important metrics that define the quality of this model and we see that forces will move atoms in directions that strongly correlate with $\vec{\mathcal{A}}(\vec{R}_j)$.

Avoiding adversarial directions: The best defense against adversarial examples is adversarial training [23, 24], which involves training on adversarial examples. A direct analogy for physical models would be to train on configurations reached by taking a step in the adversarial direction. A related protocol would be to augment the training set by taking a step from the configurations in the training set in the direction of $\vec{F}^{\mathrm{App}}$. While this is not exactly adversarial training, it is related, since the adversarial directions and $\vec{F}^{\mathrm{App}}$ are correlated. We have seen at least one example of this training protocol in the literature, where the authors found that including configurations that are 1 to 2 steps in the future of their training configurations, as predicted by their neural network, improved the overall quality of their simulation [25]. Similarly, Ref. [26] found that adding Gaussian noise during training reduced their point prediction accuracy, but improved their roll-out accuracy (which involves taking steps in the direction of their $\vec{F}^{\mathrm{App}}$). Adding Gaussian noise during training is known to increase adversarial robustness while reducing clean accuracy (accuracy on validation samples that come from the same distribution as the training set) [27], which might explain why Sanchez-Gonzalez et al. found that their models trained with noise had worse clean accuracy but better roll-out accuracy [26].

Another related protocol to adversarial training is to train on $\vec{F}^{\mathrm{Acc}}$. Since $\vec{F}^{\mathrm{Acc}}$ is also correlated with $\vec{F}^{\mathrm{App}}$, this protocol is also likely to have benefits for adversarial robustness. To test this, we trained a graph neural network (GNN) [28, 14, 29–31] on DFT energies, and we trained another GNN on the energies and forces on the same configurations, using JAX-MD [32, 33][2]. We found that the model trained only on energies has large $|\cos\theta_F^{adv}|$ for most samples in the validation set (Fig. 4(e)), where the magnitude is larger for configurations closer to the local minimum(Fig. 4(f)). For the GNN trained on both energies and forces, the situation is much better (Fig. 5(b)), although the magnitude of $\cos\theta_F^{adv}$ is still large for configurations close to the local minimum (Fig. 5(c)). Training on $\vec{F}^{\mathrm{Acc}}$ and E^{Acc} concurrently is common for ML force field, and this experiment suggests that such models may not move in their own adversarial directions. Further work is needed to see if the high $|\cos\theta_F^{adv}|$ close to the local minima is a concern for these models, and whether higher order quantities (for example, the vibrational modes) exhibit adversarial directions.

Conclusion: We have shown that the adversarial directions for approximate energy models can be aligned with their forces near local extrema. This alignment is observed for several commonly used physics and ML models (See Section 5 in the Appendix for Behler-Parrinello and graph neural networks, trained with and without forces on DFT data). We showed that this phenomenon could have drastic implications for the fidelity of physics simulations, for a particularly popular application of structural optimization. These results support the view that approximate physics models should not be evaluated by accuracy metrics on i.i.d. test sets, given their natural tendency to get out of the i.i.d. distribution.

[2]GNN implementation is available as a Colaboratory notebook at:
github.com/google/jax-md/blob/master/notebooks/neural_networks.ipynb

Broader Impact

Machine learning is gaining popularity as a modeling tool in the physical sciences. This trend is likely to continue, since scientists are having to deal with much more data than than ever before, whether the data is from experiments or from simulations. However, the reliability of these machine learning models is crucially important, if they can be of use to make scientific or technological progress. Our paper highlights a particular failure point for molecular physics models, and suggests that such models should be expected to have worse fidelity than might be expected from their performance on i.i.d. test set.

Acknowledgments and Disclosure of Funding

We are grateful to Lusann Yang for helping with the DFT simulations, and Gowoon Cheon for feedback on the manuscript and pointing out the similarity between adversarial training and the training protocol in Ref. [25]. We also thank our reviewers for valuable feedback on the manuscript.

References

[1] Gang Lu and Efthimios Kaxiras. An overview of multiscale simulations of materials. *arXiv preprint cond-mat/0401073*, 2004.

[2] Kurt Lejaeghere, Gustav Bihlmayer, Torbjörn Björkman, Peter Blaha, Stefan Blügel, Volker Blum, Damien Caliste, Ivano E Castelli, Stewart J Clark, Andrea Dal Corso, et al. Reproducibility in density functional theory calculations of solids. *Science*, 351(6280), 2016.

[3] Adri CT Van Duin, Siddharth Dasgupta, Francois Lorant, and William A Goddard. Reaxff: a reactive force field for hydrocarbons. *The Journal of Physical Chemistry A*, 105(41):9396–9409, 2001.

[4] Ferdi Aryasetiawan and Olle Gunnarsson. The gw method. *Reports on Progress in Physics*, 61 (3):237, 1998.

[5] Thomas Y Hou and Xiao-Hui Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of computational physics*, 134(1):169–189, 1997.

[6] Justin Gilmer, Ryan P Adams, Ian Goodfellow, David Andersen, and George E Dahl. Motivating the rules of the game for adversarial example research. *arXiv preprint arXiv:1807.06732*, 2018.

[7] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[8] Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles. *arXiv preprint arXiv:1707.03501*, 2017.

[9] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

[10] Jörg Behler. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of chemical physics*, 134(7):074106, 2011.

[11] Albert P Bartók, Mike C Payne, Risi Kondor, and Gábor Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical review letters*, 104(13):136403, 2010.

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

[13] Grégoire Montavon, Katja Hansen, Siamac Fazli, Matthias Rupp, Franziska Biegler, Andreas Ziehe, Alexandre Tkatchenko, Anatole V Lilienfeld, and Klaus-Robert Müller. Learning invariant representations of molecules for atomization energy prediction. In *Advances in neural information processing systems*, pages 440–448, 2012.

[14] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[15] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in neural information processing systems*, pages 991–1001, 2017.

[16] Nongnuch Artrith and Alexander Urban. An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for tio2. *Computational Materials Science*, 114:135–150, 2016.

[17] David E Shaw, Ron O Dror, John K Salmon, JP Grossman, Kenneth M Mackenzie, Joseph A Bank, Cliff Young, Martin M Deneroff, Brannon Batson, Kevin J Bowers, et al. Millisecond-scale molecular dynamics simulations on anton. In *Proceedings of the conference on high performance computing networking, storage and analysis*, pages 1–11, 2009.

[18] Chris J Pickard and RJ Needs. Ab initio random structure searching. *Journal of Physics: Condensed Matter*, 23(5):053201, 2011.

[19] Artem R Oganov, Chris J Pickard, Qiang Zhu, and Richard J Needs. Structure prediction drives materials discovery. *Nature Reviews Materials*, 4(5):331–348, 2019.

[20] BWH Van Beest, Gert Jan Kramer, and RA Van Santen. Force fields for silicas and aluminophosphates based on ab initio calculations. *Physical Review Letters*, 64(16):1955, 1990.

[21] Antoine Carré, Simona Ispas, Jürgen Horbach, and Walter Kob. Developing empirical potentials from ab initio simulations: The case of amorphous silica. *Computational Materials Science*, 124:323–334, 2016.

[22] Han Liu, Zipeng Fu, Yipeng Li, Nazreen Farina Ahmad Sabri, and Mathieu Bauchy. Machine learning forcefield for silicate glasses. *arXiv preprint arXiv:1902.03486*, 2019.

[23] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[24] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[25] Benjamin Ummenhofer, Lukas Prantl, Nils Thuerey, and Vladlen Koltun. Lagrangian fluid simulation with continuous convolutions. In *International Conference on Learning Representations*, 2019.

[26] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W Battaglia. Learning to simulate complex physics with graph networks. *arXiv preprint arXiv:2002.09405*, 2020.

[27] Nic Ford, Justin Gilmer, Nicolas Carlini, and Dogus Cubuk. Adversarial examples are a natural consequence of test error in noise. *arXiv preprint arXiv:1901.10513*, 2019.

[28] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[29] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[30] Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. *arXiv preprint arXiv:1806.01242*, 2018.

[31] Victor Bapst, Thomas Keck, A Grabska-Barwińska, Craig Donner, Ekin Dogus Cubuk, SS Schoenholz, Annette Obika, AWR Nelson, Trevor Back, Demis Hassabis, et al. Unveiling the predictive power of static structure in glassy systems. *Nature Physics*, 16(4):448–454, 2020.

[32] Samuel S Schoenholz and Ekin D Cubuk. Jax, md: End-to-end differentiable, hardware accelerated, molecular dynamics in pure python. *arXiv preprint arXiv:1912.04232*, 2019.

[33] Samuel Schoenholz and Ekin Dogus Cubuk. Jax md: A framework for differentiable physics. *Advances in Neural Information Processing Systems*, 33, 2020.

[34] Jürgen Hafner. Ab-initio simulations of materials using vasp: Density-functional theory and beyond. *Journal of computational chemistry*, 29(13):2044–2078, 2008.

[35] Nongnuch Artrith, Tobias Morawietz, and Jörg Behler. High-dimensional neural-network potentials for multicomponent systems: Applications to zinc oxide. *Physical Review B*, 83(15): 153101, 2011.

[36] Nongnuch Artrith and Jörg Behler. High-dimensional neural network potentials for metal surfaces: A prototype study for copper. *Physical Review B*, 85(4):045439, 2012.

[37] Rustam Z Khaliullin, Hagai Eshet, Thomas D Kühne, Jörg Behler, and Michele Parrinello. Graphite-diamond phase coexistence study employing a neural-network mapping of the ab initio potential energy surface. *Physical Review B*, 81(10):100103, 2010.

[38] Ekin D Cubuk, Brad D Malone, Berk Onat, Amos Waterland, and Efthimios Kaxiras. Representations in neural network based empirical potentials. *The Journal of chemical physics*, 147(2): 024104, 2017.

[39] Ruggero Lot, Franco Pellegrini, Yusuf Shaidu, and Emine Küçükbenli. Panna: Properties from artificial neural network architectures. *Computer Physics Communications*, page 107402, 2020.

[40] Kun Yao, John E Herr, David W Toth, Ryker Mckintyre, and John Parkhill. The tensormol-0.1 model chemistry: A neural network augmented with long-range physics. *Chemical science*, 9 (8):2261–2269, 2018.

[41] Atsuto Seko, Akira Takahashi, and Isao Tanaka. Sparse representation for a potential energy surface. *Physical Review B*, 90(2):024101, 2014.

[42] Jörg Behler, Roman Martoňák, Davide Donadio, and Michele Parrinello. Metadynamics simulations of the high-pressure phases of silicon employing a high-dimensional neural network potential. *Physical review letters*, 100(18):185501, 2008.

[43] Luigi Bonati and Michele Parrinello. Silicon liquid structure and crystal nucleation from ab initio deep metadynamics. *Physical review letters*, 121(26):265701, 2018.

[44] Georg Kresse and Jürgen Hafner. Ab initio molecular dynamics for liquid metals. *Physical Review B*, 47(1):558, 1993.

[45] John P Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Physical review letters*, 77(18):3865, 1996.

[46] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[47] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[48] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[49] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[50] Ekin D Cubuk, Barret Zoph, Samuel S Schoenholz, and Quoc V Le. Intriguing properties of adversarial examples. *arXiv preprint arXiv:1711.02846*, 2017.

[51] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? 2018.

[52] Samuel Dodge and Lina Karam. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th international conference on computer communication and networks (ICCCN)*, pages 1–7. IEEE, 2017.

Appendix

4 Implementation details

To investigate the presence of adversarial directions, and their overlap with the force, we experiment with standard ML architectures and physics simulation tools. All ML models were trained on a single P100 GPU. All of the machine learning experiments (BP-NN and GNN) and the physics calculations (except for DFT) were implemented within JAX-MD[3] [32, 33]. Below we summarize the tools used in this study in detail.

Quantum mechanical simulations We perform the DFT calculations using the state-of-the-art Vienna Atomistic Simulation Package (VASP) [34].

Behler-Parrinello architecture The Behler-Parrinello neural network (BP-NN) [9] is a popular architecture for learning quantum mechanical energies (for example, see [35, 36, 16, 37–40]). The BP-NN consists of hand designed features computed for each atom, α, $\phi_\alpha(\vec{R})$, that are fed into small fully-connected networks $f(\phi_\alpha(\vec{R}); \theta)$. The predictions for each atom are then summed to produce, $E^{\text{App}}(\vec{R}) = \sum_\alpha f(\phi_\alpha(\vec{R}))$. We use 2 hidden-layers and 30 hidden-nodes per layer atomic neural network, and the tanh activation function. We train using the momentum optimizer with a learning rate of 5×10^{-6} and batch size of 15, for 500 epochs. Note that this neural-network model was trained using the loss defined in Eq. 1, so the adversarial direction computed for this model is exactly the direction that maximizes the training objective.

Graph neural networks Recently, graph neural networks [28, 14, 29–31] (GNN) have emerged as an effective architecture for molecular systems. Here we construct a graph from each configuration by considering two atoms as joined if the distance between them is less than a threshold ($\sigma = 3A$). Each node is initialized to have state $n_i = 0$ and each edge is initialized to have a state equal to the displacement vector between the two atoms $\vec{r}_{\alpha\beta} = \vec{r}_\alpha - \vec{r}_\beta$ along with a global state $g = 0$. We sum messages from incoming edges to update the node state and edge state using fully-connected networks with two-hidden layers and ReLU activations. After hyperparameter optimization we find that single message passing step is optimal. We train the model for 160 epochs using ADAM optimizer, with a learning rate of 10^{-3} and a batch size of 128. The GNN is trained using the same loss as BP-NN. We utilized data augmentation of rotations and flips.

Simple analytic energy model As an example of a simple analytic energy model, we use the BKS model to simulate SiO_2. The BKS model is a Buckingham-like potential [20–22], commonly used for studying silicate glasses. The Buckingham form can be defined as:

$$U_{\alpha\beta} = \frac{q_\alpha q_\beta}{4\pi\epsilon_0 r_{\alpha\beta}} + A_{\alpha\beta}\exp\left(-\frac{r_{\alpha\beta}}{\rho_{\alpha\beta}}\right) - \frac{C_{\alpha\beta}}{r_{\alpha\beta}^6} + \frac{D_{\alpha\beta}}{r_{\alpha\beta}^{24}}, \tag{6}$$

where $\vec{r}_{\alpha\beta}$ and $U_{\alpha\beta}$ are the pairwise distance and energy between atoms α and β, q_α is the partial charge of atom α, ϵ_0 is the dielectric constant, and $A_{\alpha\beta}, \rho_{\alpha\beta}, C_{\alpha\beta}$, and $D_{\alpha\beta}$ are parameters of the approximate energy model. For the SiO_2 system there are only two types of atoms, which leads to a total of 12 parameters for curve-fitting (for example 3 real parameters are used for $A_{\alpha\beta}$: $A_{Si-Si}, A_{Si-O}, A_{O-O}$. Total energy of an atomistic configuration is then given by the sum of all pairwise energies $U_{\alpha\beta}$.

The approximate energy model defined by Eq. 6 is very different from ML models: it only has 12 parameters, and the parameters are not optimized using a well defined training set and gradient descent. Analytically tractable approximate energy models of this form are often constructed from expert knowledge of scientists who are extremely familiar with the physical system in question. The parameters are most often optimized so that simulations of the approximate energy model agree with various experimental measurements.

Physical systems We use two prototypical molecular systems for our analysis. First we consider silicon dioxide, SiO_2, which is in glass all around us, from cell-phones to windows. Our SiO_2 system has 96 atoms in its unit cell (32 Si atoms and 64 O atoms). We consider DFT, BKS, and BP-NN energy models on this system. Atomistic configurations are sampled by randomly displacing atoms

[3] Example code is available as a Colaboratory notebook at:
github.com/google/jax-md/blob/master/notebooks/neural_networks.ipynb

of equilibrium structures in random directions for training and test data, which is a commonly used sampling method [37, 41].

The other system we consider is silicon (Si) at different temperatures, as simulated by DFT molecular dynamics. This is an extremely popular system in empirical potentials research that has received considerable attention in the literature [9, 42, 43, 38]. We construct a dataset using another canonical sampling method [10, 15] where we sample quantum mechanical molecular dynamics of 64 atoms at several temperatures (300K, 600K, 900K, and 2000K). Configurations from these trajectories and their corresponding energies are then uniformly sampled to construct training and test sets. The total number of configurations collected from DFT molecular dynamics is around 56k, 36k of which was used for training and validation, and 20k was used for testing. We train a BP-NN and GNN to approximate the energies of these Si configurations. Note that for both of the Si and the SiO_2 systems, the training and test data are from an i.i.d. distribution.

To investigate the presence of adversarial directions, and their overlap with the force, we experiment with standard ML architectures and physics simulation tools. All ML models were trained on a single NVIDIA TESLA P100 GPU. Behler-Parrinello neural network trained in less than an hour, and the graph neural network trained in four hours.

When sampling training and test configurations by randomly displacing atoms, we sample distortions from an independent Gaussian distribution in each dimension for each atom, and normalize the L2-norm of the total distortion. When we calculated the average cosine-angle magnitude of configurations as a function of energy above local minimum, we used a bin-size of 50 samples.

Quantum mechanical simulations DFT calculations used a projector-augmented wave (PAW) potential [44]. Exchange-correlation functional employed was by Perdew et al. [45]. Energy cutoff was 300 eV, and k-point mesh convergence was 0.5 meV/atom. Stress convergence was 0.1 kbar. DFT calculations took about 6 minutes per sample on 32 CPUs.

Estimating the error due to high $\cos\,\theta_F^{adv}$ In simulating these molecular simulations it is commonplace to use Newton's laws so that the configuration is governed by $\vec{F} = m\ddot{\vec{R}}$ where m is the atomic mass and $\ddot{\vec{R}}$ is the second time-derivative of positions. In atomistic simulations, a commonly used time step is 1 fs (10^{-15} seconds), which we will use in this section. To evaluate the error due to the alignment of force and adversarial direction, we will take one simulation step in the direction of the $\vec{F}^{\mathrm{App}}$, by moving the atoms for 1 fs with acceleration $\vec{F}^{\mathrm{App}}/m$. Note that this is similar to a single step adversarial attack, where the distortion size is determined by Newton's laws. One main difference with adversarial attacks is that instead of using the adversarial direction commonly used in vision research, we will use $\vec{F}^{\mathrm{App}}$ which is naturally used in physical simulations to evolve the system.

5 Experiments

We now investigate the presence of adversarial directions, and their overlap with the force, for range of choices for E^{Acc} and E^{App}. After training the approximate energy model, E^{App}, we compute its adversarial direction $\vec{\mathcal{A}}(\vec{R}_j)$ using Eq. (3). To evaluate how aligned the adversarial direction is with the force, $\vec{F}^{\mathrm{App}}(\vec{R}_j)$, we compute the cosine-angle between the two vectors,

$$\cos\,\theta_F^{adv} = \frac{\vec{\mathcal{A}}(\vec{R}) \cdot \vec{F}^{\mathrm{App}}(\vec{R})}{||\vec{\mathcal{A}}(\vec{R})||\;||\vec{F}^{\mathrm{App}}(\vec{R})||}. \tag{7}$$

Note that both $\vec{\mathcal{A}}(\vec{R}_j)$ and $\vec{F}^{\mathrm{App}}$ are 3N-dimensional vectors where N is the number of atoms along with three spatial directions for each atom. Since our systems have 50 to 100 atoms, the cosine-angle between two random vectors is close to 0. For each system studied we evaluate the error of the model and the distribution of cosine-angles. We will see that even well-performing models typically have high alignment with adversarial directions. Moreover, we will find that the sign of the overlap is random; this agrees with the observation that the constant C in Eq. (4) will have a random sign. Additionally, we recall that Eq. (4) will be accurate when the system is near a minimum of the energy. To validate this approximation and to show that adversarial examples become particularly problematic near extrema, we also evaluate the average magnitude of the cosine-angle, $\cos\,\theta_F^{adv}$, as a function of the energy above the local minimum, ΔE.

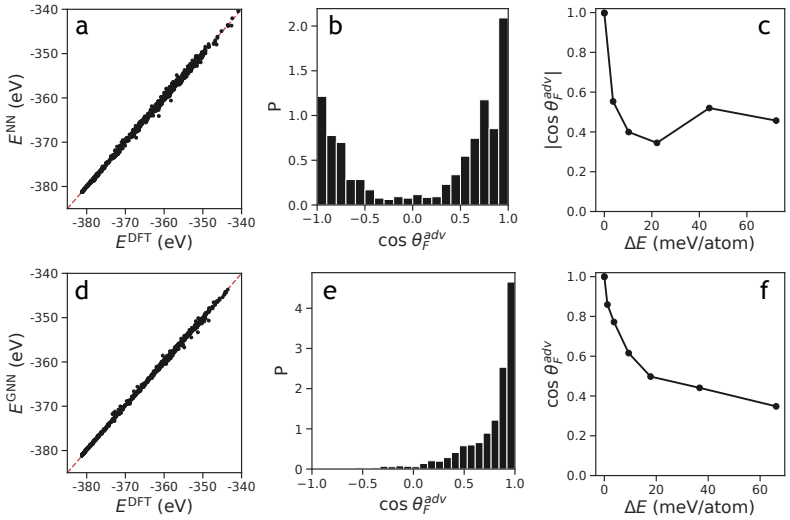

Figure 4: **Neural networks trained on DFT have adversarial directions aligned with their forces.** BP-NN and GNN is trained on DFT calculations of Si atoms. **a**) Points represent the energy prediction of the the labels (E^{DFT}) vs. the neural network predictions for BP-NN **(a)** and GNN **(d)**. Red dashed lines denote perfect agreement. Middle column shows the histogram density of $\cos\,\theta_F^{adv}$ for BP-NN **(b)** and GNN **(e)**. Right column is the average magnitude of $\cos\,\theta_F^{adv}$ vs. ΔE, the energy above local minimum, for BP-NN **(d)** and GNN **(f)**.

5.1 Neural networks approximating DFT

Next, we study the practically relevant task of fitting neural networks on DFT energies. First, we train the BP-NN on Si configurations (sampled from molecular dynamics, as described in Section 4) to achieve a test error of 4.9 meV/atom, which is comparable to the reported error of 5-6 meV/atom on this system by Behler and Parrinello [9]. Fig. 4(a) shows that the prediction of the BP-NN agree well with the labels on test configurations, however the magnitude of $\cos\,\theta_F^{adv}$ is large for most configurations (panel (b)). We see that for a significant fraction of the configurations, the adversarial direction and the $\vec{F}^{\mathrm{App}}$ is perfectly aligned. Fig. 4(c) shows that, as expected, problematic configurations are found very close to the local minima of the DFT landscape (E^{Acc}). The magnitude of overlap is smaller for configurations with higher energy.

Next, we train a GNN to see if better architectures have different behavior with respect to adversarial examples. In Fig. 4(d) we see that the energy predictions of the GNN are more accurate than the Behler-Parrinello architecture (4.5 meV/atom error vs. 4.9 meV/atom error). However, despite the improved accuracy, we see that for a majority of the configurations, the adversarial direction is approximately aligned with the GNN force (panel (e)). The alignment is again strongest for configurations close to the local minimum, and it goes down with increasing energy (Fig. 4(f)).

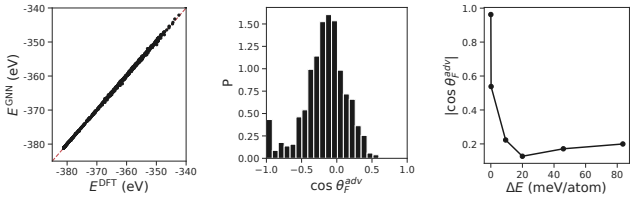

Figure 5: **a)** E^{GNN} vs. (E^{DFT}) for the Si system (GNN is trained on forces). **b)**: Histogram density of $\cos\,\theta_F^{adv}$ for test configurations. **c)** Average magnitude of $\cos\,\theta_F^{adv}$ vs. ΔE, the energy above the local minimum.

Next, we train a GNN on energies *and* forces from DFT (see Fig. 5). We see that while training on forces certainly helps, which is good news, configurations close to the local minimum still have large magnitude for $\cos\,\theta_F^{adv}$.

6 Related Work

Most of the initial research into training ML models as approximate energy functions focused on simple neural networks [9, 16] or Gaussian proccesses [11]. More recent work has utilized ideas from graph convolutional networks and message passing neural networks [28, 15, 14, 30].

To our knowledge, adversarial examples or adversarial directions have not been investigated in physics models before. Adversarial examples have been extensively studied in the deep learning field of vision [46, 23, 24, 47–50], as described in the Introduction. Performance of deep learning models was also shown to deteriorate if images are slightly changed by non-adversarial distortions such as translations, blurring, and contrast changes [51, 52].

Appendix

4 Implementation details

To investigate the presence of adversarial directions, and their overlap with the force, we experiment with standard ML architectures and physics simulation tools. All ML models were trained on a single P100 GPU. All of the machine learning experiments (BP-NN and GNN) and the physics calculations (except for DFT) were implemented within JAX-MD[3] [32, 33]. Below we summarize the tools used in this study in detail.

Quantum mechanical simulations We perform the DFT calculations using the state-of-the-art Vienna Atomistic Simulation Package (VASP) [34].

Behler-Parrinello architecture The Behler-Parrinello neural network (BP-NN) [9] is a popular architecture for learning quantum mechanical energies (for example, see [35, 36, 16, 37–40]). The BP-NN consists of hand designed features computed for each atom, α, $\phi_\alpha(\vec{R})$, that are fed into small fully-connected networks $f(\phi_\alpha(\vec{R}); \theta)$. The predictions for each atom are then summed to produce, $E^{\text{App}}(\vec{R}) = \sum_\alpha f(\phi_\alpha(\vec{R}))$. We use 2 hidden-layers and 30 hidden-nodes per layer atomic neural network, and the tanh activation function. We train using the momentum optimizer with a learning rate of 5×10^{-6} and batch size of 15, for 500 epochs. Note that this neural-network model was trained using the loss defined in Eq. 1, so the adversarial direction computed for this model is exactly the direction that maximizes the training objective.

Graph neural networks Recently, graph neural networks [28, 14, 29–31] (GNN) have emerged as an effective architecture for molecular systems. Here we construct a graph from each configuration by considering two atoms as joined if the distance between them is less than a threshold ($\sigma = 3A$). Each node is initialized to have state $n_i = 0$ and each edge is initialized to have a state equal to the displacement vector between the two atoms $\vec{r}_{\alpha\beta} = \vec{r}_\alpha - \vec{r}_\beta$ along with a global state $g = 0$. We sum messages from incoming edges to update the node state and edge state using fully-connected networks with two-hidden layers and ReLU activations. After hyperparameter optimization we find that single message passing step is optimal. We train the model for 160 epochs using ADAM optimizer, with a learning rate of 10^{-3} and a batch size of 128. The GNN is trained using the same loss as BP-NN. We utilized data augmentation of rotations and flips.

Simple analytic energy model As an example of a simple analytic energy model, we use the BKS model to simulate SiO_2. The BKS model is a Buckingham-like potential [20–22], commonly used for studying silicate glasses. The Buckingham form can be defined as:

$$U_{\alpha\beta} = \frac{q_\alpha q_\beta}{4\pi\epsilon_0 r_{\alpha\beta}} + A_{\alpha\beta}\exp\left(-\frac{r_{\alpha\beta}}{\rho_{\alpha\beta}}\right) - \frac{C_{\alpha\beta}}{r_{\alpha\beta}^6} + \frac{D_{\alpha\beta}}{r_{\alpha\beta}^{24}}, \tag{6}$$

where $\vec{r}_{\alpha\beta}$ and $U_{\alpha\beta}$ are the pairwise distance and energy between atoms α and β, q_α is the partial charge of atom α, ϵ_0 is the dielectric constant, and $A_{\alpha\beta}, \rho_{\alpha\beta}, C_{\alpha\beta}$, and $D_{\alpha\beta}$ are parameters of the approximate energy model. For the SiO_2 system there are only two types of atoms, which leads to a total of 12 parameters for curve-fitting (for example 3 real parameters are used for $A_{\alpha\beta}$: $A_{Si-Si}, A_{Si-O}, A_{O-O}$. Total energy of an atomistic configuration is then given by the sum of all pairwise energies $U_{\alpha\beta}$.

The approximate energy model defined by Eq. 6 is very different from ML models: it only has 12 parameters, and the parameters are not optimized using a well defined training set and gradient descent. Analytically tractable approximate energy models of this form are often constructed from expert knowledge of scientists who are extremely familiar with the physical system in question. The parameters are most often optimized so that simulations of the approximate energy model agree with various experimental measurements.

Physical systems We use two prototypical molecular systems for our analysis. First we consider silicon dioxide, SiO_2, which is in glass all around us, from cell-phones to windows. Our SiO_2 system has 96 atoms in its unit cell (32 Si atoms and 64 O atoms). We consider DFT, BKS, and BP-NN energy models on this system. Atomistic configurations are sampled by randomly displacing atoms

[3] Example code is available as a Colaboratory notebook at:
github.com/google/jax-md/blob/master/notebooks/neural_networks.ipynb

of equilibrium structures in random directions for training and test data, which is a commonly used sampling method [37, 41].

The other system we consider is silicon (Si) at different temperatures, as simulated by DFT molecular dynamics. This is an extremely popular system in empirical potentials research that has received considerable attention in the literature [9, 42, 43, 38]. We construct a dataset using another canonical sampling method [10, 15] where we sample quantum mechanical molecular dynamics of 64 atoms at several temperatures (300K, 600K, 900K, and 2000K). Configurations from these trajectories and their corresponding energies are then uniformly sampled to construct training and test sets. The total number of configurations collected from DFT molecular dynamics is around 56k, 36k of which was used for training and validation, and 20k was used for testing. We train a BP-NN and GNN to approximate the energies of these Si configurations. Note that for both of the Si and the SiO_2 systems, the training and test data are from an i.i.d. distribution.

To investigate the presence of adversarial directions, and their overlap with the force, we experiment with standard ML architectures and physics simulation tools. All ML models were trained on a single NVIDIA TESLA P100 GPU. Behler-Parrinello neural network trained in less than an hour, and the graph neural network trained in four hours.

When sampling training and test configurations by randomly displacing atoms, we sample distortions from an independent Gaussian distribution in each dimension for each atom, and normalize the L2-norm of the total distortion. When we calculated the average cosine-angle magnitude of configurations as a function of energy above local minimum, we used a bin-size of 50 samples.

Quantum mechanical simulations DFT calculations used a projector-augmented wave (PAW) potential [44]. Exchange-correlation functional employed was by Perdew et al. [45]. Energy cutoff was 300 eV, and k-point mesh convergence was 0.5 meV/atom. Stress convergence was 0.1 kbar. DFT calculations took about 6 minutes per sample on 32 CPUs.

Estimating the error due to high $\cos\,\theta_F^{adv}$ In simulating these molecular simulations it is commonplace to use Newton's laws so that the configuration is governed by $\vec{F} = m\ddot{\vec{R}}$ where m is the atomic mass and $\ddot{\vec{R}}$ is the second time-derivative of positions. In atomistic simulations, a commonly used time step is 1 fs (10^{-15} seconds), which we will use in this section. To evaluate the error due to the alignment of force and adversarial direction, we will take one simulation step in the direction of the $\vec{F}^{\mathrm{App}}$, by moving the atoms for 1 fs with acceleration $\vec{F}^{\mathrm{App}}/m$. Note that this is similar to a single step adversarial attack, where the distortion size is determined by Newton's laws. One main difference with adversarial attacks is that instead of using the adversarial direction commonly used in vision research, we will use $\vec{F}^{\mathrm{App}}$ which is naturally used in physical simulations to evolve the system.

5 Experiments

We now investigate the presence of adversarial directions, and their overlap with the force, for range of choices for E^{Acc} and E^{App}. After training the approximate energy model, E^{App}, we compute its adversarial direction $\vec{\mathcal{A}}(\vec{R}_j)$ using Eq. (3). To evaluate how aligned the adversarial direction is with the force, $\vec{F}^{\mathrm{App}}(\vec{R}_j)$, we compute the cosine-angle between the two vectors,

$$\cos\,\theta_F^{adv} = \frac{\vec{\mathcal{A}}(\vec{R}) \cdot \vec{F}^{\mathrm{App}}(\vec{R})}{||\vec{\mathcal{A}}(\vec{R})||\;||\vec{F}^{\mathrm{App}}(\vec{R})||}. \tag{7}$$

Note that both $\vec{\mathcal{A}}(\vec{R}_j)$ and $\vec{F}^{\mathrm{App}}$ are 3N-dimensional vectors where N is the number of atoms along with three spatial directions for each atom. Since our systems have 50 to 100 atoms, the cosine-angle between two random vectors is close to 0. For each system studied we evaluate the error of the model and the distribution of cosine-angles. We will see that even well-performing models typically have high alignment with adversarial directions. Moreover, we will find that the sign of the overlap is random; this agrees with the observation that the constant C in Eq. (4) will have a random sign. Additionally, we recall that Eq. (4) will be accurate when the system is near a minimum of the energy. To validate this approximation and to show that adversarial examples become particularly problematic near extrema, we also evaluate the average magnitude of the cosine-angle, $\cos\,\theta_F^{adv}$, as a function of the energy above the local minimum, ΔE.

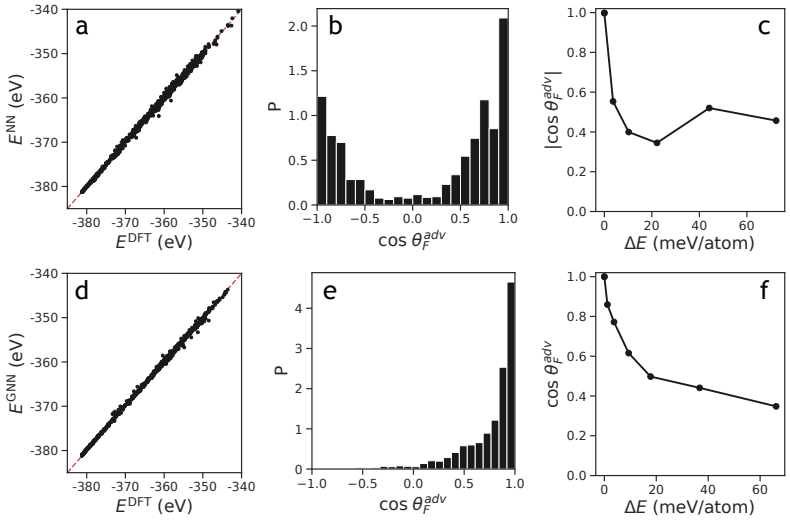

Figure 4: **Neural networks trained on DFT have adversarial directions aligned with their forces.** BP-NN and GNN is trained on DFT calculations of Si atoms. **a**) Points represent the energy prediction of the the labels (E^{DFT}) vs. the neural network predictions for BP-NN **(a)** and GNN **(d)**. Red dashed lines denote perfect agreement. Middle column shows the histogram density of $\cos\,\theta_F^{adv}$ for BP-NN **(b)** and GNN **(e)**. Right column is the average magnitude of $\cos\,\theta_F^{adv}$ vs. ΔE, the energy above local minimum, for BP-NN **(d)** and GNN **(f)**.

5.1 Neural networks approximating DFT

Next, we study the practically relevant task of fitting neural networks on DFT energies. First, we train the BP-NN on Si configurations (sampled from molecular dynamics, as described in Section 4) to achieve a test error of 4.9 meV/atom, which is comparable to the reported error of 5-6 meV/atom on this system by Behler and Parrinello [9]. Fig. 4(a) shows that the prediction of the BP-NN agree well with the labels on test configurations, however the magnitude of $\cos\,\theta_F^{adv}$ is large for most configurations (panel (b)). We see that for a significant fraction of the configurations, the adversarial direction and the $\vec{F}^{\text{App}}$ is perfectly aligned. Fig. 4(c) shows that, as expected, problematic configurations are found very close to the local minima of the DFT landscape (E^{Acc}). The magnitude of overlap is smaller for configurations with higher energy.

Next, we train a GNN to see if better architectures have different behavior with respect to adversarial examples. In Fig. 4(d) we see that the energy predictions of the GNN are more accurate than the Behler-Parrinello architecture (4.5 meV/atom error vs. 4.9 meV/atom error). However, despite the improved accuracy, we see that for a majority of the configurations, the adversarial direction is approximately aligned with the GNN force (panel (e)). The alignment is again strongest for configurations close to the local minimum, and it goes down with increasing energy (Fig. 4(f)).

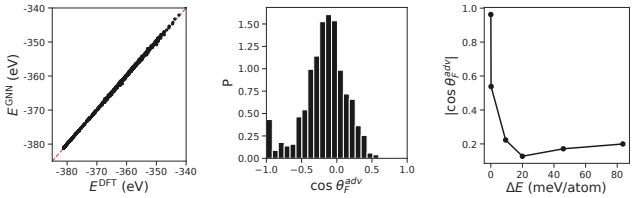

Figure 5: **a)** E^{GNN} vs. (E^{DFT}) for the Si system (GNN is trained on forces). **b)**: Histogram density of $\cos\,\theta_F^{adv}$ for test configurations. **c)** Average magnitude of $\cos\,\theta_F^{adv}$ vs. ΔE, the energy above the local minimum.

Next, we train a GNN on energies *and* forces from DFT (see Fig. 5). We see that while training on forces certainly helps, which is good news, configurations close to the local minimum still have large magnitude for $\cos\,\theta_F^{adv}$.

6 Related Work

Most of the initial research into training ML models as approximate energy functions focused on simple neural networks [9, 16] or Gaussian proccesses [11]. More recent work has utilized ideas from graph convolutional networks and message passing neural networks [28, 15, 14, 30].

To our knowledge, adversarial examples or adversarial directions have not been investigated in physics models before. Adversarial examples have been extensively studied in the deep learning field of vision [46, 23, 24, 47–50], as described in the Introduction. Performance of deep learning models was also shown to deteriorate if images are slightly changed by non-adversarial distortions such as translations, blurring, and contrast changes [51, 52].