

Neural scaling of deep chemical models

Received: 14 June 2022

Accepted: 15 September 2023

Published online: 23 October 2023

**Nathan C. Frey**^{1,6}✉, **Ryan Soklaski**^{1,7}, **Simon Axelrod**^{2,3}, **Siddharth Samsi**¹, **Rafael Gómez-Bombarelli**², **Connor W. Coley**^{4,5} & **Vijay Gadepally**¹

Massive scale, in terms of both data availability and computation, enables important breakthroughs in key application areas of deep learning such as natural language processing and computer vision. There is emerging evidence that scale may be a key ingredient in scientific deep learning, but the importance of physical priors in scientific domains makes the strategies and benefits of scaling uncertain. Here we investigate neural-scaling behaviour in large chemical models by varying model and dataset sizes over many orders of magnitude, studying models with over one billion parameters, pre-trained on datasets of up to ten million datapoints. We consider large language models for generative chemistry and graph neural networks for machine-learned interatomic potentials. We investigate the interplay between physical priors and scale and discover empirical neural-scaling relations for language models in chemistry with a scaling exponent of 0.17 for the largest dataset size considered, and a scaling exponent of 0.26 for equivariant graph neural network interatomic potentials.

The ‘unreasonable effectiveness’ of deep learning¹ in domains such as computer vision and natural language processing (NLP) relies on the ability of deep neural networks to leverage ever-increasing amounts of compute, data and model capacity. Large-scale models, including Bidirectional Encoder Representations from Transformers (BERT)² and DALL-E³, have been so successful at synthesizing information from large datasets via self-supervised pre-training and performing a variety of downstream tasks with little to no fine-tuning that most state-of-the-art models in NLP and computer vision are adapted from a small set of large, pre-trained models⁴. Naturally, we might expect that massive model and dataset scaling will be a prerequisite to achieving out-sized success for deep learning in science. Recent work such as AlphaFold⁵, the Open Catalyst Project^{6,7} and ChemBERTa⁸ indicates that larger datasets and models, pre-training and self-supervised learning—all key ingredients in computer vision and NLP—unlock new capabilities for deep learning in chemistry. However, unlike in computer vision and NLP, the path to scaling deep chemical networks and the potential benefits are unclear. Chemical deep learning can incorporate physics-based

priors that may ameliorate the steep resource requirements seen in other fields^{9–12}. Moreover, because of the heterogeneity and complexity of chemical space¹³ and molecular machine learning tasks^{14,15}, training general and robust models that perform well on a wide variety of downstream tasks remains a pressing challenge^{8,16,17}. The enormity of chemical space and heterogeneity of these tasks motivates investigations of large-scale models in chemistry, because such models are well suited to unlabelled, multi-modal datasets^{3,4}. Recently, neural-scaling laws^{18,19} have emerged as a way to characterize the striking trends of improved model performance over many orders of magnitude with respect to model size, dataset size and compute; however, these experiments require immense computational resources and rely on well-known, domain-specific model training procedures that do not apply outside of traditional deep learning application areas.

With the inordinate costs of developing and deploying large models²⁰, it is difficult to investigate neural-scaling behaviours of scientific deep learning models, which require expensive hyperparameter optimization (HPO) and experimentation. Architectures and

¹Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA, USA. ²Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA. ³Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA, USA. ⁴Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA. ⁵Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA. ⁶Present address: Prescient Design, Genentech, New York, NY, USA. ⁷Present address: Anthropic, San Francisco, CA, USA. ✉e-mail: freyn6@gene.com

hyperparameters that work well for small models and small datasets do not transfer to larger scales²¹. This presents a risk that scientific deep learning will become increasingly inaccessible as resource demands increase. Techniques for accelerating neural architecture search and hyperparameter transfer such as training speed estimation (TSE)²² and μ Transfer²¹ could accelerate the development of large-scale scientific deep learning models, where rapid advances in architecture design and complex data manifolds prevent the easy transfer of parameters and settings used in computer vision and NLP. To investigate the capabilities of deep chemical models across resource scales, practical and principled approaches are needed to accelerate hyperparameter transfer and characterize neural scaling.

In this Article, we develop strategies for scaling deep chemical models and investigate neural-scaling behaviour in large language models (LLMs) for generative chemical modelling and graph neural networks (GNNs) for machine-learned interatomic potentials. We introduce ChemGPT, a generative pre-trained transformer for autoregressive language modelling of small molecules. We train ChemGPT models with over 1 billion parameters, using datasets of up to 10 million unique molecules. We also examine large, invariant and equivariant GNNs trained on trajectories from molecular dynamics and investigate how physics-based priors affect scaling behaviour. To overcome the challenges of hyperparameter tuning at scale in new domains, we extend techniques for accelerating neural architecture search to reduce total time and compute budgets by up to 90% during HPO and neural architecture selection. We identify trends in chemical model scaling with respect to model capacity and dataset size, and show the pre-training loss performance improvements seen with increasing scale. Work concurrent with and following the original appearance of this paper has shown a wide range of performance on molecular property prediction tasks¹⁴ using pre-trained chemical language models^{23,24}, from state-of-the-art to negligible or even negative performance. New research directions involve understanding the limitations of pre-trained representations²⁵ from models including ChemGPT. Similarly, following the original appearance of our work, scaling in GNNs has shown immense success for chemical and biological systems²⁶. Our core contribution is the discovery of neural-scaling laws across extremely diverse domains of chemical deep learning: language models and neural interatomic potentials. Our results provide motivation and practical guidance for scaling studies in scientific deep learning, as well as many fruitful new research directions at the intersection of massive scale and physics-informed deep learning.

Results

In this section, we describe our main results and the workflow developed in this paper, summarized graphically in Fig. 1.

Accelerated hyperparameter optimization

To conduct extensive scaling experiments, we first need to find reasonable hyperparameters and training settings. Unlike for NLP and computer vision, there are no default model architectures, datasets, tasks, hyperparameter settings or training settings for large-scale chemical deep learning. Simply transferring empirical results from other deep learning domains or smaller-scale experiments will lead to suboptimal results²¹. Whereas large models and datasets are standard in traditional deep learning application areas, to investigate scaling in deep chemical models we must lay the groundwork for large-scale experiments. To this end, we first tackle the problem of accelerating HPO in general settings, for new model architectures, heterogeneous datasets and at scales that have not been previously investigated.

Figure 2 shows the results of training performance estimation (TPE) for ChemGPT models trained on 2 million molecules from the Molecular Sets (MOSES)²⁷ dataset. MOSES is smaller than PubChem and is representative of datasets on which chemical generative models are typically trained^{27,28}. Here we use MOSES to demonstrate how optimal

settings for a chemical LLM such as ChemGPT are quickly discovered using TPE. To enable scaling experiments, we are mainly concerned with settings related to the learning dynamics (for example, batch size and learning rate), that will impact large-scale training and fluctuate depending on the type of model and the characteristics of the dataset. To demonstrate the effectiveness of TPE, we initialize ChemGPT with the default learning rate and batch size for causal language modelling in HuggingFace. We then vary the learning rate and batch size and train models with different hyperparameters for 50 epochs. Figure 2 shows the true loss after 50 epochs versus the predicted loss using TPE after only 10 epochs. $R^2 = 0.98$ for the linear regression (equation (8)), and Spearman's rank correlation $\rho = 1.0$. With only 20% of the total training budget, we are able to identify model configurations that outperform the default settings from HuggingFace. The procedure is easily repeatable for new datasets and enables accelerated HPO.

While training procedures for LLMs such as ChemGPT are well established, scaling neural force fields (NFFs) to larger datasets and more expressive models requires new, scalable training procedures¹⁷. Large-batch training through data parallelism is one method for accelerating training, but there are known limitations and correct batch sizes vary widely for different domains²⁹. This problem is particularly acute for NFFs, where each datapoint actually contains $3N + 1$ labels for energies and atomic forces, where N is the number of atoms, creating a large effective batch size with large variance within each mini-batch. Hence, it has been observed that small batch sizes (even mini-batches of 1) work well across different NFF architectures³⁰. TPE provides a method for quickly evaluating the speed–accuracy trade off for different combinations of batch size and learning rate, which are interdependent and must be varied together to enable large-batch training.

TPE performs equally well for GNNs. We repeat the TPE procedure, varying the learning rate and batch size, for SchNet, Polarizable Atom Interaction Neural Network (PaiNN) and SpookyNet, training on 10,000 frames (1,000 frames per molecule) from the revised MD-17 (100,000 structures of molecules are taken from the original MD17 dataset by ref. 31, with energies and forces recalculated at the PBE/def2-SVP level of theory)³² dataset of 10 small organic molecules. Using only 20% of the total training budget, we achieve excellent predictive power (Fig. 3) with TPE for SchNet and PaiNN. The variance in model loss using the entire training budget is important, indicating the importance of proper HPO.

Because SpookyNet is a complex architecture that includes non-local interactions and empirical corrections, it shows slow convergence and the training speed is less correlated with the converged model loss compared with SchNet and PaiNN. However, the rank ordering of model configurations for SpookyNet from TPE is still robust (Spearman's $\rho = 0.92$), which allows for discarding non-optimal model configurations early in training, representing notable computational savings. The goodness-of-fit metrics for linear regressions using TPE are given in Table 1.

Neural scaling quantifies the improvements in loss

Next, with a strategy in place to efficiently scale up experiments using TPE, we investigate neural scaling in ChemGPT and NFFs. For each model, we perform TPE to identify good hyperparameter choices that are predicted to perform well over a range of model and dataset sizes. Then, we systematically vary the dataset size (d) and model size (m) and perform exhaustive experiments to determine the converged loss, $L(m, d)$. For efficiency and to isolate scaling behaviour, we fix hyperparameters from TPE as m and d are varied, but strictly speaking the optimal hyperparameters will change as m and d vary²¹. Due to computational resource limitations, we train ChemGPT models for a fixed number of epochs (ten) to determine the loss.

Figure 4 shows the pre-training loss as a function of model and dataset size over many orders of magnitude. Models are trained in a self-supervised, causal language modelling setting and evaluated

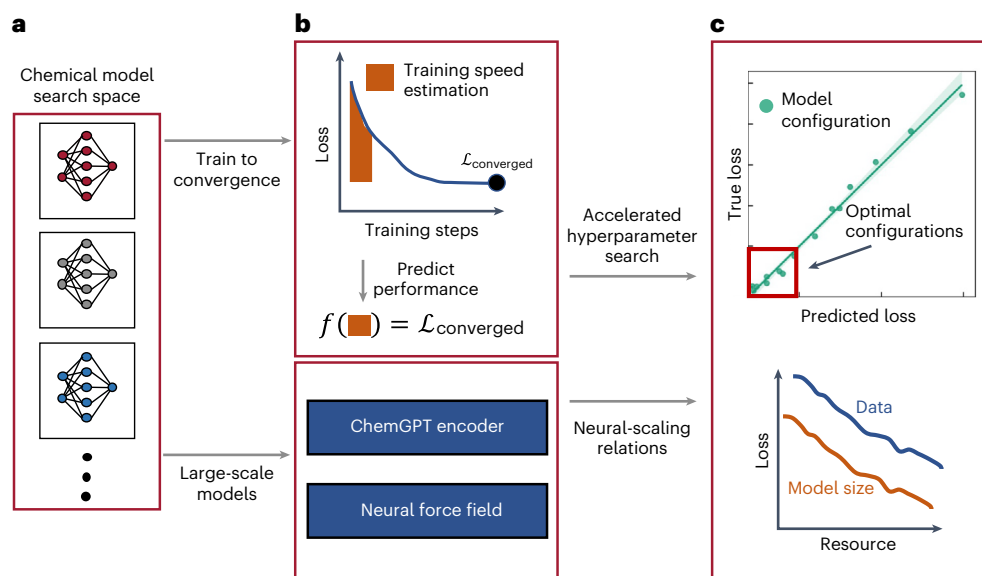


Fig. 1 | Discovery of neural-scaling relations for deep chemical models. **a, b.** Over a domain of model candidates (**a**), final, converged model loss is predicted from only a few initial epochs of training for large-scale models (**b**). **c.** Non-optimal model architectures and hyperparameter configurations

are identified early in training, allowing for efficient selection of the ideal architecture and hyperparameters. The model with the best hyperparameters is then trained with varying model and dataset sizes to discover neural-scaling relations.

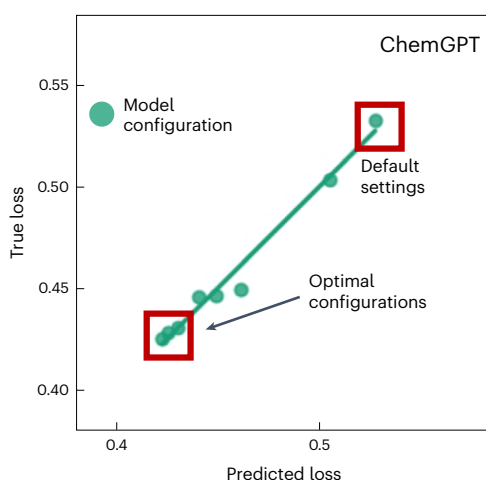


Fig. 2 | Optimal models are identified early in training with TPE and training of non-optimal models is stopped to save 80%+ total compute consumption. ChemGPT final validation loss (cross-entropy for causal language modelling) predicted from 20% of training budget using TPE. Model configurations are determined through a grid search of different batch sizes and learning rates. Models are trained on two million molecules from MOSES.

on next-token prediction for a fixed validation set. Surprisingly, no limitations in loss improvement are seen with increasing scale. The pre-training loss monotonically improves with increasing dataset size up to nearly 10 million molecules. Furthermore, for a fixed data budget, increasing model size provides monotonic improvements to the pre-training loss until the model reaches 1 billion+ non-embedding parameters. This indicates that even for small datasets, much larger models than were previously considered for deep generative modelling^{28,33} may be useful for pre-training. For the largest dataset considered here, diminishing returns to loss improvements are seen for models above 100 million non-embedding parameters. Interestingly, greater loss improvements are seen with increasing model sizes for smaller datasets than larger ones. For the largest dataset considered,

model loss saturates quickly beyond 100 million parameters. However, for the smallest dataset considered, the loss plateaus for model sizes between 10^5 and 10^7 parameters and then improves considerably. This indicates that for a fixed, small pre-training data budget, notable improvements in the pre-training loss are possible simply by scaling up the model size. Irrespective of model size, increasing dataset size provides continuous improvements to the loss with no evidence of diminishing returns for the dataset sizes considered here.

Depending on the dataset size, regimes of power-law-like scaling behaviour are seen for different ranges of model sizes. Power-law scaling is graphically identifiable as an approximately straight line fit of loss versus model size on a log-log plot. For larger datasets, power-law scaling is observed for smaller model sizes. For example, the largest dataset shows approximate power-law scaling for models between 10^5 and 10^7 non-embedding parameters (Supplementary Fig. 1). Conversely, for smaller datasets, power-law scaling is observed for larger models and over a more limited range of model sizes. The smallest dataset shows approximate power-law scaling for models between 10^7 and 10^8 non-embedding parameters (not shown).

The breakdown in power-law scaling is indicative of 'resolution limited' neural scaling³⁴, where the model is sufficiently large but the dataset is not, or vice versa. Identifying these resolution-limited regimes from the neural-scaling relations allows us to understand in general terms whether model loss improvements are limited by data availability or model capacity. The scaling exponent β is equal to 0.17 ± 0.01 for the largest dataset (Supplementary Fig. 1), after discarding the three largest models from the power-law fit. $\beta = 0.30 \pm 0.01$ for the next largest dataset (Supplementary Fig. 2). The scaling exponent quantifies the loss improvements due to increasing model size, for a fixed data budget. A larger value of β corresponds to a steeper slope and better performance with increasing data/model size. The breakdown in power-law scaling is reflective of so-called broken neural-scaling laws³⁵, which indicate that smoothly broken power-law functional forms are more general descriptions of neural-scaling behaviour.

GNNs exhibit robust neural-scaling behaviour

The potential benefits of large-scale GNNs are less clear than for LLMs, as are the relevant parameters to vary, due to the inequivalence of

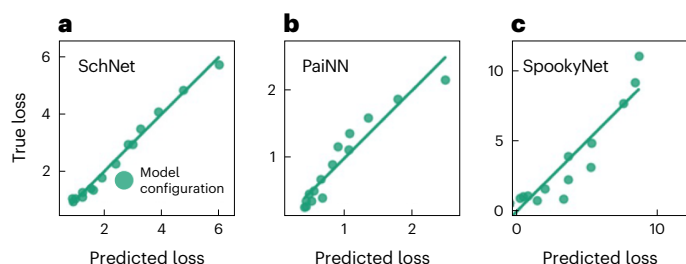


Fig. 3 | Optimal models are identified early in training with TPE and training of non-optimal models is stopped to save 80%+ total compute consumption. a–c, NFF (SchNet (a), PaiNN (b) and SpookyNet (c)) model performance—measured via equation (5)—predicted from $\leq 20\%$ of the training budget using TPE. Model configurations are determined through a grid search of different batch sizes and learning rates. Models are trained on 10,000 frames from the revised MD-17 dataset.

Table 1 | Goodness-of-fit metrics for GNN model performance using linear regression from TPE

Model	R^2	Spearman's ρ
SchNet	0.99	0.99
PaiNN	0.91	0.97
SpookyNet	0.86	0.92

depth and width for GNNs³⁶ and additional parameters beyond notions of model size that impact performance, for example, nearest-neighbour cut-off in graph construction. To simplify GNN scaling experiments, here we vary GNN capacity (depth $\times$ width) by systematically changing network width and the number of convolutions (depth). We train GNNs to predict atomic forces from the ANI-1x dataset (5 million density functional theory calculations of small organic molecules)³⁷, the largest publicly available dataset of energies and forces for small molecules. NFF models are trained with a learning rate scheduler that reduces the learning rate every 50 epochs without improvement in the validation loss, until the learning rate reaches 10^{-7} . The loss is an L1 loss (equation (5)), shown in Fig. 5 over four orders of magnitude of dataset size.

The neural-scaling results for the equivariant GNN, PaiNN (Fig. 5), show monotonic improvements to the loss with increasing dataset size. For a fixed dataset size, the converged loss is strongly correlated with the total training time (compute) and model capacity. Other than for 10^3 datapoints (for which some small models reach convergence quickly), the converged loss has a Spearman correlation coefficient $\rho \geq 0.88$ with the model capacity and $\rho \geq 0.75$ with the total training time. This means that the best models are those with optimal capacity that are able to train the longest without the validation loss plateauing. The optimal capacity and depth versus width change with the dataset size, that is, the ideal GNN capacity is dataset-size dependent, and these choices can impact the converged loss. These effects may also be artefacts of random initialization that would diminish with repeated trials. Interestingly, there is a stark change at 10^4 datapoints—the converged loss is then nearly perfectly rank correlated with model capacity (Spearman's $\rho \geq 0.93$). This might indicate that substantial overlap exists between the training and validation sets, such that higher capacity models are merely exhibiting better memorization than lower-capacity models. In these experiments, the validation set is constructed from unseen geometries and seen species (chemical species are the same in the training and validation sets). Repeating these experiments with a hold-out set of unseen chemical species will reveal whether the same trend holds, which would indicate that rather than memorizing, the network is achieving generalization to new chemistries.

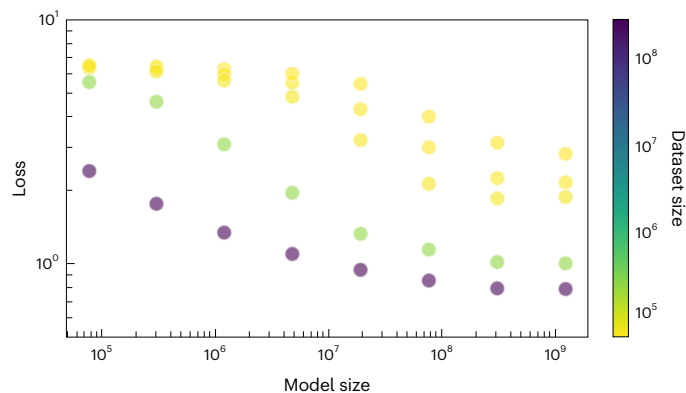


Fig. 4 | Neural scaling of ChemGPT model performance (validation loss) as a function of model (number of non-embedding parameters) and dataset (number of tokens) size. ChemGPT is pre-trained on up to 10 million molecules (300 million tokens) from PubChem. Performance improvements are seen for models up to 1 billion non-embedding parameters and continuous improvements are observed with increasing pre-training dataset size.

We observe similar trends in neural scaling for the invariant GNN, SchNet (Supplementary Fig. 3), although the equivariant GNNs, PaiNN and Allegro (Supplementary Fig. 4), show better scaling efficiency. A comparison of neural scaling between SchNet, PaiNN and Allegro for models with fixed capacity (equation (6)), $c = 64$ (4 layers, width 16), is shown in Supplementary Fig. 5. Over many orders of magnitude of dataset size, PaiNN and Allegro show greater sample efficiency, quantified by the calculated scaling exponents (Supplementary Table 1). That is, not only do the equivariant GNNs achieve better performance for a given data budget but also they exhibit larger β scaling parameter values, meaning that the loss improves more quickly with increasing amounts of training data. This is due to the models' equivariance, which is known to produce greater sample efficiency^{9,10,38}, but it is interesting to note that this trend persists to much larger and more chemically diverse datasets than were previously considered, which typically include only 10^2 – 10^3 molecular geometries from a single molecular species. We observe the same trends for calculated scaling exponents when the smallest (10^2) and largest (10^5) datasets are excluded from the power-law fits (Supplementary Table 1). Our results and recent work³⁹ on hierarchical learning in equivariant GNNs suggest that the tensor order of features has an important role in the sample efficiency of these models. Future theoretical and empirical work is needed to untangle the competition between equivariance that is enforced via architectures and features and 'learned' equivariance⁴⁰ achieved through data augmentation and training data.

Neural scaling enables substantial improvements to loss

Next, we briefly highlight the practical outcomes and usages of TPE and neural scaling as enabling technologies for scalable scientific deep learning. On the basis of the results presented above, TPE can be used in conjunction with any HPO routine to enable aggressive early stopping and accelerate HPO without sacrificing improvements to the loss. Clearly, the benefits of this approach become more pronounced in chemical and biological applications, where new network architectures must be continuously retrained, optimized and evaluated on heterogeneous datasets.

Similarly, neural scaling provides practical ways to improve model pre-training loss and efficiency. Given an unlimited data and computation budget, the minimum loss in the neural-scaling plot and corresponding model can be used. For example, the 300 million parameter ChemGPT model trained on 300 million tokens minimizes the loss in Fig. 4. Likewise, the PaiNN model with capacity $\sim 1,000$ trained on 10^5 frames minimizes the loss in Fig. 5. This may be valuable for

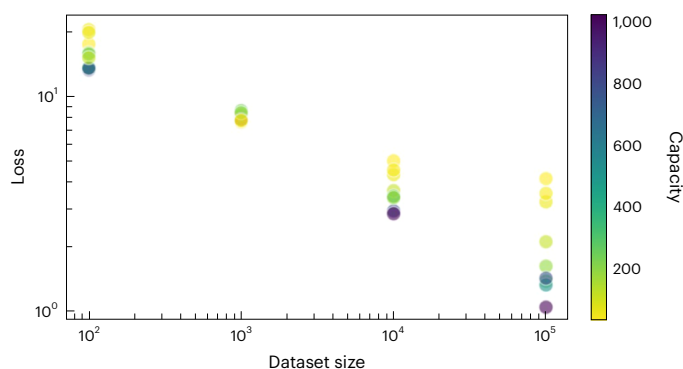


Fig. 5 | Neural scaling of PaiNN model performance (validation loss) as a function of model capacity (depth $\times$ width) and dataset size (number of geometries). PaiNN is trained to predict atomic forces from density functional theory calculations on small organic molecules from the ANI-1x dataset. Improvements to the loss are seen for models with greater capacity and continuous improvements are observed with increasing dataset size.

pre-trained models that are designed to be reused and fine-tuned, where the training cost is amortized over many downstream applications. However, for many scientific applications, greedily optimizing for the minimum loss is not practical or even necessary. From the neural-scaling results, identifying regions with the steepest slope allows for optimal and efficient allocation of resources. For example, for large chemical language models, the greatest loss improvements (Fig. 4) are seen for large data budgets when scaling up small models (10^5 parameters). For small data budgets, more rapid loss improvements are seen when scaling up medium-sized models (10^7 parameters). For NFFs, there are diminishing returns with increasing dataset sizes for low-capacity models, while high-capacity models show rapid improvements with increasing dataset size (Fig. 5). The benefits from scaling model and dataset sizes should therefore be balanced against the increased computational costs to find the most computationally and data-efficient opportunities for improvement. Beyond optimizing resource allocation, the grand challenge for large pre-trained models is to achieve new capabilities and superior performance on downstream tasks.

Discussion

In this paper, we developed and applied strategies for scaling large chemical language models and GNN interatomic potentials. To enable the efficient scaling of deep chemical models under computational resource constraints, we introduced TPE, a generalization of TSE that reduces the computational costs of HPO and model selection for chemical language models and GNN interatomic potentials. The use of TPE enabled large-scale experiments, training GPT-style chemical models with over 1 billion non-embedding parameters on nearly 10 million molecules. It also made training tractable for invariant and equivariant GNNs with a wide range of model capacities on up to 100,000 three-dimensional molecular geometries (~4.5 million force labels). We discovered empirical power-law ‘neural scaling’ behaviour that quantifies how model loss depends on the scale of model and dataset size over many orders of magnitude. These results enable optimal allocation of computational and data budgets for maximally efficient model loss improvements, and make scalable scientific deep learning more accessible to a broader community of researchers. A key finding in our work is that for both large chemical language models and NFFs, we have not saturated model loss with respect to model size, dataset size or compute. Much further work remains to be done in investigating the limitations of scaling for chemistry. Finally, the effects of physics-based priors on scaling behaviour give a rich description of how the incorporation of physics, known empirical relationships and other forms of knowledge into machine learning frameworks impact

both learning quality and efficiency. Future work in this area is well poised to yield fundamental advances in scientific machine learning.

Methods

In this section, we report details of the models considered in the paper and settings for the experiments performed in this paper. We define neural scaling and the model architectures considered here, which are chosen specifically for their likelihood to exhibit interesting scaling behaviour. Then we introduce strategies to enable scaling large chemical models and investigations of scaling behaviour.

Neural scaling

For large language and computer vision models trained to convergence with sufficient model parameters and/or data, performance is characterized by empirical scaling laws where the loss scales as a power law¹⁸ of the form

$$L(R) = \alpha R^{-\beta} \quad (1)$$

for coefficient α , scaling exponent β and resource R . R is the number of model parameters, dataset size or compute. β measures the slope of the power law and indicates the scaling efficiency of the model with respect to a scaling factor, R . The power-law trends break down in ‘resolution limited’ regimes³⁴, indicating that the model (dataset) size is insufficient for the given amount of data (model parameters).

Neural scaling presents a best-case scenario for model pre-training loss improvements with increasing resources, and allows for optimal allocation of fixed budgets, for example, to decide whether longer training, more data or larger models will be most efficient for improving pre-training loss. Comparing neural-scaling exponents also provides a fundamental metric for measuring resource efficiency across model architectures. Investigations into neural scaling in the NLP domain have revealed general conclusions about overfitting, sensitivity to architectural choices, transfer learning and sample efficiency¹⁸. These factors are equally or more important in scientific deep learning applications, where rapid advances are being made in specialized architecture development, and it is often unclear how architectures will perform beyond the small benchmark datasets that are commonly available in scientific settings.

Large chemical language models

Strings are a simple representation for molecular graphs⁴¹, thereby making sequence-based machine learning models a natural choice for working with chemical data. Following the demonstrated pre-training loss improvements of transformer-based models with increasing model and dataset sizes^{8,18,34}, we designed a large generative language model for chemistry called ChemGPT to investigate the impact of dataset and model size on pre-training loss. ChemGPT is a generative pre-trained transformer 3 (GPT3)-style model^{42,43} based on GPT-Neo^{44,45} with a tokenizer for self-referencing embedded strings (SELFIES)^{41,46} representations of molecules. SELFIES enforce chemical validity and are straightforward to tokenize, but ChemGPT can easily be used with simplified molecular-input line-entry system (SMILES) strings as well²⁸. For chemical language modelling, a set of molecules ($x_1, x_2, \dots, x_n$) is represented with each molecule as a sequence of symbols ($s_1, s_2, \dots, s_n$). The probability of a sequence, $p(x)$ is factorized as the product of conditional probabilities⁴⁷:

$$p(x) = \prod_{i=1}^n p(s_i | s_1, \dots, s_{i-1}). \quad (2)$$

ChemGPT uses the transformer⁴⁸ architecture with a self-attention mechanism to compute conditional probabilities, estimate $p(x)$, and sample from it to generate new molecules. ChemGPT is pre-trained on molecules from PubChem⁴⁹ with a causal language modelling task,

where the model must predict the next token in a sequence, given the previous tokens. ChemGPT models of up to 1 billion non-embedding parameters are trained on up to 10 million molecules, whereas typical chemical generative models have less than 1 million parameters and are trained on less than one million samples^{28,33}.

GNN force fields

For many tasks in chemistry, molecular geometry and three-dimensional structure are essential and string-based representations of the chemical graph are not sufficient. NFFs are GNNs that take molecular geometries as inputs, described by a set of atomic numbers ($Z_1, \dots, Z_n | Z_i \in \mathbb{N}$) and Cartesian coordinates ($\mathbf{r}_1, \dots, \mathbf{r}_n | \mathbf{r}_i \in \mathbb{R}^3$). The NFF with parameters θ, f_θ , predicts a real-valued energy $\hat{E} = f_\theta(X)$ for an atomistic configuration X . The NFF produces energy-conserving atomic forces by differentiating the energies with respect to the atomic coordinates

$$\hat{F}_{ij} = -\frac{\partial \hat{E}}{\partial r_{ij}} \quad (3)$$

for atom i and Cartesian coordinate j . Typically, the network is trained by minimizing the loss $\mathcal{L}$ computed from the average mean squared error for a mini-batch of size N

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N [\alpha_E \|E_i - \hat{E}_i\|^2 + \alpha_F \|F_i - \hat{F}_i\|^2] \quad (4)$$

where α_E and α_F are coefficients that determine the relative weighting of energy and force predictions during training⁵⁰. For scaling experiments we use the L1 loss or mean absolute error

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N [\alpha_E \|E_i - \hat{E}_i\| + \alpha_F \|F_i - \hat{F}_i\|] \quad (5)$$

which we empirically find to show more robust convergence behaviour.

In this work, we consider four flavours of NFFs: SchNet⁵¹, PaiNN⁵², Allegro¹⁰ and SpookyNet³⁰. This series of models represents increasingly physics-informed model architectures, from models with internal layers that manipulate only E(3) invariant quantities (SchNet) to those that use E(3) equivariant quantities (PaiNN, Allegro, SpookyNet), strictly local models with learned many-body functions and no message passing (Allegro), and physically informed via empirical corrections (SpookyNet). The power and expressivity of these GNNs can be defined in terms of their capacity³⁶

$$c = d \times w \quad (6)$$

where d is depth (number of layers or convolutions⁵¹) and w is width (the embedding dimension or number of basis functions employed by each convolution). Capacity is a simple parameter to vary during neural-scaling experiments, because model size is not a strictly useful scaling parameter for GNNs³⁶. Typical evaluations of NFFs consider training dataset sizes of less than 1,000 three-dimensional geometries of a single chemical species, which leads to insensitivity to model capacity because of the simplicity of the learning task¹⁷. Here, we consider up to 100,000 training geometries (corresponding to 4.5 million force labels) and GNNs with millions of trainable parameters.

Accelerating HPO with TPE

Because model hyperparameters, including learning rates and batch sizes, are essential for achieving optimal losses and are non-transferable between different domains and model/dataset sizes²¹, we need efficient strategies for scalable HPO in deep chemical models. We adapt TSE²², a simple technique for ranking computer vision architectures during neural architecture searches, to accelerate HPO and model selection

for ChemGPT and GNNs. We call this method TPE, as it uses training speed to more generally enable performance estimation across a wide range of applications. TPE generalizes TSE to HPO for new deep learning domains (LLMs, GNNs) and can be used to directly predict converged loss, in addition to rank ordering different architectures. While not the main contribution of this work, TPE is an effective strategy for accelerating scaling studies under resource constraints. TPE is used for rapid experimentation and to discover which hyperparameters are most important in new domain applications, as well as what hyperparameter regimes to investigate. Similar methods including Hyberband⁵³ accelerate HPO by automating early stopping during training. The technical details of TPE are provided in the ‘Training performance estimation’ section in Methods.

Experimental settings

All experiments described in this paper were conducted on NVIDIA Volta V100 graphics processing units (GPUs) with 32 GB of memory per node and 2 GPUs per node. All models were implemented in PyTorch⁵⁴ and trained with the distributed data parallel accelerator⁵⁵, the NVIDIA Collective Communication Library, PyTorch Lightning⁵⁶ and LitMatter⁵⁷ for multi-GPU, multi-node training.

Large language models

The ChemGPT model architecture is based on the GPT-Neo^{44,45} transformer implementation in HuggingFace⁵⁸. The model has 24 layers, with variable width, w , where $w \in (16, 32, 64, 128, 256, 512, 1,024, 2,048)$ and w determines the model size. Model sizes range from 77,600 to 1,208,455,168 non-embedding parameters. The model is trained via stochastic gradient descent with the AdamW⁵⁹ optimizer, using a learning rate of 2×10^{-5} , a per-GPU batch size of 8 and a constant learning rate schedule with 100 warm-up steps for scaling experiments. Models were trained for 10 epochs in a self-supervised manner, with a cross-entropy loss for causal language modelling. The number of epochs for training was chosen due to computational limitations, but importantly it is large enough to clearly distinguish differences in model performance from the empirical scaling results. As the initial publication of this work, new ‘compute optimal’ scaling laws⁶⁰ have been discovered for general LLMs. Our results and this recent work clearly suggest that with increased compute and engineering time, larger chemical models could be trained.

The training dataset for scaling experiments is PubChem10M⁸, a set of 10 million SMILES strings. Five percent of the data is randomly sampled and held out as a fixed validation set of size 500,000 molecules. Variable training datasets with sizes 10^n , where $n \in (2, 3, 4, 5, 6)$, were used. The largest training dataset includes all molecules in PubChem10M, excluding the validation set. The maximum vocabulary size was 10,000 and the maximum sequence length was 512 tokens. SMILES strings were converted to SELFIES using version 1.0.4 of the SELFIES library⁴⁶. SELFIES were tokenized by splitting individual strings into minimally semantically meaningful tokens denoted by brackets, including start-of-string, end-of-string and padding tokens. Dataset sizes range from 51,200 to 304,656,384 tokens.

Graph neural networks

We train GNNs to predict the forces of molecular geometries. Force-only training ($\alpha_E = 0$ in equation (5)) was used for neural-scaling experiments to improve convergence and avoid issues with systematic drift in predicted energies, which we identified during the course of this work and plan to address in future work. We use the SchNet⁶¹, PaiNN⁵², Allegro¹⁰ and SpookyNet³⁰ models. Model implementations are from the NeuralForceField repository^{50,62,63} and the Allegro repository¹⁰. Model sizes (w in equation (6)) were varied between 16, 64 and 256, while the number of layers/convolutions (d in equation (6)) was chosen to be 2, 3 or 4. A 5 Å nearest-neighbour cut-off was used. All other model hyperparameters were set to default values from the original implementations.

GNN models were trained with stochastic gradient descent using the Adam⁶⁴ optimizer. For Allegro, $l=1$ internal features were used.

A learning rate scheduler reduced the learning rate by 0.5× after 30 epochs without improvement in the validation loss, with a minimum learning rate of 10^{-7} . Early stopping was applied after 50 epochs without improvement in the validation loss, and training was capped at 1,000 epochs. Initial learning rates of 10^{-3} , 10^{-4} and 10^{-5} , and per-GPU batch sizes of 4, 8, 16, 32 and 64 were used during HPO experiments, while keeping the network architecture hyperparameters fixed. Models were trained for 50 epochs during HPO to approximate a full training budget, with a limited percentage of the total training budget used to calculate TSE.

The training dataset was assembled from ANI-1x^{37,65}, which contains energies and forces from 5 million density functional theory calculations for small molecules. A fixed validation dataset of 50,000 frames was held out by random sampling. Different splits of training were taken with sizes 10^n where $n \in (2, 3, 4, 5, 6)$. Training datasets for TPE were assembled by randomly sampling 1,000 structures from molecular dynamics trajectories for each of the 10 molecules available in the revised MD-17³² dataset, for a total of 10,000 training samples. A validation dataset of equal size was constructed from the remaining geometries. Revised MD-17 is an updated version of the MD-17³¹ dataset, recomputed at the PBE/def2-SVP level of theory with strict convergence criteria to remove noise found in the original MD-17 dataset.

Training performance estimation

HPO typically involves training tens or hundreds of networks and using random search and/or Bayesian optimization to identify optimal hyperparameters. For optimal performance, the process must be repeated when considering new datasets or distribution shift.

By calculating the ‘training speed’ from only the first few epochs of training, the converged model performance is predicted and optimal hyperparameters are identified using only a small fraction of the total training budget. For example, networks that require 100 epochs to train to convergence are trained for only 10–20 epochs, and the final performance is predicted using TPE to identify the best performing networks, thereby saving 80–90% of the total training budget.

Training speed is estimated by summing the training losses of each mini-batch during the first T epochs of training. After training the network for T epochs with B training steps per epoch, TSE is defined as

$$\text{TSE} = \sum_{t=1}^T \left(\frac{1}{B} \sum_{i=1}^B \mathcal{L}(f_{\theta(t,i)}(\mathbf{X}_i), \mathbf{y}_i) \right), \quad (7)$$

for a loss function $\mathcal{L}$ and a neural network $f_{\theta(t,i)}$, with parameters θ at epoch t and mini-batch i . $(\mathbf{X}_i, \mathbf{y}_i)$ is a tuple of inputs and labels in the i th mini-batch. TSE is correlated with the converged performance of the network and can be used to rank networks early in training to yield substantial compute savings. Given a sufficient number of networks (5–10) that are trained to convergence, a linear regression of the form

$$L = m \times \text{TSE} + b \quad (8)$$

is fit with parameters m and b and the calculated TSE values to predict the converged loss, L . This allows predictions of converged network loss for partially trained networks evaluated during HPO based on its TSE values. Optimal hyperparameters are chosen to minimize TSE. In our experiments, we noted that L is monotonic in TSE, meaning that equation (8) is not needed to simply choose the best hyperparameters. The TSE values computed after a small number of epochs are sufficient for ranking model configurations and finding the optimal ones. Although leveraging equation (8) requires training some small number of networks to convergence to fit the parameters, it provides the benefit of being able to predict the expected performance of new

hyperparameter choices. In particular, this may provide guidance if a particular target loss value is desired, as equation (8) can be used to predict the performance gains potentially accessible through HPO. We find that TPE is robust over multiple orders of magnitude of learning rate for the networks and training regimes considered here.

Data availability

PubChem data for pre-training large language models are available through DeepChem⁶⁶. The Molecular Sets (MOSES) data are available through GitHub²⁷. The Enamine HTS Collection is available [here](#). The ANI-1x data for training neural force fields is available through Figshare³⁷. The revised MD-17 dataset was accessed [here](#). The FreeSolv⁶⁷ and Tox21⁶⁸ datasets are available through the Therapeutics Data Commons¹⁵ and MoleculeNet¹⁴.

Code availability

The code used to perform the experiments and TPE reported in this paper is available via GitHub in the LitMatter repository⁵⁷. ChemGPT is also available through the MolFeat library⁶⁹ and the ROGI-XD library²⁵. Neural force field model code is available [here](#) and Allegro model code is available [here](#). The GPT-Neo model that ChemGPT is based on is available [here](#). PubChem10M tokenizers using SELFIES versions 1.0.4 and 2.0.0 are available through the LitMatter repository and the HuggingFace Hub. Because of the substantial computational resources required to train large models and the value of those models, pre-trained model checkpoints for ChemGPT are available via the HuggingFace Hub. Pre-trained model checkpoints for PaiNN and Allegro are available through Figshare⁷⁰.

References

- Sejnowski, T. J. The unreasonable effectiveness of deep learning in artificial intelligence. *Proc. Natl Acad. Sci. USA* **117**, 30033–30038 (2020).
- Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. Preprint at <https://arxiv.org/abs/1810.04805> (2018).
- Ramesh, A. et al. Zero-shot text-to-image generation. In *Proc. 38th International Conference on Machine Learning* Vol. 139, 8821–8831 (PMLR, 2021).
- Bommasani, R. et al. On the opportunities and risks of foundation models. Preprint at <https://arxiv.org/abs/2108.07258> (2021).
- Jumper, J. et al. Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
- Chanussot, L. et al. Open Catalyst 2020 (OC20) dataset and community challenges. *ACS Catal.* **11**, 6059–6072 (2021).
- Sriram, A., Das, A., Wood, B. M., Goyal, S. & Zitnick, C. L. Towards training billion parameter graph neural networks for atomic simulations. Preprint at <https://arxiv.org/abs/2203.09697> (2022).
- Chithrananda, S., Grand, G. & Ramsundar, B. Chemberta: large-scale self-supervised pretraining for molecular property prediction. Preprint at <https://arxiv.org/abs/2010.09885> (2020).
- Batzner, S. et al. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat. Commun.* **13**, 2453 (2022).
- Musaelian, A. et al. Learning local equivariant representations for large-scale atomistic dynamics. *Nat. Commun.* **14**, 579 (2023).
- Trewartha, A. et al. Quantifying the advantage of domain-specific pre-training on named entity recognition tasks in materials science. *Patterns* **3**, 100488 (2022).
- Kalinin, S. V., Ziatdinov, M., Sumpter, B. G. & White, A. D. Physics is the new data. Preprint at <https://arxiv.org/abs/2204.05095> (2022).
- Coley, C. W. Defining and exploring chemical spaces. *Trends Chem.* **3**, 133–145 (2021).
- Wu, Z. et al. Moleculenet: a benchmark for molecular machine learning. *Chem. Sci.* **9**, 513–530 (2018).

15. Huang, K. et al. Therapeutics data commons: machine learning datasets and tasks for drug discovery and development. In *Proc. Neural Information Processing Systems Track on Datasets and Benchmarks 1* (eds Vanschoren, J. & Yeung, S.) (Curran Associates, 2021).
16. Pappu, A. & Paige, B. Making graph neural networks worth it for low-data molecular machine learning. Preprint at <https://arxiv.org/abs/2011.12203> (2020).
17. Gasteiger, J. et al. How do graph networks generalize to large and diverse molecular systems? Preprint at <https://arxiv.org/abs/2204.02782> (2022).
18. Kaplan, J. et al. Scaling laws for neural language models. Preprint at <https://arxiv.org/abs/2001.08361> (2020).
19. Henighan, T. et al. Scaling laws for autoregressive generative modeling. Preprint at <https://arxiv.org/abs/2010.14701> (2020).
20. Sevilla, J. et al. Compute trends across three eras of machine learning. In *2022 International Joint Conference on Neural Networks (IJCNN)* 1–8 (IEEE, 2022).
21. Yang, G. et al. Tensor programs V: tuning large neural networks via zero-shot hyperparameter transfer. Preprint at <https://arxiv.org/abs/2203.03466> (2022).
22. Ru, B. et al. Speedy performance estimation for neural architecture search. In *Advances in Neural Information Processing Systems* Vol. 34 (eds Ranzato, M. et al.) 4079–4092 (Curran Associates, 2021).
23. Ross, J. et al. Large-scale chemical language representations capture molecular structure and properties. *Nat. Mach. Intell.* **4**, 1256–1264 (2022).
24. Ahmad, W., Simon, E., Chithrananda, S., Grand, G. & Ramsundar, B. ChemBERTa-2: towards chemical foundation models. Preprint at <https://arxiv.org/abs/2209.01712> (2022).
25. Graff, D. E., Pyzer-Knapp, E. O., Jordan, K. E., Shakhnovich, E. I. & Coley, C. W. Evaluating the roughness of structure–property relationships using pretrained molecular representations. Preprint at <https://arxiv.org/abs/2305.08238> (2023).
26. Musaelian, A., Johansson, A., Batzner, S. & Kozinsky, B. Scaling the leading accuracy of deep equivariant models to biomolecular simulations of realistic size. Preprint at <https://arxiv.org/abs/2304.10061> (2023).
27. Polykovskiy, D. et al. Molecular Sets (MOSES): a benchmarking platform for molecular generation models. *Front. Pharmacol.* **11**, 1931 (2020).
28. Skinnider, M. A., Stacey, R. G., Wishart, D. S. & Foster, L. J. Chemical language models enable navigation in sparsely populated chemical space. *Nat. Mach. Intell.* **3**, 759–770 (2021).
29. McCandlish, S., Kaplan, J., Armodei, D. & OpenAI Dota Team An empirical model of large-batch training. Preprint at <https://arxiv.org/abs/1812.06162> (2018).
30. Unke, O. T. et al. SpookyNet: learning force fields with electronic degrees of freedom and nonlocal effects. *Nat. Commun.* **12**, 7273 (2021).
31. Chmiela, S. et al. Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv.* **3**, 1603015 (2017).
32. Christensen, A. S. & von Lilienfeld, O. A. On the role of gradients for machine learning of molecular energies and forces. *Mach. Learn. Sci. Technol.* **1**, 045018 (2020).
33. Flam-Shepherd, D., Zhu, K. & Aspuru-Guzik, A. Keeping it simple: language models can learn complex molecular distributions. *Nat. Commun.* **13**, 3293 (2022).
34. Bahri, Y., Dyer, E., Kaplan, J., Lee, J. & Sharma, U. Explaining neural scaling laws. Preprint at <https://arxiv.org/abs/2102.06701> (2021).
35. Caballero, E., Gupta, K., Rish, I. & Krueger, D. Broken neural scaling laws. Preprint at <https://arxiv.org/abs/2210.14891> (2022).
36. Loukas, A. What graph neural networks cannot learn: depth vs width. Preprint at <https://arxiv.org/abs/1907.03199> (2019).
37. Smith, J. S. et al. The ANI-1ccx and ANI-1x data sets, coupled-cluster and density functional theory properties for molecules. *Sci. Data* **7**, 1–10 (2020).
38. Batatia, I., Kovács, D. P., Simm, G. N. C., Ortner, C. & Csányi, G. MACE: higher order equivariant message passing neural networks for fast and accurate force fields. In *Advances in Neural Information Processing Systems* Vol. 35 (eds Koyejo, S. et al.) 11423–11436 (Curran Associates, 2022).
39. Rackers, J. A. & Rao, P. Hierarchical learning in Euclidean neural networks. Preprint at <https://arxiv.org/abs/2210.0476> (2022).
40. Gruver, N., Finzi, M., Goldblum, M. & Wilson, A. G. The lie derivative for measuring learned equivariance. Preprint at <https://arxiv.org/abs/2210.02984> (2022).
41. Krenn, M. et al. SELFIES and the future of molecular string representations. *Patterns* **3**, 100588 (2022).
42. Radford, A. et al. Language models are unsupervised multitask learners. *OpenAI Blog* **1**, 9 (2019).
43. Brown, T. et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **33**, 1877–1901 (2020).
44. Black, S., Leo, G., Wang, P., Leahy, C. & Biderman, S. GPT-Neo: large scale autoregressive language modeling with Mesh-Tensorflow. *Zenodo* <https://doi.org/10.5281/zenodo.5297715> (2021).
45. Gao, L. et al. The Pile: an 800gb dataset of diverse text for language modeling. Preprint at <https://arxiv.org/abs/2101.00027> (2020).
46. Krenn, M., Häse, F., Nigam, A., Friederich, P. & Aspuru-Guzik, A. Self-referencing embedded strings (selfies): a 100% robust molecular string representation. *Mach. Learn. Sci. Technol.* **1**, 045024 (2020).
47. Bengio, Y., Ducharme, R. & Vincent, P. A neural probabilistic language model. In *Advances in Neural Information Processing Systems* Vol 13 (eds Leen, T. et al.) (MIT Press, 2000).
48. Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30**, (2017).
49. Kim, S. et al. PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Res.* **49**, 1388–1395 (2021).
50. Schwalbe-Koda, D., Tan, A. R. & Gómez-Bombarelli, R. Differentiable sampling of molecular geometries with uncertainty-based adversarial attacks. *Nat. Commun.* **12**, 5104 (2021).
51. Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A. & Müller, K.-R. SchNet—a deep learning architecture for molecules and materials. *J. Chem. Phys.* **148**, 241722 (2018).
52. Schütt, K., Unke, O. & Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning* 9377–9388 (PMLR, 2021).
53. Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A. & Talwalkar, A. Hyperband: a novel bandit-based approach to hyperparameter optimization. *J. Mach. Learn. Res.* **18**, 6765–6816 (2017).
54. Paszke, A. et al. Pytorch: an imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32**, 8024–8035 (2019).
55. Li, S. et al. PyTorch distributed: experiences on accelerating data parallel training. Preprint at <https://arxiv.org/abs/2006.15704> (2020).
56. Falcon, W. et al. Pytorch lightning. *GitHub* <https://github.com/PyTorchLightning/pytorch-lightning> (2019).
57. Frey, N. C. et al. Scalable geometric deep learning on molecular graphs. In *NeurIPS 2021 AI for Science Workshop* (2021).
58. Wolf, T. et al. HuggingFace’s transformers: state-of-the-art natural language processing. Preprint at <https://arxiv.org/abs/1910.03771> (2019).
59. Loshchilov, I., Hutter, F. Decoupled weight decay regularization. Preprint at <https://arxiv.org/abs/1711.05101> (2017).

60. Hoffmann, J. et al. Training compute-optimal large language models. Preprint at <https://arxiv.org/abs/2203.15556> (2022).
61. Schütt, K. T. et al. SchNet: a continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems* Vol. 30 (eds Guyon, I. et al.) 992–1002 (Curran Associates, 2017).
62. Axelrod, S. & Gomez-Bombarelli, R. Molecular machine learning with conformer ensembles. *Mach. Learn. Sci. Technol.* **4**, 035025 (2023).
63. Axelrod, S., Shakhnovich, E. & Gómez-Bombarelli, R. Excited state, non-adiabatic dynamics of large photoswitchable molecules using a chemically transferable machine learning potential. *Nat. Commun.* **13**, 3440 (2022).
64. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at <https://arxiv.org/abs/1412.6980> (2014).
65. Smith, J. S., Nebgen, B., Lubbers, N., Isayev, O. & Roitberg, A. E. Less is more: sampling chemical space with active learning. *J. Chem. Phys.* **148**, 241733 (2018).
66. Ramsundar, B. et al. *Deep Learning for the Life Sciences* (O'Reilly Media, 2019).
67. Mobley, D. L. & Guthrie, J. P. FreeSolv: a database of experimental and calculated hydration free energies, with input files. *J. Comput. Aided Mol. Des.* **28**, 711–720 (2014).
68. Huang, B. & Von Lilienfeld, O. A. Understanding molecular representations in machine learning: the role of uniqueness and target similarity. *J. Chem. Phys.* **145**, 161102 (2016).
69. Noutahi, E. et al. rbyrne-momatx: datamol-io/molfeat: 0.8.9. Zenodo <https://doi.org/10.5281/zenodo.7955465> (2023).
70. Honda, S., Shi, S. & Ueda, H. R. SMILES transformer: pre-trained molecular fingerprint for low data drug discovery. (2019).

Acknowledgements

We acknowledge the MIT SuperCloud and the Lincoln Laboratory Supercomputing Center for providing HPC and consultation resources that contributed to the research results reported within this paper. We acknowledge the MIT SuperCloud team: W. Arcand, D. Bestor, W. Bergeron, C. Byun, M. Hubbell, M. Houle, M. Jones, J. Kepner, A. Klein, P. Michaleas, L. Milechin, J. Mullen, A. Prout, A. Reuther, A. Rosa and C. Yee. We acknowledge J. Marchese for proofreading. This material is based on work supported by the Assistant Secretary of Defense for Research and Engineering under Air Force contract number FA8702-15-D-0001, and by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator under Cooperative Agreement number FA8750-19-2-1000. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Assistant Secretary of

Defense for Research and Engineering, or the United States Air Force. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Author contributions

N.C.F., R.G.-B., C.W.C. and V.G. conceptualized the project. N.C.F. developed the methodology, performed the investigations and made the visualizations. All authors contributed to analysing the results, and writing and editing the paper. V.G. supervised the project.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s42256-023-00740-3>.

Correspondence and requests for materials should be addressed to Nathan C. Frey.

Peer review information *Nature Machine Intelligence* thanks the anonymous reviewers for their contribution to the peer review of this work. Primary Handling Editor: Jacob Huth, in collaboration with the *Nature Machine Intelligence* team.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023