nature reviews physics

https://doi.org/10.1038/s42254-024-00709-0

Perspective

Check for updates

Rigor with machine learning from field theory to the Poincaré conjecture

Sergei Gukov [1,2], James Halverson [3,4] ✉ & Fabian Ruehle [3,4,5] ✉

Abstract

Despite their successes, machine learning techniques are often stochastic, error-prone and blackbox. How could they then be used in fields such as theoretical physics and pure mathematics for which error-free results and deep understanding are a must? In this Perspective, we discuss techniques for obtaining zero-error results with machine learning, with a focus on theoretical physics and pure mathematics. Non-rigorous methods can enable rigorous results via conjecture generation or verification by reinforcement learning. We survey applications of these techniques-for-rigor ranging from string theory to the smooth 4D Poincaré conjecture in low-dimensional topology. We also discuss connections between machine learning theory and mathematics or theoretical physics such as a new approach to field theory motivated by neural network theory, and a theory of Riemannian metric flows induced by neural network gradient descent, which encompasses Perelman's formulation of the Ricci flow that was used to solve the 3D Poincaré conjecture.

Sections

[1]Richard N. Merkin Center for Pure and Applied Mathematics, California Institute of Technology, Pasadena, CA, USA. [2]Dublin Institute for Advanced Studies, Dublin, Ireland. [3]Department of Physics, Northeastern University, Boston, MA, USA. [4]The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, Cambridge, USA. [5]Department of Mathematics, Northeastern University, Boston, MA, USA. ✉e-mail: j.halverson@northeastern.edu; f.ruehle@northeastern.edu

Introduction

We live in a remarkable time in the history of science: from the perspective of our distant descendants, we are near the beginning of the use of computer science for scientific discovery. In the future, some scientific discoveries may rely on the promise of quantum computing, but today there is an ongoing revolution in artificial intelligence (AI) and machine learning (ML) that is already transforming the natural sciences, for example, in protein folding[1] (see also general[2] and field-specific[3,4] reviews).

Despite numerous success stories, ML techniques are often error-prone, stochastic and blackbox. Classification problems always have some degree of error, and this error may be induced by the addition of adversarial noise that, for instance, can cause a neural network (NN) to mis-classify a turtle as a rifle and a cat as guacamole[5,6]. Such ML techniques are stochastic in a number of ways. In the supervised setting that tries to make predictions given model inputs, the predictions of the NN, and therefore the errors, generally depend on both a random initialization and a random training process. In the deep reinforcement learning (RL) setting that tries to learn a strategy for gameplay that maximizes rewards, the training process is stochastic owing to randomness in the exploration of the environment. Finally, ML techniques are often blackbox in the sense that the trained ML algorithm has millions, or even billions, of parameters that are difficult to interpret and understand.

These techniques may, therefore, appear ill-suited for application in fields such as theoretical physics and pure mathematics that prioritize rigorous results and deep understanding. Notions of rigor differ across scientific communities. Uncontrolled approximations are definitively not rigorous, and therefore controlled approximations are a significant upgrade, especially when accompanied by convergence guarantees that exist in certain limits, such as in Monte Carlo sampling in condensed matter physics or lattice field theory. However, unless one is truly in the limit in which convergence occurs, these are still techniques with non-zero, but controllable, error bars. Instead, in this Perspective, we are interested in presenting 'zero-error' results, the gold standard in many formal areas of theoretical physics and pure mathematics, which is henceforth what we mean by rigor.

In this Perspective, we survey techniques for obtaining rigor with ML, exemplified by recent results. We focus on two central ideas: making applied ML techniques rigorous and ensuring rigor by using theoretical ML from the outset.

We will consider conjecture generation and rigorous solution verification with RL as a means of making applied ML techniques rigorous. In conjecture generation, a human domain expert is brought into the ML loop to understand model predictions with interpretable AI techniques, with the hope of generating a conjecture that can be rigorously proven by a human. We discuss applications of this technique in obtaining new theorems in string theory, algebraic geometry and knot theory. Rigorous results may also be obtained by RL, through the definition of a game whose objective is to find a trajectory through a space of states that establishes a mathematical fact. We will discuss applications of this approach to low-dimensional topology. This direction includes a state-of-the-art program that can establish ribbonness of knots (far beyond typical human expert abilities) and which was used to rule out hundreds of proposed counterexamples to the smooth Poincaré conjecture in four dimensions (SPC4)[7].

We also discuss results in an emerging direction that seeks to use ML theory in theoretical physics and pure mathematics, including completely new approaches to both field theory and metric flows. These developments exploit recent results on the statistics and dynamics of NNs (refs. 8–10). On the statistics side, we present a correspondence between NNs and quantum field theory (QFT). This correspondence has the potential to provide a non-perturbative definition of QFT in the continuum. We discuss the relation to the Feynman path integral, the NN origin of interactions and symmetries in a field theory, and the realization of ϕ^4 theory as a NN field theory[11]. On the dynamics side, we consider a metric on a Riemannian manifold that is represented by a NN, trained with gradient descent. This framework has been used in approximations of numerical Calabi–Yau (CY) metrics, a crucial result in string theory. Neural tangent kernel (NTK) theory[9,10] is used to develop an associated theory of metric flows that generalizes known flows in mathematics, for instance Perelman's formulation[12] of Ricci flow[13] as a gradient flow, which may be explicitly realized with a NN metric flow.

There are many research areas and directions not covered in this Perspective, in part due to space limitations. For example, one active area of research is the application of ML tools to automated proof assistants[14–17]. These applications also provide rigorous results, by helping the automated prover to find the right sequence of logical steps to go from statement A to statement B, that is, to find a path among logical operations that connects assumptions of a theorem in question to its conclusion[18]. For the use of ML in various systems such as Lean, Isabelle and Mizar, see refs. 19–22.

Rigorous results from applied ML

One way in which ML can lead to rigorous results is its use in conjunction with domain experts. An example is conjecture generation, wherein a ML algorithm assists in formulating a conjecture that can subsequently be proven. Conjectures propose a relation between properties of an object that were previously not known to be related. NNs, being universal function approximators, can be used in a supervised setup to search for such relations: if the algorithm predicts one property from another, this hints at the existence of a relation. If the ML algorithm was a whitebox algorithm, such as decision trees or symbolic regression, the relation could be inferred directly. But in the case of a blackbox algorithm such as NNs, attribution techniques can be used, which help determine the features that are most relevant for NN predictions. In practice, moving from a trained ML algorithm to a mathematical theorem requires a back-and-forth between the ML and the human to refine the conjecture before it is proven.

Conjecture generation was introduced in ref. 23 in the context of string theory, which used a decision tree and logistic regression, whereas the use of NNs for this purpose was introduced in refs. 24–26. These ideas were also applied in knot and graph theory[27–29]. Moreover, the utility of performing simple feature reduction techniques such as principal component analysis has led to conjectures involving mirror symmetry for Fano varieties[30] or Bogomol'nyi-Prasad-Sommerfield spectra for superconformal field theories (S Gukov & R-K Seong, private communication). Many more examples can be found in refs. 3,4. See also ref. 31 for a recent theoretical approach to conjecture generation. Because conjecture generation requires interpretable AI, another avenue is to arrive at conjectures with symbolic regression, which fits data to simple mathematical expressions, for instance, in Newtonian mechanics and dark matter simulations[32].

Another way in which ML can lead to rigorous results is to use a trained ML agent to play a game wherein winning corresponds to solving a scientific problem such that the solution may be rigorously verified. This is the domain of RL, which optimizes a policy function that selects actions based on the current state of the system. As the agent explores an environment, it receives rewards that are used to update the policy function, leading to improved behaviour over time. RL was famously used in the games of Go and Chess, wherein through self-play and only knowing the rules of the game, the AI system was able to defeat the top human professional players. These techniques can be interpretable to domain experts by analysing the results of many games played by the trained AI agent. For instance in Chess, the games played by AlphaZero[33] demonstrate that it rediscovered the most popular openings used by humans[34] but also devised useful strategies that surprised grandmasters[35], such as a proclivity to push the *h*-pawn or favour positional play over pawn grabbing. We will explore examples of both techniques in string theory and knot theory.

String theory and algebraic geometry

Computation of physical quantities in string theory regularly requires algebraic geometry. Therefore, the first use of neural networks in string theory targeted questions in computational algebraic geometry[24–26], an area in which computations are notoriously challenging. In all those cases, the fact that NNs were able to predict the result with high accuracy suggested the existence of a hitherto unknown relation between the input features and the output. However, extracting an understanding of the reason the NN manages to perform so well, or whiteboxing it, is difficult. Remarkably, the computation of dimensions of line bundle cohomologies, for which ref. 26 showed very good performance with a NN, was later discovered[36–40] to be related to identifying patterns or regions in the input space, wherein a simple polynomial related to a topological index describes the individual cohomologies. Although this does not prove that the NN in ref. 26 made use of this fact, NNs have been shown to use pattern recognition and regression for predictions in other application areas. The relation uncovered in ref. 25, which conjecturally relates the minimum volume of a Sasaki–Einstein cone to simple toric data, remains unproven as of now.

Others papers[23,41] followed a different route: instead of uncovering an unknown relation by using a NN as a means to check existence of a map from features to labels, the authors used whitebox ML techniques such as logistic regression or decision trees from the onset. Similar to the other papers[24–26], they targeted algebraic geometry data. The conjecture uncovered in ref. 23 relates the existence of a ray in a toric variety that defines an elliptically fibred CY fourfold to a particular feature that is interesting for particle physics, E_6 gauge symmetry, whereas ref. 41 is concerned with Brill–Noether theory.

Rigorous results may also be obtained in string theory with RL, which we will discuss in the next section in the context of knot theory (see refs. 42–47).

Knot theory and the Poincaré conjecture

ML is increasingly applied in pure mathematics. Here, we focus on low-dimensional topology, especially knot theory. Knots are of interest not only as fundamental objects in mathematics but also because of their relation to observables in QFT and their relation to other aspects of topology, such as the Poincaré conjecture.

A mathematical knot is a circle embedded in the 3D space; an embedded collection of circles is a generalization known as a link. Its projection to a generic 2D plane yields a planar diagram, represented by strands of the knot that cross over or under each other. For a fixed knot or link K, the diagram is not unique because K can be continuously deformed and different projections may be chosen. Two diagrams represent the same link or knot if and only if they are related by a sequence of simple moves known as Reidemeister moves: the first three moves are illustrated in Fig. 1a. Quantities associated to a knot K that are invariant under Reidemeister moves (or ambient space isotopy) are topological invariants of the knot and may be integers, vectors or polynomials, for example. For studies using supervised ML to predict knot invariants, see refs. 28,29,48. We will focus on works that obtain rigorous results about knots via conjecture generation or RL.

Conjecture generation was used to prove a new theorem about knots[27]. A fully connected feed-forward network was trained to predict an invariant of a knot K, known as the signature $\sigma(K)$, from a set of 12 knot invariants. The model was trained to high accuracy, and an attribution technique called gradient saliency[49], which computes a score

$$r_{\mathrm{i}} = \frac{1}{|X|} \sum_{x \in X} \left| \frac{\partial L}{\partial x_{\mathrm{i}}} \right| \tag{1}$$

that averages the gradient of the loss function L with respect to a given input feature x_{i} over all of the examples x in a dataset X. This helps identify the invariants that are most important for predicting $\sigma(K)$. This interpretability analysis identified three invariants (the complex meridional and real longitudinal translations on the boundary torus of the knot complement) as much more significant than the other invariants for determining the knot signature, and they were used by domain experts to generate a conjecture about the signature of a knot. As in an earlier string theory work[23], the conjecture needed refinement by human experts. A final conjecture was proven, leading to a new theorem that for any hyperbolic knot, there exists a constant c such that

$$|2\sigma(K) - \mathrm{slope}(K)| \leq \mathrm{vol}(K)\,\mathrm{inj}\,(K)^{-3}, \tag{2}$$

in terms of the knot slope, hyperbolic volume and injectivity radius. Conceptually, the theorem was a surprise to some topologists because it relates geometric topological invariants, such as the slope and volume, to algebraic topological invariants, such as the signature, which a priori seem to be of different character.

One may also use RL to rigorously establish properties of knots. A knot K is said to be the unknot, which is topologically trivial, if there is a sequence of Reidemeister moves that simplifies any planar diagram representing the knot to a standard circle. The unknotting problem (or UNKNOT) seeks to determine whether or not K is the unknot, a problem that clearly has complexity growing with the number of crossings N in the planar diagram. The problem is known to be in NP $\cap$ co-NP (refs. 50–52), which means that both its positive and negative instances can be verified in polynomial time. In ref. 53, various RL agents were trained to find sequences of Reidemeister moves that simplify representatives of the unknot to the standard circle. The RL agents significantly outperformed a random walker that selected Reidemeister moves from a uniform distribution. In particular, a trust region policy optimization agent showed consistent performance for unknots described in terms of braids with increasing number of crossings, which correlates with how tangled up a knot representative is (Fig. 2). (Every knot can be written in terms of an element in a non-Abelian group called the braid group[54]. The length ℓ of the corresponding braid word is at least as big as the minimum number N_{min} of crossings in the knot projection, $\ell \geq N_{\mathrm{min}}$.)

A close cousin of the unknotting problem is the problem of distinguishing Kirby diagrams that represent 3-manifolds or 4-manifolds. A Kirby diagram basically consists of the data of a link with some integer

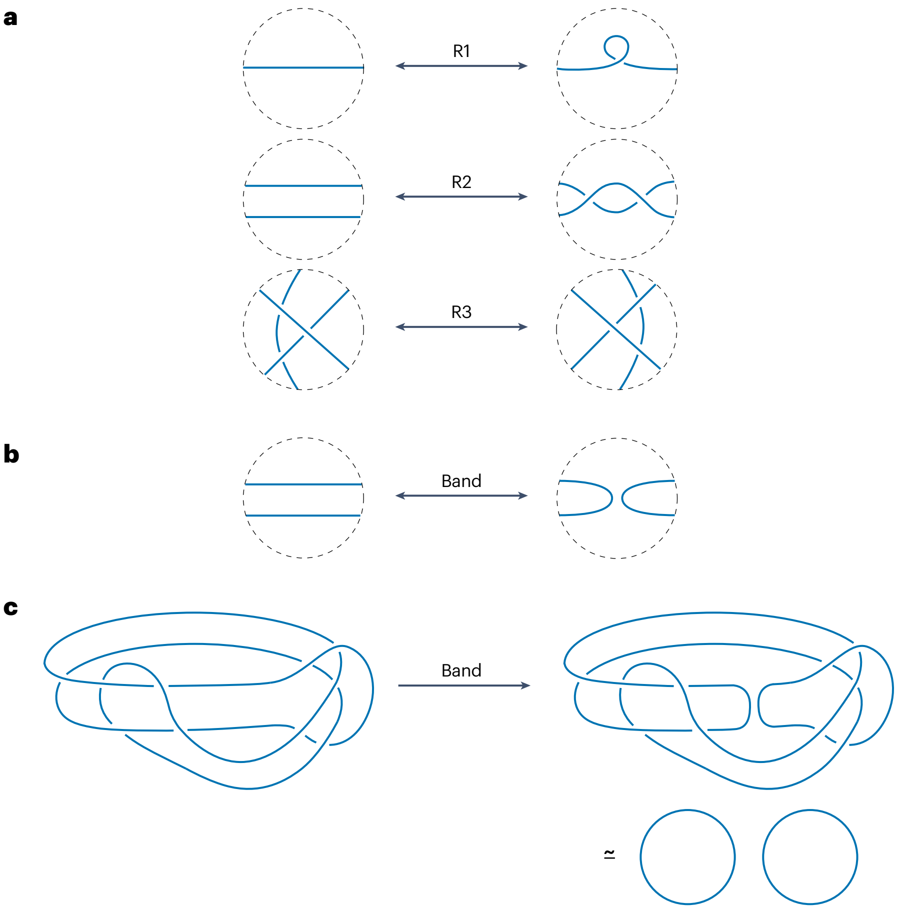

Fig. 1 | Moves and singularities of ribbon knots. **a**, The three Reidemeister moves (R1, R2 and R3). **b**, The band move, depicted, needs to preserve orientations on the link. **c**, After applying a band move to the square knot, the result can be deformed (via Reidemeister moves) into the unlink with two components[7].

labels, one for each link component. Much like planar knot projections that are related by Reidemeister moves, Kirby diagrams representing the same manifold are related by a small set of moves called Kirby moves. In a simplified setting wherein all link components are copies of the unknot, this data can be conveniently encoded in a combinatorial structure of a graph, called the plumbing graph. Graph neural networks provide a natural choice of architectures to study this problem, on which a trained asynchronous advantage actor–critic agent performed extremely well[55], outperforming, for example, a Deep-Q Network RL agent.

Another variant of the unknotting problem may be used to establish that a knot is ribbon, which is related to SPC4. A knot is ribbon if it bounds a ribbon disk, which is one that lives in a 3D space and has appropriately mild self-intersections. A knot is slice if it bounds a smoothly embedded disk in 4D space. Singularities of a ribbon disk are mild enough that there is a canonical procedure to add a dimension and turn it into a smoothly embedded disk in 4D; every ribbon knot is, therefore, slice.

SPC4 states that if a smooth 4-manifold is homoeomorphic to 4-sphere S^4, then it is diffeomorphic to S^4. SPC4 and sliceness are directly related to one another. Namely, if there is a pair of knots satisfying the following conditions:

- K_1 and K_2 have the same 0-surgery,
- K_1 is slice, and
- K_2 is not slice,

then an exotic 4-sphere may be constructed as homoeomorphic, but not diffeomorphic, to S^4, disproving the long-standing conjecture. (For the definition of 0-surgery, see conjecture 2.2 in ref. 7). Many pairs of knots satisfying the first condition are known. The computation of topological invariants, known as slice obstructions, may be used to establish the third condition, but there is no known algorithm for establishing sliceness or ribbonness. One may address ribbonness by modifying the allowed actions of the unknotting problem by adding band addition (see the last move in Fig. 1a). A knot is ribbon if there is a sequence of Reidemeister moves and band additions that simplify a planar diagram representing the knot to a collection of unlinked unknots (Fig. 1b).

In ref. 7, RL and Bayesian optimization of a Markov decision process were used to establish that a knot is ribbon, via the generalization of the unknotting game. The most effective overall agent was Bayesian optimized by taking speed into account, leading to a state-of-the-art ribbon verifier[56], which was able to establish ribbonness of some knots with many crossings (this was tested for up to 70 crossings). The agent was used to demonstrate that certain pairs of knots satisfying the first condition are both ribbon and, therefore, both slice. This ruled out over 800 potential counterexamples to SPC4.

Rigorous results from ML theory

Another option for obtaining rigorous results with ML is to use results from ML theory, instead of making numerical results rigorous.

This promising approach is still in its infancy, and we will restrict ourselves to summarizing a work in physics and mathematics that uses ML theory results on NN statistics or learning dynamics.

Central to our discussion will be the use of NN theory. A NN is simply a family of functions

$$\phi_\theta : \mathbb{R}^{n_{\mathrm{in}}} \to \mathbb{R}^{n_{\mathrm{out}}}, \tag{3}$$

with parameters θ, wherein one may choose a more general domain and co-domain, if desired. The input and output dimension of the NN are defined to be n_{in} and n_{out}, respectively. We will often suppress the subscript θ for notational simplicity. Specifying a NN requires choosing a concrete functional form, known as the architecture, that is usually a composition of simpler functions. When a NN is initialized on a computer, however, initial values of the parameters must be chosen by sampling θ from an initialization parameter distribution $P(\theta)$. The architecture and $P(\theta)$, therefore, specify an ensemble of functions at initialization, and an actively researched question in ML theory relates to the statistics of these ensembles (see, for instance, ref. 10).

One aspect of NN statistics that we will discuss below is known as the neural network–Gaussian process (NNGP) correspondence. It was first discovered[8] in the 1990s in the simplest case of a single-layer fully connected width-N NN, which in the case $n_{\mathrm{out}} = 1$, and there is no bias in the linear layers. The architecture and parameter densities given by

$$\phi(x) = \sum_{i=1}^{N} \sum_{j=1}^{n_{\mathrm{in}}} w_i' a(w_{ij} x_j), \quad P(w') = \mathcal{N}\left(0, \frac{\sigma_{w'}^2}{N}\right), \quad P(w) = \mathcal{N}\left(0, \frac{\sigma_w^2}{n_{\mathrm{in}}}\right), \tag{4}$$

where $a : \mathbb{R} \to \mathbb{R}$ is an element-wise non-linearity that is part of the architecture choice, the parameter set θ in this case corresponds to the matrix components w_i' and w_{ij}, σ is the standard deviation, and $\mathcal{N}$ denotes the normal (Gaussian) distribution. It has been shown that as $N \to \infty$, ϕ is drawn from a Gaussian process[8], which means that for any set of NN inputs $\{x_i\}$, the associated vector of outputs $\phi(\{x_i\})$ is drawn from a multivariate Gaussian distribution. Alternatively, in the case of continuous inputs, the functions $\phi(x)$ are drawn from a Gaussian distribution whose covariance is given by the two-point function $G^{(2)}(x, y) = \mathbb{E}[\phi(x)\phi(y)]$, where $\mathbb{E}$ denotes the expectation over the ensemble of functions, which may be computed[57] by integrating over the NN parameters

$$G^{(2)}(x, y) = \mathbb{E}[\phi(x)\phi(y)] = \int \mathrm{d}\theta\, P(\theta)\, \phi(x)\phi(y). \tag{5}$$

The corresponding function distribution $P(\phi) := e^{-S(\phi)}$ is specified in terms of the functional

$$S[\phi] = \frac{1}{2} \int \mathrm{d}^{n_{\mathrm{in}}}x\, \mathrm{d}^{n_{\mathrm{in}}}y\, \phi(x) G^{(2)}(x, y)^{-1} \phi(y), \tag{6}$$

known in physics as the (quadratic) action with Euclidean correlator $G^{(2)}(x, y)$ that is an analogue of the Feynman propagator. The quantity $G^{(2)}(x, y)^{-1}$ is defined via $\int \mathrm{d}^{n_{\mathrm{in}}}y G^{(2)}(x, y)^{-1} G^{(2)}(y, z) = \delta(x - z)$. The NNGP correspondence has been extended to many more architectures[58], wherein the generality of the phenomenon arises from the ubiquity of the central limit behaviour in NNs, generalizing the notion of 'width'. The higher moments are correlation functions $G^{(n)}(x_1, \ldots, x_n) = \mathbb{E}[\phi(x_1)\ldots\phi(x_n)]$ which may also be computed by a parameter space integral akin to equation (5), or alternatively via free theory Feynman diagrams with propagator given by $G^{(2)}(x, y)$. Foreshadowing, the NNGP is a free field theory that one may turn into an interacting theory by including $1/N$ corrections.

Another result we discuss concerns NN training dynamics. It arises when a NN is trained by continuous-time gradient descent, in which case the dynamics of learning over time t (which is independent of network input x) are given by

$$\frac{\mathrm{d}\phi(x)}{\mathrm{d}t} = \sum_{x' \in D} \Theta(x, x') \frac{\delta L[\phi(x')]}{\delta \phi(x')}, \quad \Theta(x, x') = \sum_{I} \frac{\partial \phi(x)}{\partial \theta_I} \frac{\partial \phi(x')}{\partial \theta_I}, \tag{7}$$

where $\Theta(x, x')$ is called the empirical NTK, D is the full training dataset (not a mini-batch) and L is the loss function evaluated on inputs x'. This equation arises from a short computation that utilizes only chain rules and the gradient descent update rule $\mathrm{d}\theta_I/\mathrm{d}t = -\sum_{x' \in D} \partial L[\phi(x')]/\partial \theta_I$. The empirical NTK governs the gradient descent dynamics of the finite-width NN but is difficult to compute because modern NNs have millions,

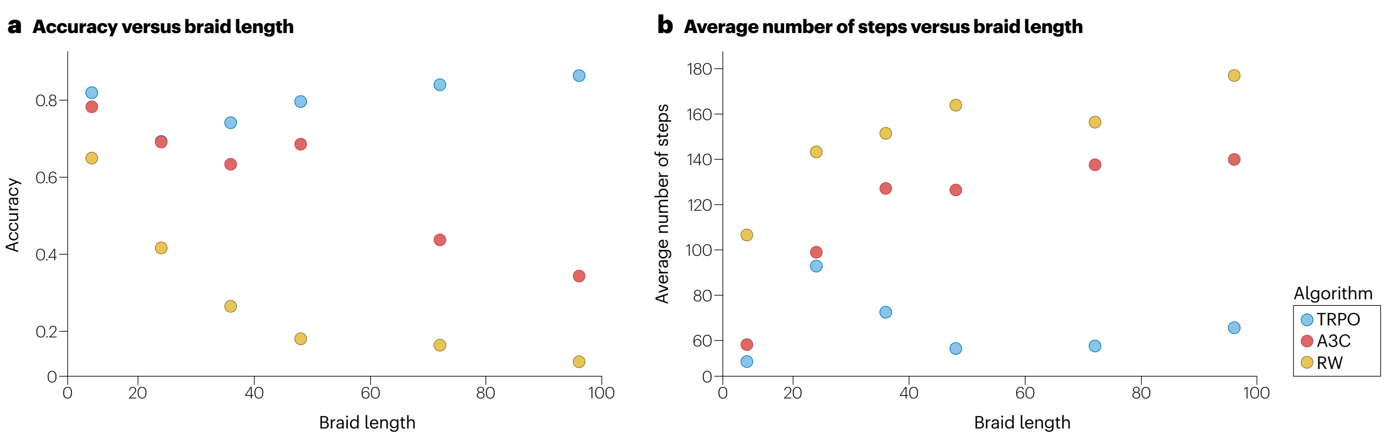

Fig. 2 | Performance comparison of the TRPO, A3C and RW algorithms on the unknotting problem. a, Fraction of unknots whose braid words could be reduced to the empty braid word as a function of initial braid word length ℓ. **b**, Average number of actions necessary to reduce the input braid word to the empty braid word as a function of ℓ. Figure reproduced with permission from ref. 53, CC-BY 4.0. A3C, asynchronous advantage actor–critic; RW, random walker; TRPO, trust region policy optimization.

or billions, of parameters that evolve in time. However, for many architectures, the dynamics simplify in the large-N limit owing to the law of large numbers and a natural linearization, yielding

$$\lim_{N\to\infty} \Theta(x,x') =: \bar{\Theta}(x,x'), \tag{8}$$

where $\bar{\Theta}(x,x')$ is a t-independent deterministic kernel that may be computed once and for all at initialization. $\bar{\Theta}$ may be referred to as the NTK, or the frozen-NTK. If L is the mean-squared error loss, the dynamics for all times may be computed exactly analytically[10] and one may compute the expected prediction of an infinite number of infinite-N neural networks trained to infinite time, allowing a computation that otherwise would not be possible.

NN–FT correspondence

From the discussion in the previous section, it is clear that NNs describe ensembles of functions, akin to statistical field theories in physics. In this sense, a NN architecture and parameter distribution $P(\theta)$ give a new way to define a field theory. We focus on the applications of these ideas within physics, though using them for ML is also interesting (see, for instance, ref. 59).

By the NNGP correspondence, there are many NN architectures that admit an $N\to\infty$ limit in which the NN is drawn from a Gaussian process at initialization. Gaussians are determined by the mean and variance, that is, their one-point correlation function $G^{(1)}(x) = \mathbb{E}[\phi(x)] = \int d\theta\, P(\theta)\, \phi(x)$ (which is often zero) and two-point correlation function $G^{(2)}(x,y)$ defined in equation (5). Any n-point correlation function can be computed in parameter space, generalizing the two-point calculation[57], via

$$G^{(n)}(x_1,\ldots,x_n) = \mathbb{E}_\theta[\phi(x_1)\ldots\phi(x_n)] = \int d\theta\, P(\theta)\phi(x_1)\ldots\phi(x_n). \tag{9}$$

Crucially, these integrals are explicitly computable for some NN architectures, yielding exact correlation functions. In the Gaussian case, $G^{(1)}(x,y)$ and $G^{(2)}(x,y)$ determine an associated Gaussian action, which takes the form in equation (6) in the case of vanishing one-point function. Given that action, the correlators may be computed via the Feynman path integral. Hence, the NNGP correspondence is a duality between a Gaussian process and a free field theory.

This idea can be generalized to a NN–FT correspondence[60,61], which describes interacting, non-Gaussian theories (see ref. 11 for an introduction and summary of the current status of research on this topic). Central to the duality is the idea that field theories may be considered in either the parameter space or function space descriptions. For example, defining a field theory by a NN architecture and parameter distribution makes the parameter description manifest, but determining the associated interacting action $S[\phi]$ that generalizes equation (6) of the NNGP correspondence requires some work. Conversely, one could define a field theory in terms of an action $S[\phi]$, as in QFT classes, and then attempt to determine a NN architecture and parameter distribution that realizes it.

These field theories are usually Euclidean, because most NNs are defined on $\mathbb{R}^n$. If the correlators of the Euclidean theory satisfy the Osterwalder–Schrader axioms[62], the theory can be continued to a Lorentzian signature, defining a NN QFT (ref. 61). For other physics-motivated NN–FT works, see refs. 63–65, and see ref. 66 for work that relates NN dynamics and cosmological dynamics. For ML work on finite-width corrections, see ref. 59 and references therein, and the introduction of ref. 11 for a discussion of more recent literature.

We emphasize that the duality works both ways and a theory may be studied in parameter space even if the action is unknown! In other words, the properties of a field theory are determined simply by the architecture and $P(\theta)$. To illustrate this, consider a toy model, wherein the functions that describe the fields are simply linear functions

$$\phi:\mathbb{R}\to\mathbb{R}, \quad \phi(x) = \theta x, \tag{10}$$

where the slope θ is sampled from some distribution $P(\theta)$. The correlators in equation (9) can be computed exactly in this case, $G^{(n)}(x_1,\ldots,x_n) = \mu_n x_1\ldots x_n$, where $\mu_n = \mathbb{E}_\theta[\theta^n]$ are the moments of $P(\theta)$. The theory has non-Gaussian interactions when $P(\theta)$ is non-Gaussian, which means that its moments are non-zero for at least some $n > 2$. Because the Gaussianity of a theory is a consequence of the central limit theorem (CLT), such interactions arise by breaking the assumptions of the CLT: one can either keep N finite or break statistical independence of parameters in $P(\theta)$. Put differently, the change of $P(\theta)$ in parameter space leads to a modification of the function space action that can manifest itself in interaction terms[11,61].

We will see an example, but before that, let us discuss symmetries[67] in the NN–FT correspondence, using again the example in equation (10). One can see from the exact correlators $G^{(n)}(x_1,\ldots,x_n)$ that the theory is scale invariant, but not translation invariant. If $\mu_{2n+1} = 0$, $P(\theta)$ is even and the theory is parity invariant, because the negative sign in $x\to -x$ may be absorbed into a redefinition of θ. Thus, symmetries of the architecture and parameter space distribution lead to symmetries in function space and vice versa.

To summarize, even without knowing the action, one can do the following:

- determine symmetries of the underlying field theory from the architecture and symmetries of $P(\theta)$,
- introduce interactions by manipulating $P(\theta)$ or the width N such that it breaks the assumptions of the CLT, and
- determine interactions by computing parameter space integrals over $P(\theta)$.

In fact, by inverting the correspondence outlined above, one can find the action from the parameter space description of the NN–FT correspondence by computing couplings in terms of Feynman diagrams, whose vertices are the connected correlators[11].

To illustrate how a change in $P(\theta)$ turns on interactions, we look at a canonical example in QFT known as ϕ^4 theory. This theory adds an interaction term $S_{\text{int}} = \frac{\lambda}{4!}\int d^{n_{\text{in}}}x\phi^4(x)$, where λ is the coupling constant, to the free scalar action $S[\phi] = \int d^{n_{\text{in}}}x\; \phi(x)(\Box + m^2)\phi(x)$, where $\Box = \partial_i\partial^i$ and m is the mass, a specification of the Gaussian process action in equation (6). A NN field theory realizing the free scalar has the architecture

$$\phi_{a,b,c}(x) = \sqrt{\frac{2\mathrm{vol}(B^d_\Lambda)}{\sigma_a^2(2\pi)^{n_{\text{in}}}}}\sum_{i,j}\frac{a_i\cos(b_{ij}x_j + c_i)}{\sqrt{\mathbf{b}_i^2 + m^2}}. \tag{11}$$

with parameters $\theta = \{a,b,c\}$ drawn from

$$P_G(a) = \prod_i e^{-\frac{N}{2\sigma_a^2}a_i a_i}, \quad P_G(b) = \prod_i P_G(\mathbf{b}_i), \quad P_G(c) = \prod_i P_G(c_i). \tag{12}$$

Here, $P_G(\mathbf{b}_i)$ and $P_G(c_i)$ are uniform distributions over an n_{in}-dimensional ball of radius Λ and the interval $[-\pi,\pi]$, respectively. It is perhaps not surprising that the architecture is a normalized plane wave, mirroring the canonical functional form for non-NN treatments in quantum field theory, though other NNGP realizations of the free scalar might exist. Adding the λ-dependent term S_{int} to the function distribution $P[\phi]$ corresponds, in the parameter space description, to a λ-dependent deformation of the parameter distribution

$$P(\theta) = P(a, b, c) = P_G(a)P_G(b)P_G(c)\, e^{-\frac{\lambda}{4!}\int d^dx \phi_{a,b,c}(x)^4}, \tag{13}$$

which only depends on $\theta = \{a, b, c\}$ through their distributions and through $\phi_{a,b,c}$ in equation (11). The existence of interactions in the NN picture arises owing to the breaking of statistical independence via the parameter distribution deformation, violating the assumption of independence in the CLT. The result is standard Euclidean ϕ^4 theory with cut-off Λ, wherein the cutoff Λ appears in the NN description via B^d_Λ, the ball of b-type parameters. This NN–FT recovers the NNGP of equation (6) in the limit $\lambda \to 0$ (ref. 61), which then has only a single non-vanishing moment,

$$G^{(2)}(p) = \frac{1}{p^2 + m^2}, \tag{14}$$

that is, $\lambda \to 0$ recovers a free scalar with mass m.

Metric flows with NNS and the Ricci flow

If a NN is used to represent a metric g_{ij} on a Riemannian manifold M, the training dynamics of the NN corresponds to a flow in the space of Riemannian metrics. We will characterize this flow, following ref. 68.

This idea is partly motivated by recent results on numerical metrics on CY manifolds[69–74]. Physically, a CY manifold is a solution to the leading-order string theory equations of motion, and therefore, they are one of the most-studied types of compactification manifolds for yielding a 4D low-energy effective theory. The particle content and low energy couplings are determined at leading order by the geometry and topology of the CY manifold. Mathematically, a CY manifold is a complex Kähler manifold with vanishing first Chern class $c_1(\mathrm{TM}) = 0$, where TM is the tangent bundle. By the theorem of Yau[75], any CY manifold with fixed Kähler class admits a Ricci-flat Kähler metric known as the CY metric, which is unique due to a theorem of Calabi[76].

The proof is non-constructive, however, and zero non-trivial explicit CY metrics are known for compact M, even though CY manifolds have been studied by mathematicians and string theorists for decades. These facts motivate the use of numerical techniques, for example, via Donaldson's algorithm[77], or more recently via NNs that represent the metric[69–74], which give the current state-of-the-art results. For instance, one could train the NN g_{ij} to minimize a loss given by

$$L[g] = \sum_{i,j} |R_{ij}(g)|^2, \tag{15}$$

which would drive g_{ij} toward the Ricci-flat metric (R_{ij} is the Ricci tensor). In practice, it is more efficient to optimize surrogate losses that build in more structure of the problem, for instance, rephrasing the problem as a second-order differential equation of Monge–Ampère type[75].

Returning to flows, if a NN represents a metric on M and it is trained to approximate the CY metric, then the CY metric is a fixed point of the NN metric flow. Another famous metric flow is the Ricci flow[13] given by

$$\frac{\mathrm{d}g_{ij}(x)}{\mathrm{d}t} = -2\, R_{ij}(x), \tag{16}$$

where $R_{ij}(x)$ is the Ricci tensor. The CY metric is also a fixed point of the Ricci flow because it is Ricci-flat, $R_{ij}(x) = 0$. Are these two flows related? A priori, they look very different because the NN metric flow is defined by gradient descent on a scalar loss functional (gradient flow) and the Ricci flow has a tensorial metric update that is not obviously a gradient flow. However, in his work that proved the 3D Poincaré conjecture, Perelman[12] showed that a t-dependent diffeomorphism of the Ricci flow is a gradient flow defined by

$$\begin{aligned} \frac{\mathrm{d}g_{ij}(x)}{\mathrm{d}t} &= \frac{\delta F[\phi, g]}{\delta g_{ij}(x)} = -2[R_{ij}(x) + \nabla_i \nabla_j \phi(x)] \\ F[\phi, g] &= \int_M (R + |\nabla\phi|^2) e^{-\phi} \mathrm{d}V \end{aligned} \tag{17}$$

where a dilaton field $\phi(x)$ that has its own dynamics dictated by $\delta F/\delta\phi$ is introduced. Because this version of the Ricci flow is a gradient flow, it might be possible to realize it as a NN metric flow.

Motivated by these results, Halverson and Ruehle[68] developed a theory of NN metrics flows, assuming that a NN represents a metric g_{ij} on M and the metric is trained by updating the parameters via gradient descent with respect to a scalar loss functional $L[g]$. The dynamics of the metric at x may be computed either from a finite set of points $\{x_i\}$ sampled according to some measure on M, or in the continuum (infinite-data) limit. In the continuum case, the metric flow is given by

$$\begin{aligned} \frac{\mathrm{d}g_{ij}(x)}{\mathrm{d}t} &= -\int_X \mathrm{d}\nu(x')\, \Theta_{ijkl}(x, x') \frac{\delta L[g(x')]}{\delta g_{kl}(x')} \\ \Theta_{ijkl}(x, x') &:= \sum_I \frac{\partial g_{ij}(x)}{\partial \theta_I} \frac{\partial g_{kl}(x')}{\partial \theta_I}, \end{aligned} \tag{18}$$

where $\Theta_{ijkl}(x, x')$ is the empirical metric-NTK that is derived in the same way as the NTK, but keeps track of the tensorial indices associated to the metric. In practice, a natural choice is to take the measure to be the volume measure, $\mathrm{d}\nu = \mathrm{d}V$. This general equation for NN metric flows is markedly different from Perelman's formulation of Ricci flow. Unlike the Ricci flow, a general NN metric flow is governed by the kernel Θ_{ijkl} that (1) changes the nature of the update equation as it evolves, (2) is non-local, communicating loss fluctuations at x' to metric updates at x, and (3) mixes components of the metric in a non-trivial way owing to the tensor indices.

To obtain Perelman's Ricci flow, one must fix the kernel in time, induce locality and eliminate component mixing. To fix the kernel in time, take an $N \to \infty$ limit in which the NTK becomes frozen (see ref. 78 for various simple architectures that do this). In such a case, the dynamics becomes

$$\frac{\mathrm{d}g_{ij}(x)}{\mathrm{d}t} = -\int_X \mathrm{d}\nu(x')\, \bar{\Theta}_{ijkl}(x, x') \frac{\delta L(g(x'))}{\delta g_{kl}(x')}, \tag{19}$$

where $\bar{\Theta}_{ijkl}$ is the metric-NTK in the frozen $N \to \infty$. We call these dynamics an infinite NN metric flow. Architectures for which the frozen metric-NTK is of the form $\bar{\Theta}_{ijkl}(x, x') = \bar{\Omega}(x)\delta(x - x')\, \delta_{ik}\delta_{jl}$ for some function $\bar{\Omega}$ eliminate non-locality and component mixing in the kernel, simplifying the dynamics to

$$\frac{\mathrm{d}g_{ij}(x)}{\mathrm{d}t} = -\bar{\Omega}(x) \frac{\delta L[g(x)]}{\delta g^{ij}(x)}. \tag{20}$$

We call these dynamics a local NN metric flow. For any architecture giving such a flow (for which an explicit example is presented in ref. 68), Perelman's formulation of the Ricci flow is obtained simply by choosing the loss to be $L[g(x)] = F[g]/\bar{\Omega}(x)$.

NTK theory gives a natural characterization of metric flows induced by NN gradient descent, demonstrating that they are an important generalization of the Ricci flow. In particular, for finite NNs, the metric evolves according to a non-local, time-dependent kernel that mixes components.

Renormalization group flows, optimal transport and Bayesian inference

ML theory has also begun to interface with physics in ways that provide new insights into renormalization in field theory, using ideas from Bayesian inference and diffusion.

Diffusion models have become popular generative models; for example, to generate one-megapixel images of galaxies, one can draw randomly from a 3M-dimensional pixel space (3M because there are three colours per pixel), subject to the constraint that the drawn pixels lead to an image of a galaxy. Achieving this requires drawing from a complicated distribution in pixel space. Diffusion models go the opposite route: starting from images of galaxies, the pixels undergo diffusion and mix until they look like random noise. Inverting this process then leads to a map from noise to images of galaxies. In terms of probability distributions, one can think of the generation process or the inverse diffusion process as changing the distribution from seemingly random pixels to galaxy picture pixels and the flow in distribution space as an (optimal) transport problem.

From a physics perspective, one can think of diffusion as 'destroying' information in the image; removing data points from an inference problem has a similar effect. As a process, this resembles renormalization group flow in QFT, wherein information is lost through coarse graining which forgets about irrelevant operators. Keeping track of these operators leads to the notion of an exact renormalization group flow. This idea is used in three studies[79–81] to connect the exact renormalization group flow to an optimal transport and Bayesian inference problem, respectively. Phrasing in terms of Bayesian inference, wherein the information is added through updating the prior as new observations come in, allows to give an information-theoretic meaning to exact renormalization. By providing new ways to understand and think about renormalization, these works may lead to new insights into the structure of QFTs.

Outlook

In this Perspective, we have discussed zero-error applications of ML to theoretical physics and pure mathematics. These fields are late adopters of ML because they require rigor (by which we mean zero-error results) and interpretability, whereas in general, ML techniques are often stochastic, error-prone and blackbox. The application of ML methods to these fields, thus, requires rethinking and modifying techniques that are readily applied in other natural sciences. We focused on two main directions: making applied ML rigorous and applying theoretical ML. We exemplified the former with conjecture generation and rigorous verification by RL, and the latter using NN theory.

In conjecture generation, a human is brought into the loop to interpret what the ML algorithm has learned and turn it into a conjecture which can then be refined and proven by domain experts. This has been done in supervised ML: if a NN or other algorithm (symbolic regression or decision trees) can learn a high-accuracy map from inputs to the labels, for which no known relation exists, this hints at a new connection. Given that setting up such supervised learning problems can be done very quickly, this allows for scanning theoretical data for new relations. We gave examples of this idea in string theory, algebraic geometry and knot theory.

A second avenue is to use RL for problems in mathematics. The idea is not to attempt to whitebox the ML algorithm but instead to look at episode rollouts of the RL agent to infer the solution strategy learned by the agent. In particular, RL rollouts can be used to obtain (provably correct) truth certificates for decision problems of the type 'Does object O have property X?'. RL is useful in such cases because one can set up an RL algorithm that manipulates object O until property X is manifest and the algorithm reaches a terminal state. The rollouts of the episode that lead to the terminal state are then the truth certificate for the decision problem. Rollouts were used to rule out hundreds of proposed counterexamples to SPC4, and to establish sliceness for new knots.

A different approach is to use ML theory to obtain rigorous results, entirely avoiding the error introduced by applied ML techniques. For instance, owing to the CLT, the statistics of the functions expressed by NNs become tractable in the infinite-parameter regime; they are draws from Gaussian processes. In the context of physics, this leads to a correspondence between NN and statistical field theories, wherein the infinite-parameter regime defines generalized free field theories, and leaving this regime corresponds to turning on interactions. This correspondence provides a new definition of a field theory, motivated by ML theory, and therefore opens a different approach to the study of new and existing field theories. We exemplified a few aspects of this correspondence, including the role of the CLT, the origin of symmetries and the realization of ϕ^4 theory as a NN field theory. Hopefully, this correspondence can lead to a better understanding of non-perturbative QFTs in the future, which at present is one of the major open problems in field theory.

We also discussed the theory of flows in the space of Riemannian metrics induced by gradient descent when the metric is modelled as a NN. This theoretical framework encompasses, for instance, recent empirical results that use NNs as state-of-the-art approximations to CY metrics. In the infinite-parameter limit, the metric flow simplifies and NTK theory may be used. Under some additional architecture assumptions, the flow may be made local, and Perelman's formulation of Ricci flow as a gradient flow is realized as a special case of NN metric flows.

Further progress with ML techniques to obtain zero-error rigorous results could proceed along a number of directions.

One avenue is to extend or systematize the techniques discussed in this Perspective. For instance, conjecture generation and subsequent theorem proving is effective but still haphazard in process; it would be advantageous to have a more systematic framework for conjecture generation, potentially in collaboration with automated theorem-proving systems. Further developments that obtain rigorous results by using ML theory are also natural. In the context of the NN–FT correspondence, one might wish to study conformal field theories, fermions or gauge fields.

A second avenue is to expand the types of ML techniques used in obtaining rigorous results. For instance, we have discussed applications of supervised learning, RL, and ML theory, but deep generative models are notably absent. A natural application is to generate interesting examples that aid in obtaining rigorous results. For example, in certain conjecture generation applications[23,27], the initial ML-inspired conjectures have counterexamples obtained cleverly by humans that are used to refine the conjecture; it would be useful to automate the counterexample-finding process with a generative model. In other applications wherein rigor is sought, it would be useful to have a generative model that could produce examples conditional on certain properties. For instance, in the context of the discussed approach to SPC4, one would like to have a generative model that produces pairs of knots (K_1, K_2) with the same zero surgeries, subject to the constraint that K_2 definitely has a slice obstruction, and K_1 has no known slice obstruction. This was not the case in ref. 7, and such a generative model has the advantage that if K_1 can be shown to be ribbon or slice for any such pair, then SPC4 is false.

Published online: 8 April 2024

References

1. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
2. Carleo, G. et al. Machine learning and the physical sciences. *Rev. Mod. Phys.* **91**, 045002 (2019).
3. Ruehle, F. Data science applications to string theory. *Phys. Rept.* **839**, 1–117 (2020).
4. He, Y. *Machine Learning in Pure Mathematics and Theoretical Physics* (World Scientific, 2023).
5. Athalye, A., Engstrom, L., Ilyas, A. & Kwok, K. Synthesizing robust adversarial examples. *Proc. Mach. Learn. Res.* **80**, 284–293 (2018).
6. Athalye, A., Carlini, N. & Wagner, D. Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. Preprint at https://arxiv.org/abs/1802.00420 (2018).
7. Gukov, S., Halverson, J., Manolescu, C. & Ruehle, F. Searching for ribbons with machine learning. Preprint at https://doi.org/10.48550/arXiv.2304.09304 (2023).
8. Neal, R. M. *Bayesian Learning for Neural Networks*. PhD thesis, Univ. Toronto (1995).
9. Jacot, A., Gabriel, F. & Hongler, C. Neural tangent kernel: convergence and generalization in neural networks. In *32nd Conference on Neural Information Processing Systems* (eds Garnett, R. et al.) 1–10 (2018).
10. Lee, J. et al. Wide neural networks of any depth evolve as linear models under gradient descent. *J. Statist. Mech.* **2020**, 124002 (2019).
11. Demirtas, M., Halverson, J., Maiti, A., Schwartz, M. D. & Stoner, K. Neural network field theories: non-Gaussianity, actions, and locality. *Mach. Learn. Sci. Technol.* **5**, 015002 (2023).
12. Perelman, G. The entropy formula for the Ricci flow and its geometric applications. Preprint at https://doi.org/10.48550/arXiv.math/0211159 (2002).
13. Hamilton, R. S. Three-manifolds with positive Ricci curvature. *J. Differ. Geom.* **17**, 255–306 (1982).
14. Gauthier, T., Kaliszyk, C., Urban, J., Kumar, R. & Norrish, M. TacticToe: learning to prove with tactics. *J. Automat. Reason.* **65**, 257–286 (2018).
15. Szegedy, C. (ed.). *A Promising Path Towards Autoformalization and General Artificial Intelligence* (Springer, 2020).
16. Wu, Y. et al. Autoformalization with large language models. Preprint at https://doi.org/10.48550/arXiv.2205.12615 (2022).
17. Lample, G. et al. HyperTree proof search for neural theorem proving. Preprint at https://doi.org/10.48550/arXiv.2205.11491 (2022).
18. Hales, T. C. Developments in formal proofs. Asterisque Exp. No. 1086, 387–410 (2015).
19. Alama, J., Heskes, T., Kühlwein, D., Tsivtsivadze, E. & Urban, J. Premise selection for mathematics by corpus analysis and kernel methods. *J. Automat. Reason.* **52**, 191–213 (2014).
20. Blanchette, J. C., Greenaway, D., Kaliszyk, C., Kühlwein, D. & Urban, J. A learning-based fact selector for Isabelle/HOL. *J. Automat. Reason.* **57**, 219–244 (2016).
21. Nagashima, Y. Simple dataset for proof method recommendation in Isabelle/HOL (dataset description). In *Intelligent Computer Mathematics: 13th International Conference* 297–302 (ACM, 2020).
22. Piotrowski, B., Mir, R. F. & Ayers, E. Machine-learned premise selection for lean. Preprint at https://doi.org/10.48550/arXiv.2304.00994 (2023).
23. Carifio, J., Halverson, J., Krioukov, D. & Nelson, B. D. Machine learning in the string landscape. *J. High Energy Phys.* **09**, 157 (2017).
24. He, Y.-H. Deep-learning the landscape. *Phys. Lett. B* **774**, 564–568 (2017).
25. Krefl, D. & Seong, R.-K. Machine learning of Calabi-Yau volumes. *Phys. Rev. D* **96**, 066014 (2017).
26. Ruehle, F. Evolving neural networks with genetic algorithms to study the string landscape. *J. High Energy Phys.* **08**, 038 (2017).
27. Davies, A. et al. Advancing mathematics by guiding human intuition with ai. *Nature* **600**, 70–74 (2021).
28. Craven, J., Jejjala, V. & Kar, A. Disentangling a deep learned volume formula. *J. High Energy Phys.* **06**, 040 (2021).
29. Craven, J., Hughes, M., Jejjala, V. & Kar, A. Learning knot invariants across dimensions. *SciPost Phys*. **14**, 021 (2023).
30. Brown, G. et al. Computation and data in the classification of Fano varieties. Preprint at https://doi.org/10.48550/arXiv.2211.10069 (2022).
31. Mishra, C., Moulik, S. R. & Sarkar, R. Mathematical conjecture generation using machine intelligence. Preprint at https://doi.org/10.48550/arXiv.2306.07277 (2023).
32. Cranmer, M. D. et al. Discovering symbolic models from deep learning with inductive biases. In *Advances in Neural Information Processing Systems 33* (NeurIPS, 2020).
33. Silver, D. et al. Mastering the game of Go without human knowledge. *Nature* **550**, 354–359 (2017).
34. Silver, D. et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm (2017). Preprint at https://doi.org/10.48550/arXiv.1712.01815 (2017).
35. Strogatz, S. One giant step for a chess-playing machine. *The New York Times* https://www.nytimes.com/2018/12/26/science/chess-artificial-intelligence.html (2018).
36. Klaewer, D. & Schlechter, L. Machine learning line bundle cohomologies of hypersurfaces in toric varieties. *Phys. Lett. B* **789**, 438–443 (2019).
37. Brodie, C. R., Constantin, A., Deen, R. & Lukas, A. Topological formulae for the zeroth cohomology of line bundles on del Pezzo and Hirzebruch surfaces. *Compl. Manif.* **8**, 223–229 (2021).
38. Brodie, C. R., Constantin, A., Deen, R. & Lukas, A. Index formulae for line bundle cohomology on complex surfaces. *Fortsch. Phys.* **68**, 1900086 (2020).
39. Brodie, C. R., Constantin, A., Deen, R. & Lukas, A. Machine learning line bundle cohomology. *Fortsch. Phys.* **68**, 1900087 (2020).
40. Brodie, C. R. & Constantin, A. Cohomology chambers on complex surfaces and elliptically fibered Calabi-Yau three-folds. Preprint at https://doi.org/10.48550/arXiv.2009.01275 (2020).
41. Bies, M. et al. Machine learning and algebraic approaches towards complete matter spectra in 4d F-theory. *J. High Energy Phys.* **01**, 196 (2021).
42. Halverson, J., Nelson, B. & Ruehle, F. Branes with brains: exploring string vacua with deep reinforcement learning. *J. High Energy Phys.* **06**, 003 (2019).
43. Cole, A., Krippendorf, S., Schachner, A. & Shiu, G. Probing the structure of string theory vacua with genetic algorithms and reinforcement learning. In *35th Conference on Neural Information Processing Systems* (NeurIPS, 2021).
44. Krippendorf, S., Kroepsch, R. & Syvaeri, M. Revealing systematics in phenomenologically viable flux vacua with reinforcement learning. Preprint at https://doi.org/10.48550/arXiv.2107.04039 (2021).
45. Abel, S., Constantin, A., Harvey, T. R. & Lukas, A. String model building, reinforcement learning and genetic algorithms. In *Nankai Symposium on Mathematical Dialogues* (iNSPIRE, 2021).
46. Abel, S., Constantin, A., Harvey, T. R. & Lukas, A. Evolving heterotic gauge backgrounds: genetic algorithms versus reinforcement learning. *Fortsch. Phys.* **70**, 2200034 (2022).
47. Constantin, A., Harvey, T. R. & Lukas, A. Heterotic string model building with monad bundles and reinforcement learning. *Fortsch. Phys.* **70**, 2100186 (2022).
48. Hughes, M. C. A neural network approach to predicting and computing knot invariants. *J. Knot Theory Ramif.* **29**, 2050005 (2020).
49. Sundararajan, M., Taly, A. & Yan, Q. Axiomatic attribution for deep networks. In *Proc. 34th International Conference on Machine Learning* (eds Precup, D. & Teh, Y. W.) Vol. 70, 3319–3328 (PMLR, 2017).
50. Hass, J., Lagarias, J. C. & Pippenger, N. The computational complexity of knot and link problems. *J. ACM* **46**, 185–211 (1999).
51. Kuperberg, G. Knottedness is in NP, modulo GRH. *Adv. Math.* **256**, 493–506 (2014).
52. Lackenby, M. The efficient certification of knottedness and Thurston norm. *Adv. Math.* **387**, 107796 (2021).
53. Gukov, S., Halverson, J., Ruehle, F. & Sułkowski, P. Learning to unknot. *Mach. Learn. Sci. Technol.* **2**, 025035 (2021).
54. Alexander, J. W. A lemma on systems of knotted curves. *Proc. Natl Acad. Sci. USA* **9**, 93–95 (1923).
55. Ri, S. J. & Putrov, P. Graph neural networks and 3-dimensional topology. *Mach. Learn. Sci. Tech.* **4**, 035026 (2023).
56. Gukov, S., Halverson, J., Manolescu, C. & Ruehle, F. An algorithm for finding ribbon bands. *GitHub* https://github.com/ruehlef/ribbon (2023).
57. Williams, C. K. In *Advances in Neural Information Processing Systems* 295–301 (1997).
58. Yang, G. Tensor programs I: wide feedforward or recurrent neural networks of any architecture are Gaussian processes. In *Advances in Neural Information Processing Systems 32* (NeurIPS, 2019).
59. Roberts, D. A., Yaida, S. & Hanin, B. *The Principles of Deep Learning Theory: an Effective Theory Approach to Understanding Neural Networks* (Cambridge Univ. Press, 2022).
60. Halverson, J., Maiti, A. & Stoner, K. Neural networks and quantum field theory. *Mach. Learn. Sci. Tech.* **2**, 035002 (2021).
61. Halverson, J. Building quantum field theories out of neurons. Preprint at https://doi.org/10.48550/arXiv.2112.04527 (2021).
62. Osterwalder, K. & Schrader, R. Axioms for Euclidean green's functions. *Commun. Math. Phys.* **31**, 83–112 (1973).
63. Erbin, H., Lahoche, V. & Samary, D. O. Non-perturbative renormalization for the neural network-QFT correspondence. *Mach. Learn. Sci. Tech.* **3**, 015027 (2022).
64. Grosvenor, K. T. & Jefferson, R. The edge of chaos: quantum field theory and deep neural networks. *SciPost Phys.* **12**, 081 (2022).
65. Banta, I., Cai, T., Craig, N. & Zhang, Z. Structures of neural network effective theories. Preprint at https://doi.org/10.48550/arXiv.2305.02334 (2023).
66. Krippendorf, S. & Spannowsky, M. A duality connecting neural network and cosmological dynamics. *Mach. Learn. Sci. Technol.* **3**, 035011 (2022).
67. Maiti, A., Stoner, K. & Halverson, J. in *Machine Learning in Pure Mathematics and Theoretical Physics* Ch. 8, 293–330 (2023).

68. Halverson, J. & Ruehle, F. Metric flows with neural networks. Preprint at https://doi.org/10.48550/arXiv.2310.19870 (2023).
69. Anderson, L. B. et al. Moduli-dependent Calabi-Yau and SU(3)-structure metrics from machine learning. *J. High Energy Phys.* **05**, 013 (2021).
70. Douglas, M. R., Lakshminarasimhan, S. & Qi, Y. Numerical Calabi-Yau metrics from holomorphic networks. In *Proc. 2nd Mathematical and Scientific Machine Learning Conference* Vol. 145, 223–252 (PMLR, 2022).
71. Jejjala, V., Mayorga Pena, D. K. & Mishra, C. Neural network approximations for Calabi-Yau metrics. *J. High Energy Phys.* **08**, 105 (2022).
72. Larfors, M., Lukas, A., Ruehle, F. & Schneider, R. Learning size and shape of Calabi-Yau spaces. In *Fourth Workshop on Machine Learning and the Physical Sciences* (2021).
73. Larfors, M., Lukas, A., Ruehle, F. & Schneider, R. Numerical metrics for complete intersection and Kreuzer–Skarke Calabi–Yau manifolds. *Mach. Learn. Sci. Tech.* **3**, 035014 (2022).
74. Gerdes, M. & Krippendorf, S. CYJAX: a package for Calabi-Yau metrics with JAX. *Mach. Learn. Sci. Tech.* **4**, 025031 (2023).
75. Yau, S.-T. On the Ricci curvature of a compact Kähler manifold and the complex Monge-Ampére equation, I. *Commun. Pure Appl. Math.* **31**, 339–411 (1978).
76. Calabi, E. *On Kähler Manifolds with Vanishing Canonical Class* 78–89 (Princeton Univ. Press, 2015).
77. Donaldson, S. K. Some numerical results in complex differential geometry (2005). Preprint at https://doi.org/10.48550/arXiv.math/0512625 (2005).
78. Yang, G. Tensor programs II: neural tangent kernel for any architecture. Preprint at https://doi.org/10.48550/arXiv.2006.14548 (2020).
79. Cotler, J. & Rezchikov, S. Renormalization group flow as optimal transport. *Phys. Rev. D* https://doi.org/10.1103/physrevd.108.025003 (2023).
80. Berman, D. S. & Klinger, M. S. The inverse of exact renormalization group flows as statistical inference. Preprint at https://doi.org/10.48550/arXiv.2212.11379 (2022).
81. Berman, D. S., Klinger, M. S. & Stapleton, A. G. Bayesian renormalization. *Mach. Learn. Sci. Technol.* **4**, 045011 (2023).

Acknowledgements
S.G. is supported in part by a Simons Collaboration Grant on New Structures in Low-Dimensional Topology and by the Department of Energy grant DE-SC0011632. J.H. is supported by National Science Foundation CAREER grant PHY-1848089. F.R. is supported by the National Science Foundation grants PHY-2210333 and startup funding from Northeastern University. The work of J.H. and F.R. is also supported by the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions).

Author contributions
The authors contributed equally to all aspects of the article.

Competing interests
The authors declare no competing interests.

Additional information
Peer review information *Nature Reviews Physics* thanks Marika Taylor, Sven Krippendorf and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.