Slide1: Hello everyone. Today I'm going to talk about a paper called "advancing mathematics by guiding human intuition with AI". This is one of the most excited work I've read recently. Acutually this is my first time to do the paper sharing in the group meeting. I will try my best to present this work well and I hope all of you can enjoy in this work.

Slide2: There are five parts of today's presentation. The first part is a short introduction about the paper background. The second part is the motivation of this work. The third part is the explanation of the proposed framework and the details of techiniques used in this work. The fourth part is an demonstration of an example. The last part is the conclusion and contribution analysis of this work.

Slide3: This article is published on the famous journal Nature. This paper is published online on December 1st in 2021 and featured the front cover of Nature issued on December 2nd 2021. This article is open accessible online and you don't have to subscribe the nature. I attached the link below. If you are interested in, please feel free to check it.

Slide4: There are several authors of this work. They are mainly from three organizations, deepmind, unversity of oxford and univerity of sydney. I just want to give some detailed information about the first author and the corresponding authors. The first author is Alex Davies. He is the Founding lead of AI for Maths of DeepMind. The corresponding authors are very famous. Demis is the Founder and CEO of DeepMind. Pushmeet is the Head of Research of DeepMind.

Slide5: Let us review two basic concepts before we go the details of the paper. The first one is the conjecture. Conjectures are statements that are suspected to be true but have not been proven to hold in all cases. And A theorem is a statement that has been proved or can be proved.

The work of mathematicians is to find some relations between two objects. For example, the mathematicians studied the relationship between the hypotenuse and sides of the right triangles. Finally they found the pythagorean theorem.

Slide6: But this is not the all cases. The prerequisite of finding the pythagorean theorem is that the mathematicians think there's some relations between the hypotenuse and sides of the right triangles. However, sometimes we may think there's no relation between two objects, but acutally there is. For example, here we use x as a positive integer and pi(x) is the number of primer numbers smaller than or equal to x. For many years, the mathematicians believed that the occurrences of prime numbers has no distribution. But actually there is. Nowadays, we all know that pi(x) is asymptotically equal to the x over log x, and that is the famous prime number theorem.

Moreover, sometimes there is no relation between two objects but the mathematicians believer there is. Then they will waste a lot of time on a unpromising direction.

So what can the AI do in this process. Machine learning is good at finding a relations. The main idea of this paper is that they want to use machine learning help mathematicians to confirm the existence of the relation and determine which quantities are useful for formulating conjectures.

Slide7: Now we come to the formulation part of the paper. So what we care about in mathematics is that we first have a mathematical object called z. X(z) is some features of z and Y(z) is some other features of z. The mathematicians always want to find some relations between X(z) and Y(z). In other word, we may hypothesize that there exists a function f such that $f(X(z)) \approx Y(z)$.

Let me give you an example to illustrate the formulation here. The mathematical objects we want to study are the convex polyhedrons. So z is a convex polyhedron. We define V is the number of vertices of a polyhedron, E is the number of edges, Vol is the volume, Sur is the surface area and F is the number of faces. Then we can define X(z) as a vector of V, E, Vol and Sur. And Y(z) as a vector of F.

Actually you can define the X(z) and Y(z) respectively with other choices. I just want to give you an illustration here.

Slide8: This is the flowchart of the proposed framework in this paper. The grey rectangle are the mathematcians steps, that is the work of mathematcians. The blue rectangle are the computational steps, that is the work of computers. The red stars is the traditional way to prove a theorm. I will explain the traditional way first.

The mathematicians first hypothesize that there might be a relation between X(z) and Y(z). Then they use the computers to generate lots of data, that is lots of pairs of X(z) and Y(z). Here Pz is the distribution of z. Then, if they can observe some relations between X(z) and Y(z) in the dataset, they can formally propose a conjecture candidate f prime. If they cannot observe some relations between X(z) and Y(z), they may resample another dataset. If they still cannot observe any relations, they may reformulate the hypothesis, that is they think there might be no relation between X(z) and Y(z).

The problem in this process is in the step from Generate data to conjecture candidate. When X(z) and Y(z) have high dimensions, it's not easy for mathematicians to notice the existence of a relation by just observing the data. As a consequence, we can use the a machine learning model here. We can use the X(z) as the features and Y(z) as the labels. We can simply train a supervised learning model to confirm that there is a relation between X(z) and Y(z). If the accuracy of the model is higher than random guessing, then we think there should be some relations between X(z) and Y(z).

Another question is that sometimes X(z) has high dimensions, and we don't know which parts of X(z) has influence on predicting Y(z). So to solve this problem, the author proposed to use the attribution interrogate to determine which parts of X(z) are useful. I will elaborate on the details of the supervised model and attribution interrogate in the following slides.

So what we can conclude from this flowchart is that the proposed framework is useful for quickly finding a conjecture candidate but not proving a theorem.

Slide9: In this slide, let us talk about the details of the supervised learning model with the convex polyhedrons example. As we defined previously, z is the convex polyhedron, X(z) is a vector of the number of vertices and edges, as well as the volume and surface area of a convex polyhedron. Y(z) is the a vector of the number of faces. We hypythesize that there is a relation between X(z) and Y(z), that is we want to find a function f such that f(X(z)) is approximately equal to the Y(z).

This is a typical supervised learning. We can use the X(z) as the features and the Y(z) as the labels. Before fitting the model, we need a dataset first. We collect many polyhedrons and record their number of vertices, edges, and faces, as well as volume and surface area. Then we use the model to fit our dataset. We can simply use a linear regression model here.

We finally found that the inner product of X(z) and (-1, 1, 0, 0) is equal to Y(z). That is –V + E + 2 = F. If we rewrite the formula as 2 = F + V – E, you got the Euler's polyhedron formula.

Slide10: In this case, the volume and surface area of a convex polyhedron are not the predictors for the number of faces, since the weights of them is zero. This statements is obviously true since we simply use a linear regression model here. Once the X(z) and Y(z) both have high dimensions, we may use a deep model. It's difficult for us to interpret which features in X(z) are useful. But we still need to know this because we need them to formulate the conjecture.

In this paper, the authors exploited the attribution interrogate.

Slide11: The attribute interrogate is popularly used in computer vision. The motivation here is that we want to know which features have more influence on the result. For example, for the handwritten digits, we want to know Which pixels can tell you that this is number one? For the cats from the imagenet, we want the trained model to explain why this is a picture of cat? Why this is not

Slide12: To solve this question, we can use a very naive mathematical idea called gradient saliency. Let us consider a very simple neural network. It has only one layer from the input to the output, and use an activation function to get the final results.

So here we consider a classification task and use the X as the input, Y as the output and sigma as the activation function. X has dimension of 4 and Y has dimension of 3. Then, if we use the matrix representation, we can write Y is equal to the transpose of w times X plus b and activated by the activation function.

Specifically, I can write the probability to class a1 is $w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b_1$ and activated by the activation function. Then we can calculate the partial derivatives of a1 respect to x1, x2, x3 and x4. a1 is most sensitive to the change of xi if partial derivatives of a1 respect to xi has the largest value. In other word, xi is most influential features for a1. With this technique, we can easily know which parts of X are useful for predicting Y only if this function $Y = \sigma(w^T \cdot X + b)$ is differentiable.

Slide13: In the following slides, I will show how the proposed framework can help mathematicians in proposing a conjecture. The framework help the mathematicians find a conjecture in the knots theory of topology.

Slide14: Before we go to the details of the experiment parts, I want to talk some basic terms in the knots theory. Knots are ropes in 3D world where two ends are joined. All of figures here are examples of knots. The second term is the Knots equivalency. That is if you can change a knot to other knot without using scissors to cut it off, then they are considered as equivalent. You just pull the knots and make it to another knots. Then they are considered as equivalent. The last term is the Knot Invariants: which are the "quantities" that are the same for equivalent knots. There are two types of Knot Invariants, Geometric and Algebraic. This might not make too much sense. But we can take an analogy of the isomorphism we learned in the discrete mathematics. For example, if two graph are isomorphic, then the number of edges connected to the corresponding vertices is an invariant. Otherwise, they will not be isomorphic. For example, these two graphs are not isomorphic since this vertex connected to 2 edges and this vertex connected to 3 edges. These two graphs are isomorphic, then the number of edges connected to the corresponding vertices is same. Each vertex is connected to two edges.

So the knots invariants is the similar idea. If two knots are considered as equivalent, then there're some quantities which are same.

Slide15: In this case, the mathematicians want to find some relations between the geometric invariants and the algebraic invariants. So here z is the knots, and X(z) is the geometric invariants, Y(z) is the algebraic invariants. The mathematicians think there might be some undiscovered relations between the geometric and algebraic invariants.

Specifically, they think that we can use the geometric invariants to predict the value of signature, which is an algebraic invariants.

Slide16: They use all the geometric invariants in the experiments. Here im is to take the imaginary part of a complex number. Re is to take the real part of a complex number. X(z) is a vector of all the geometric invariants. <Im(Meridional translation), Longitudinal translation, Re(Meridional translation), Im(Short geodesic), Injectivity radius, Cusp volume, Symmetry group, Torsion degree, Re(Short geodesic), Volume, Chern–Simons, Adjoint torsion degree>. And Y(z) is a vector of the signature.

They collects many knots and build a dataset. Then use a MLP to fit this dataset and get the accuracy score of 83%. I checked some open repos in GitHub and found that we can even improve the accuracy to 97%. This results confirms that there must be some relation between the geometric invariants and the signature.

Slide17: The question is that we put all of the geometric invariants in X(z). We don't know which invariants are really useful. So we use the attribution interrogate here. We can see in the left figure, the imaginary part of meridional translation, longitudinal translation, and real part of meridional translation have much more impacts on predicting the signature. So these can give mathematicians an intuition that the conjecture candidate should mostly consider these three.

The horizontal axis of the right figure is the real part of the meridional translation, the vertical axis is the value of signature. The color represents the value of longitudinal translation. We can see that this figure has obvious regularity. It visually shows that there must be some relations between the signature the these geometric invariants.

Slide18: With the intuition guided by the AI, mathematicians can formulate the conjecture now. Firstly, let us define some notations.

Miu , lambda

In mathematics, a hyperbolic link is a link in the 3-sphere with complement that has a complete Riemannian metric of constant negative curvature, i.e. has a hyperbolic geometry. A hyperbolic knot is a hyperbolic link with one component.

- Im(Meridional translation), Longitudinal translation, Re(Meridional translation) are the influential predictors
- We define μ = Meridional translation and λ = Longitudinal translation
- We define slope(K) = Re(λ/μ), where K is a knot.
- We define signature as $\sigma(K)$, volume as vol(K) and the injectivity radius inj(K)
- Conjecture candidate: There exist constants c1 and c2 such that, for every hyperbolic knot K,

$$|2\sigma(K) - \mathrm{slope}(K)| < c_1 \mathrm{vol}(K) + c_2$$

- But found a counterexample in the dataset
- Final Conjecture: There exists a constant c such that, for any hyperbolic knot K,

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

- This conjecture is proved and published on the paper "The signature and cusp geometry of hyperbolic knots"

What we can see here is that the final conjecture is not only composed by the first three influential features. It also includes the injectivity radius, and the cusp volume, which are the fifth and sixth influential features. So this reflects the title of this paper. AI can only guided the intuition of the mathematicians but not give a conjecture directly. It can only tell the mathematicians that you should take insight of the first three but not only the first three.

Slide19:
Slide20:
Slide21:

Slide22: Ensure what contribution you want to make. You want to propose new algorithms of machine learning, improve performance of typical machine learning tasks, or apply machine learning to new domains. Never be satisfied with current achievements. Even though the computers help us prove the 4 color theorem and many other famous theorems, it still has contribution in other ways.
The machine learning model is really simple in this paper, it has about 100 hundred lines of code. It's much shorter than the assignments we did in undergraduate. But it's still extremely useful. When we want apply the machine learning to other domains, we don't have to use complicated models or techniques. Sometimes a naïve idea can get impressive results.

Slide23: Here're the references of today's presentation. The first one is the paper itself. The second on is the textbook about the knots theory.

Slide24: That's all of today's presentation. If you have any questions about the paper or comments for my presentation, please feel free to discuss with me.