Machine Learning for String Vacua

FABIAN RUEHLE (CERN & UNIVERSITY OF OXFORD)

Recent Developments in Strings and Gravity

12.09.2019

Based on:

- **Computational Complexity of Vacua and Near-Vacua in Field and String Theory**
  w/ Jim Halverson [1809.08279]
- **Branes with Brains: Exploring String Vacua with Deep Reinforcement Learning**
  w/ Jim Halverson and Brent Nelson [1903.11616]

Recap - Machine Learning in String Theory

In June 2017, with 2 weeks, 4 groups proposed independently to use ML in ST

[He `17; Krefl,Seung `17; Ruehle `17; Carifio,Halverson,Krioukov,Nelson `17]

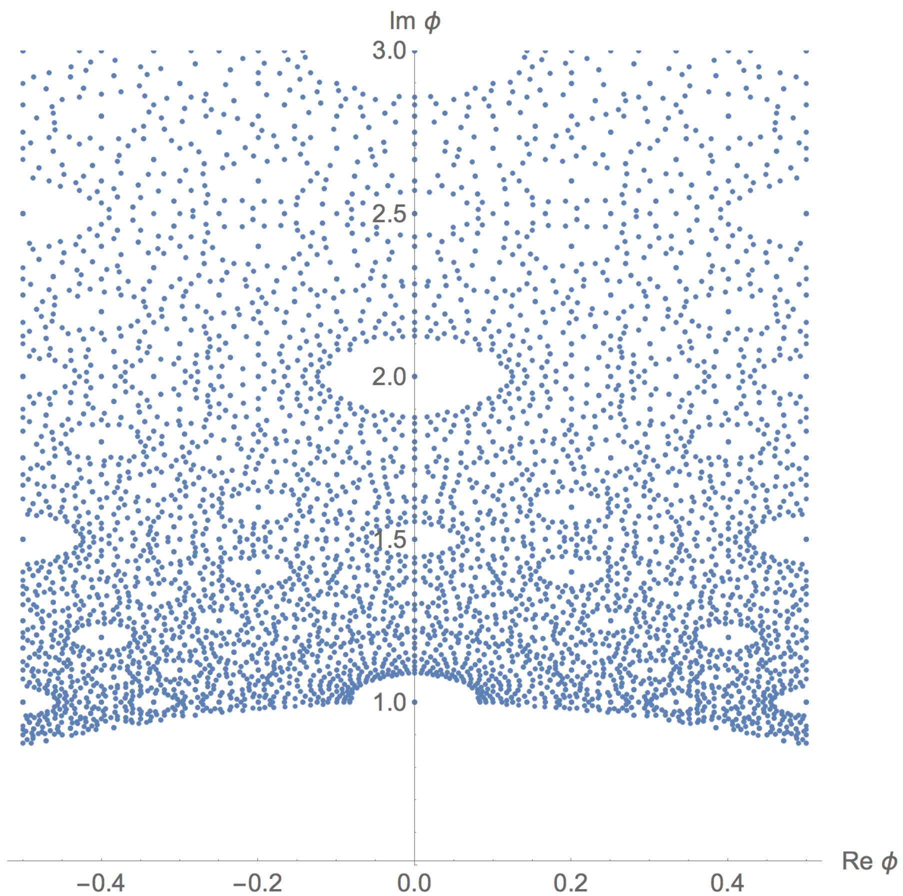
Recap - Machine Learning in String Theory

In June 2017, with 2 weeks, 4 groups proposed independently to use ML in ST

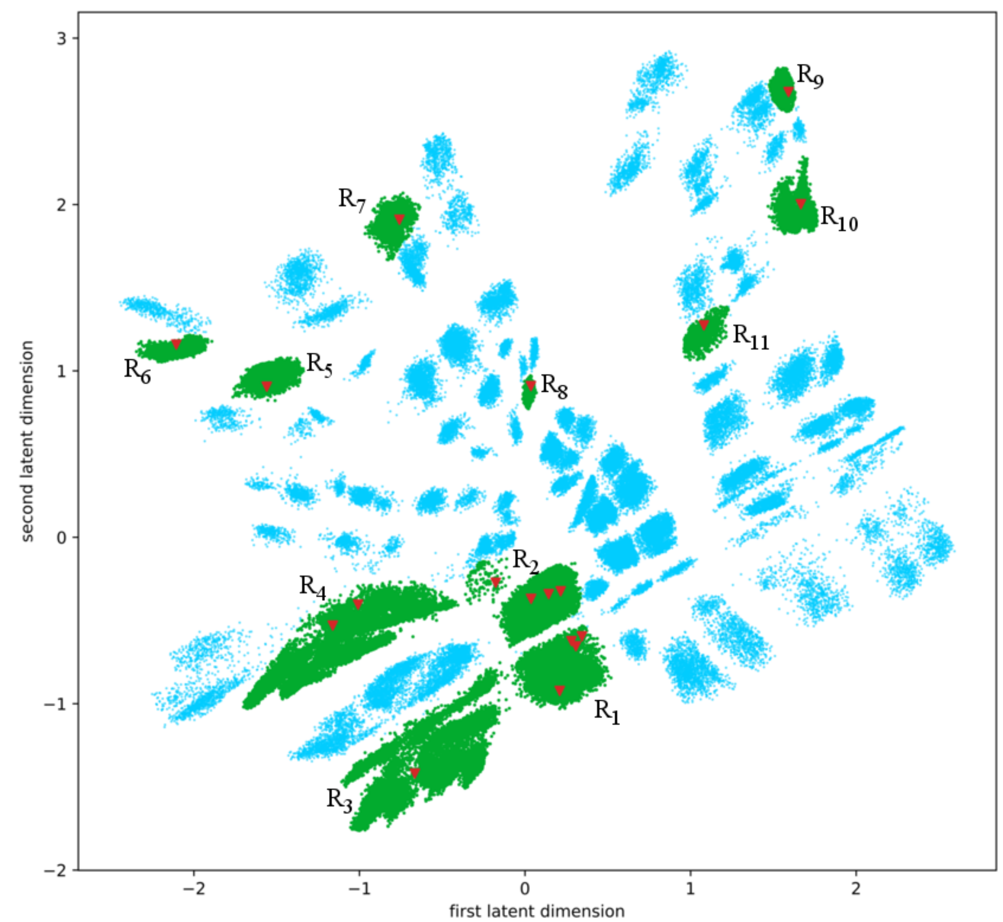
[He `17; Krefl,Seung `17; Ruehle `17; Carifio,Halverson,Krioukov,Nelson `17]

Structure of vacua (Unsupervised ML)

- Clustering, Feature extraction
- Topological data analysis



[Cole,Shiu `17,`18]



[Mutter,Parr,Vaudrevange `18]

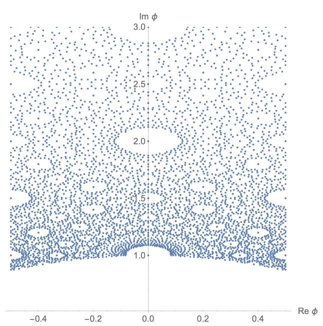
Recap - Machine Learning in String Theory

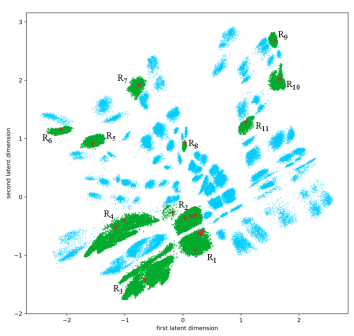
In June 2017, with 2 weeks, 4 groups proposed independently to use ML in ST

[He `17; Krefl,Seung `17; Ruehle `17; Carifio,Halverson,Krioukov,Nelson `17]

Structure of vacua (Unsupervised ML)

- Clustering, Feature extraction
- Topological data analysis





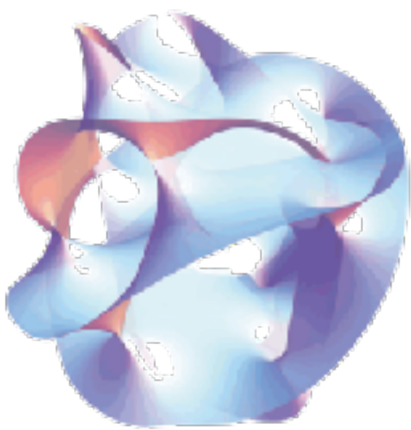
Recap - Machine Learning in String Theory

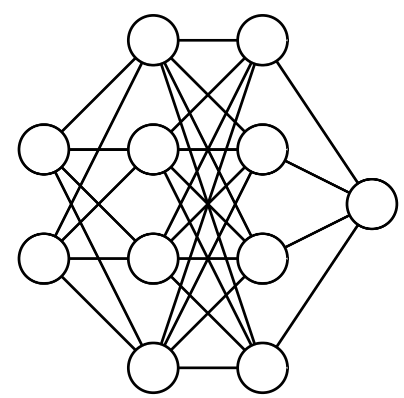
In June 2017, with 2 weeks, 4 groups proposed independently to use ML in ST

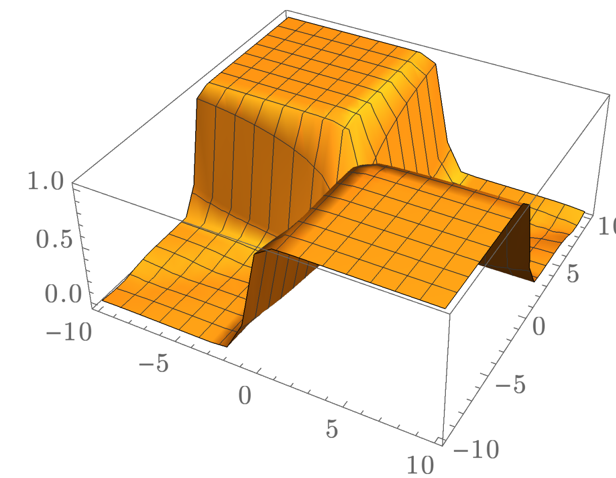
[He `17; Krefl,Seung `17; Ruehle `17; Carifio,Halverson,Krioukov,Nelson `17]

Bypass Computations (Supervised ML)
- Deep neural networks
- Support vector machines

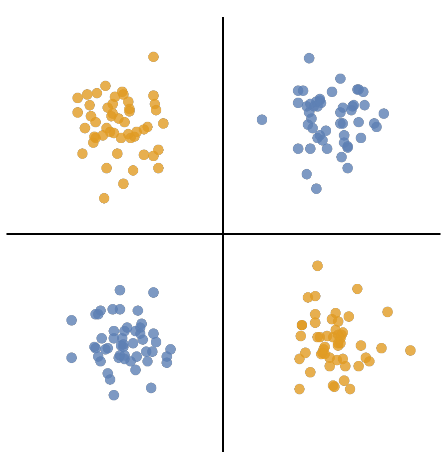
[Wang,Zhang `18; Bull,He,Jejjala,Mishra `18; Klaewer,Schlechter `18; He `18; Jejjala,Kar,Parrikar `19; Bull,He,Jejjala,Mishra `19; He,Lee `19]

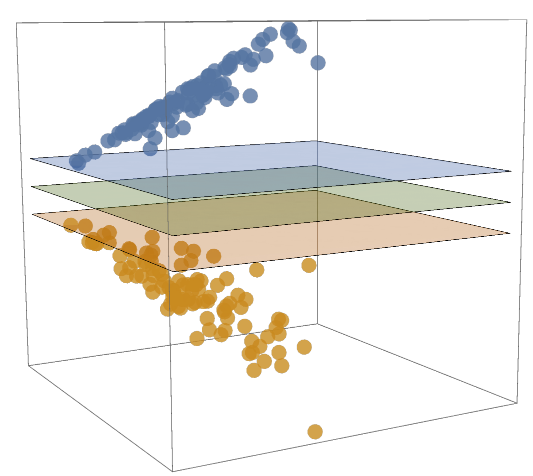


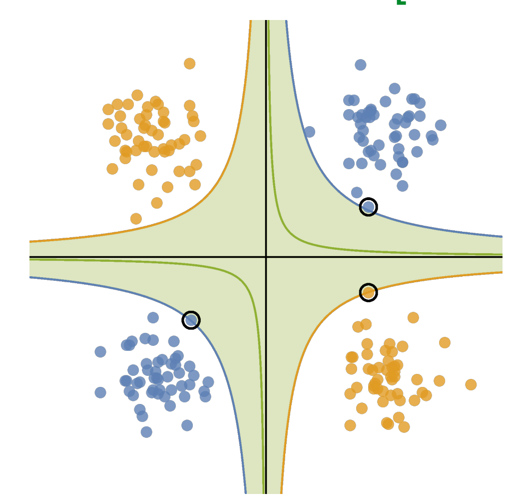




[Ruehle `17]



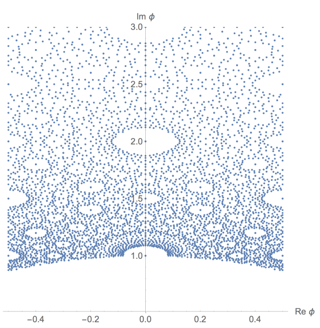


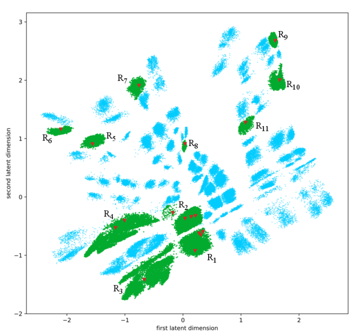


Recap - Machine Learning in String Theory

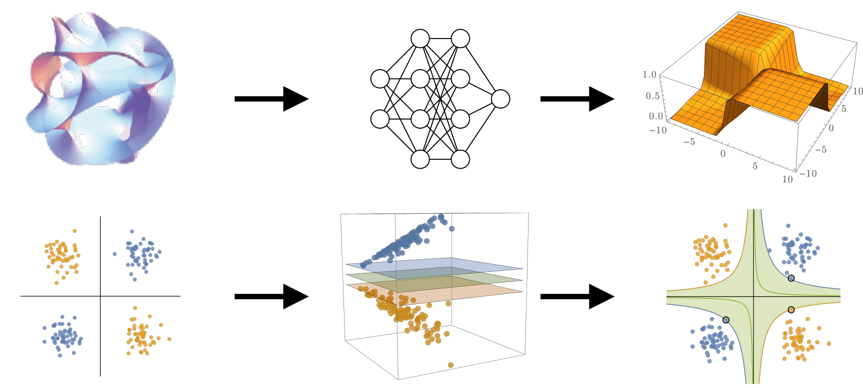
In June 2017, with 2 weeks, 4 groups proposed independently to use ML in ST
[He `17; Krefl,Seung `17; Ruehle `17; Carifio,Halverson,Krioukov,Nelson `17]

Structure of vacua (Unsupervised ML)
- Clustering, Feature extraction
- Topological data analysis





Bypass Computations (Supervised ML)
- Deep neural networks
- Support vector machines



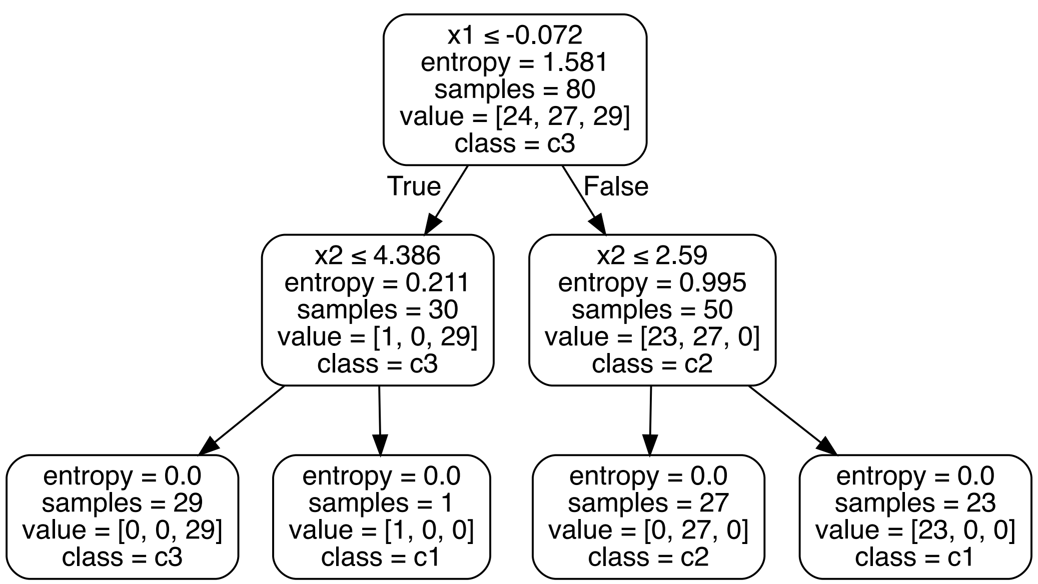
Recap - Machine Learning in String Theory

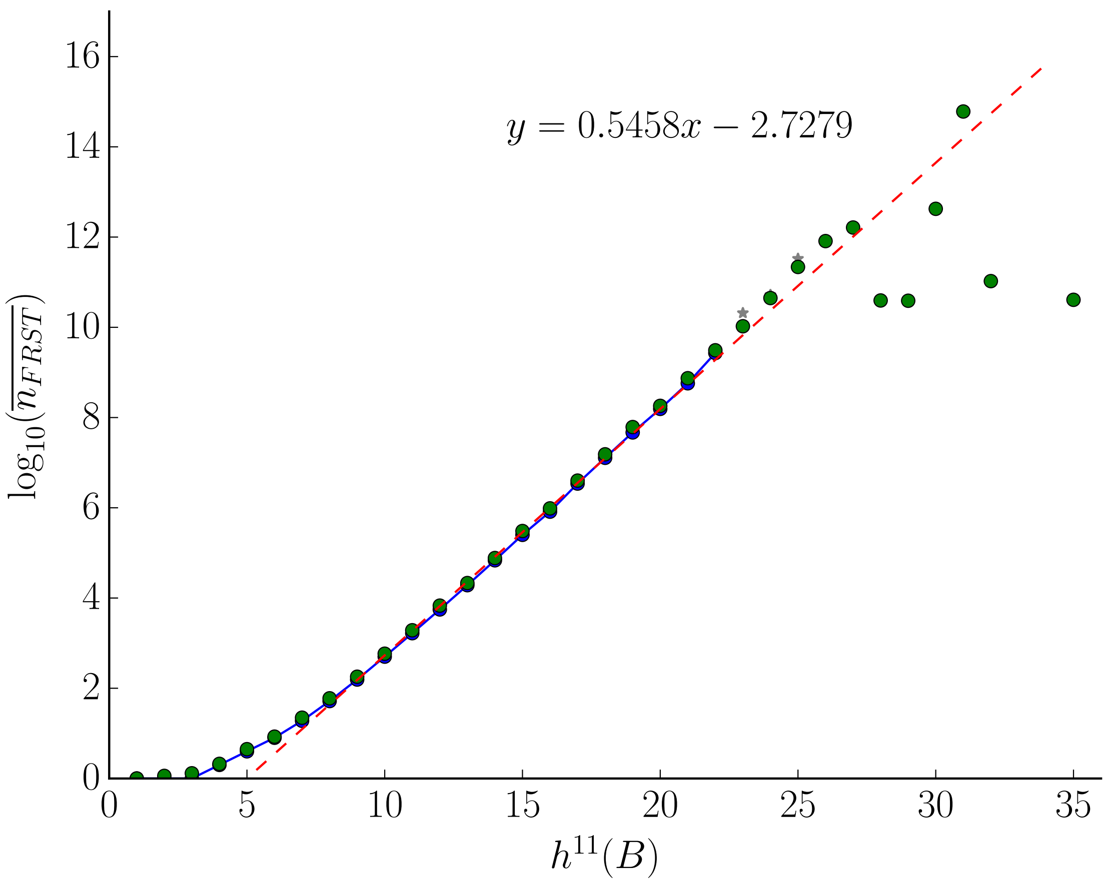
In June 2017, with 2 weeks, 4 groups proposed independently to use ML in ST

[He `17; Krefl,Seung `17; Ruehle `17; Carifio,Halverson,Krioukov,Nelson `17]

Conjecture generation (Intelligible AI)

- Decision Trees
- Regression





[Carifio,Halverson,Krioukov,Nelson `17; Altman,Carifio,Halverson,Nelson `18]

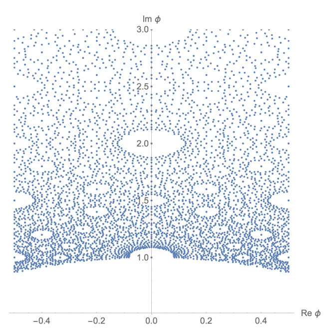
Recap - Machine Learning in String Theory

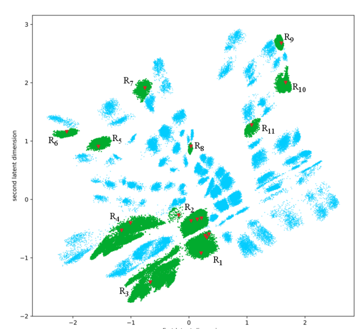
In June 2017, with 2 weeks, 4 groups proposed independently to use ML in ST

[He `17; Krefl,Seung `17; Ruehle `17; Carifio,Halverson,Krioukov,Nelson `17]

Structure of vacua (Unsupervised ML)

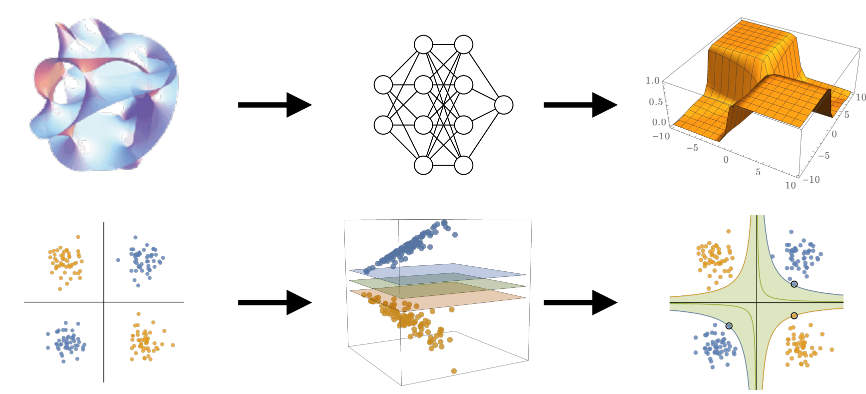
- Clustering, Feature extraction
- Topological data analysis





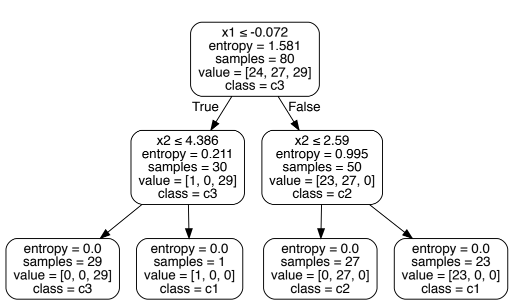
Bypass Computations (Supervised ML)

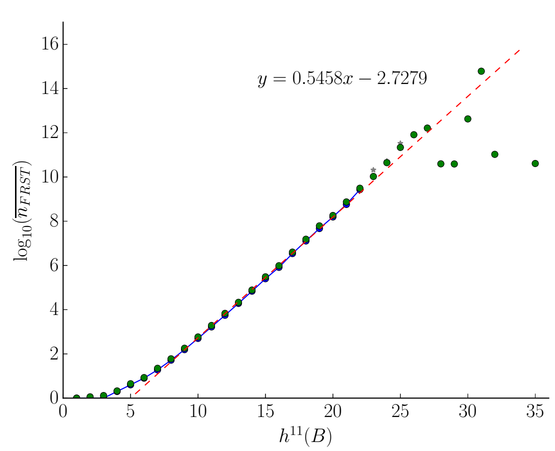
- Deep neural networks
- Support vector machines



Conjecture generation (Intelligible AI)

- Decision Trees
- Regression





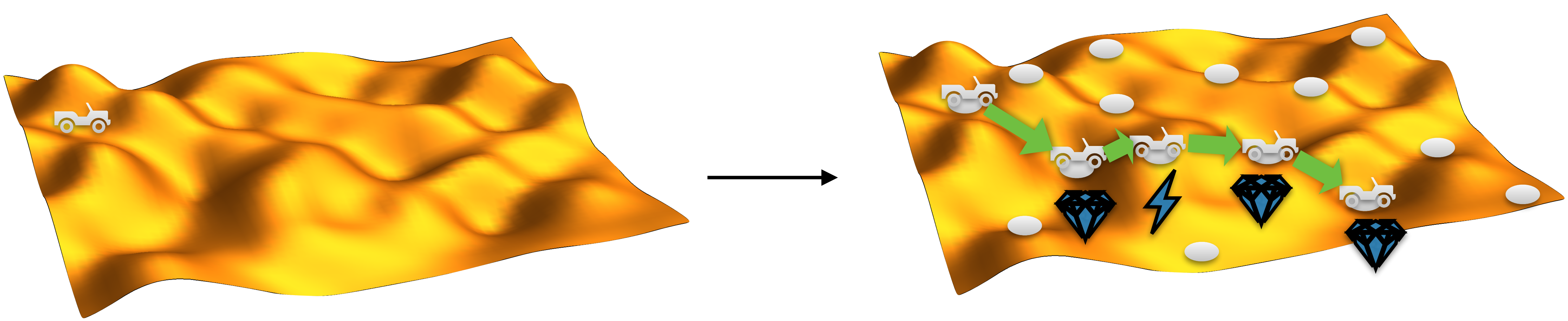
Recap - Machine Learning in String Theory

In June 2017, with 2 weeks, 4 groups proposed independently to use ML in ST

[He `17; Krefl,Seung `17; Ruehle `17; Carifio,Halverson,Krioukov,Nelson `17]

Search the landscape (Semi-supervised ML)

- MC tree searches
- Dynamic programming in MDP
- Reinforcement Learning



[Halverson,Nelson,Ruehle `17]

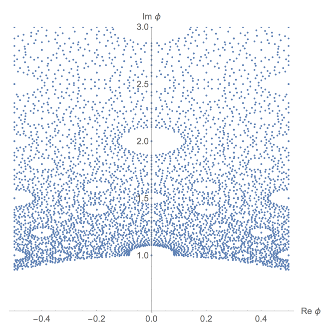
Recap - Machine Learning in String Theory

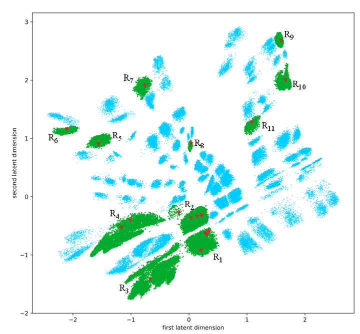
In June 2017, with 2 weeks, 4 groups proposed independently to use ML in ST

[He `17; Krefl,Seung `17; Ruehle `17; Carifio,Halverson,Krioukov,Nelson `17]

Structure of vacua (Unsupervised ML)

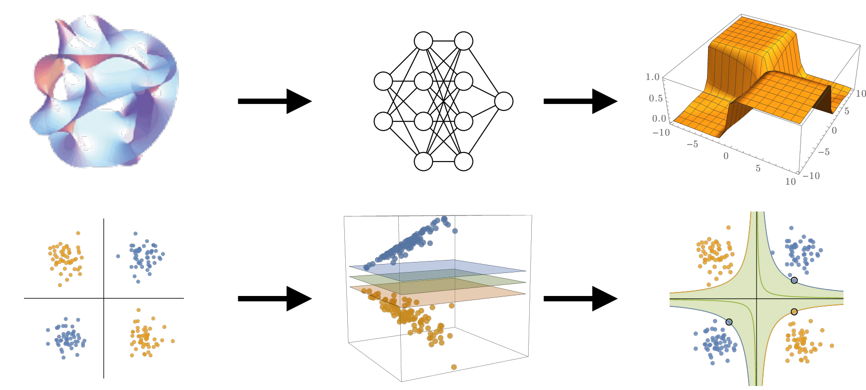
- Clustering, Feature extraction
- Topological data analysis





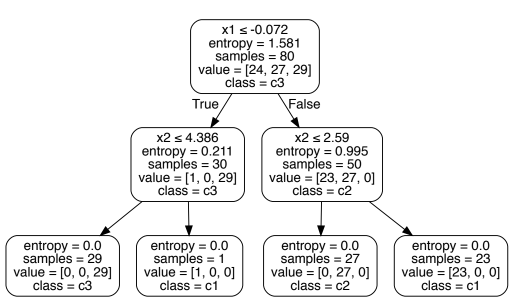
Bypass Computations (Supervised ML)

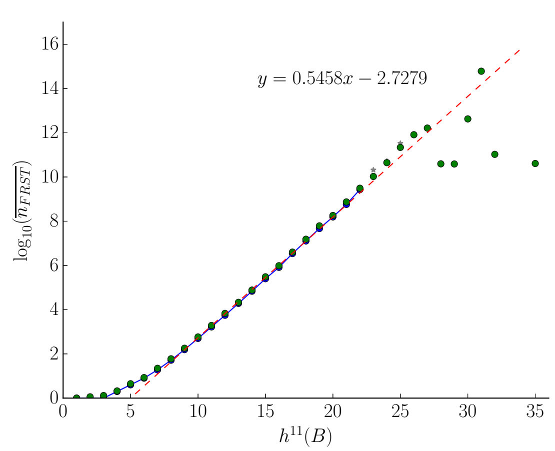
- Deep neural networks
- Support vector machines



Conjecture generation (Intelligible AI)

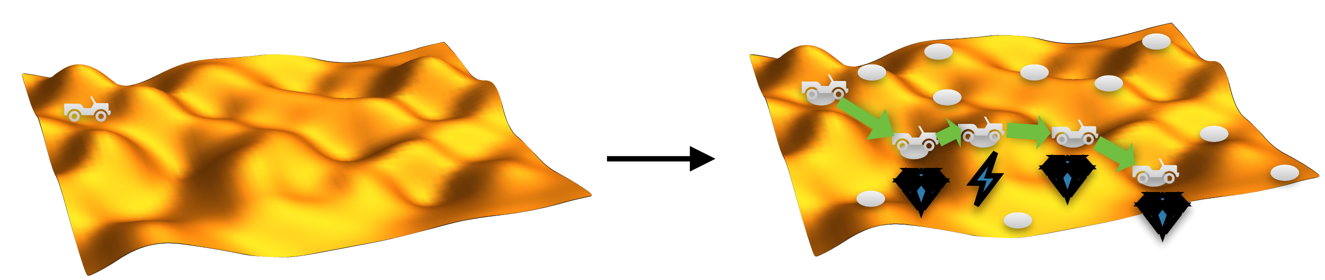
- Decision Trees
- Regression





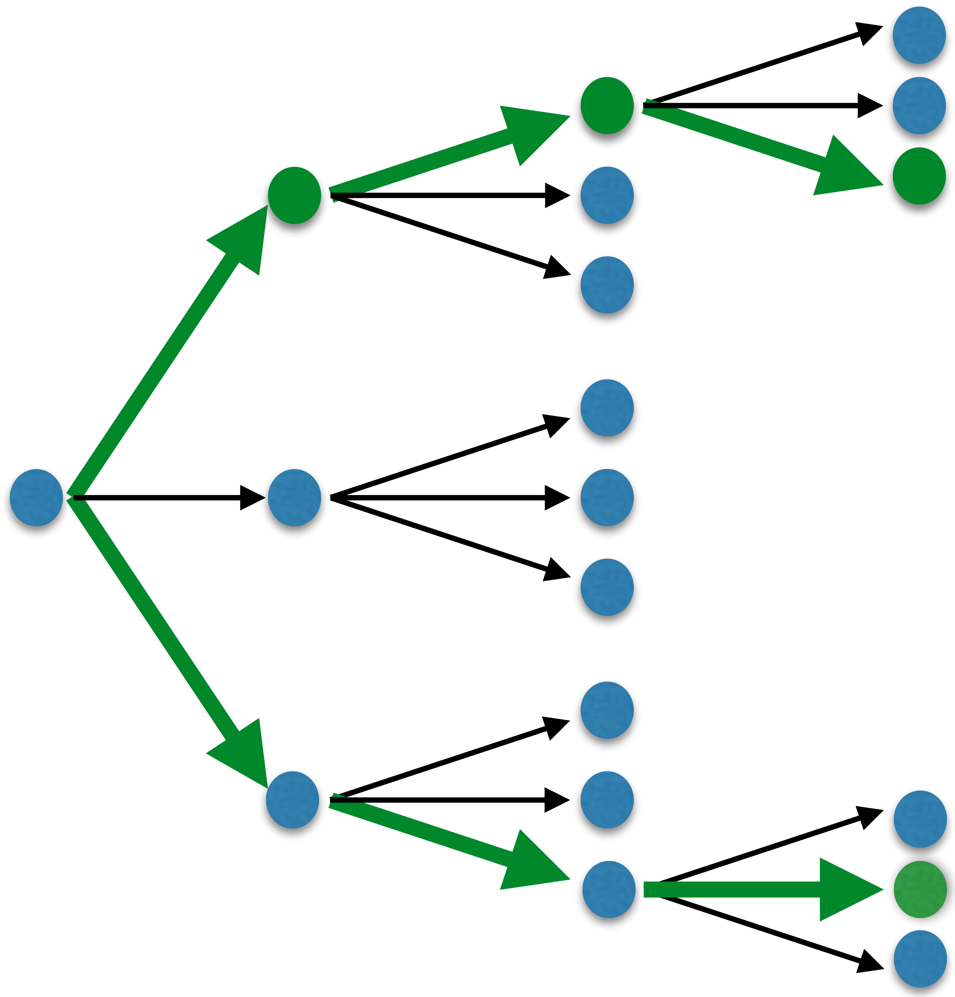
Search the landscape (Semi-supervised ML)

- MC tree searches
- Dynamic programming in MDP
- Reinforcement Learning



Outline

- Computational complexity and decidability
  - Intro
  - Computationally hard problems in string theory
- Machine learning the landscape of IIA toroidal orientifolds
- Conclusion



$$\mathrm{P} \stackrel{?}{=} \mathrm{NP}$$

Computational Complexity and Decidability

Definitions

Problem: A problem $F : I \to B$ is a map from instances to outputs

Dec. Problem: A problem where $B = \{\text{yes, no}\}$

Usually easier to make statements for dec. problems

Definitions

Problem: A problem $F : I \to B$ is a map from instances to outputs

Dec. Problem: A problem where $B = \{\text{yes, no}\}$

Usually easier to make statements for dec. problems

Often, problems can be reformulated as dec. problems with additional parameters, e.g.:

Problem:

Find the minimum of a scalar function $f : \mathbb{R} \to \mathbb{R}$

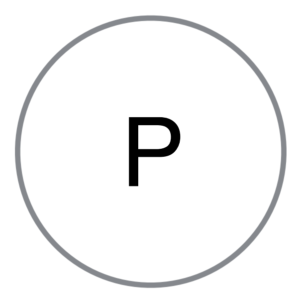
Decision Problem:

Does there exist an $x_* \in \mathbb{R}$ s.t. $f(x_*) \leq \xi$ for some $\xi \in \mathbb{R}$

Reductions and hardness

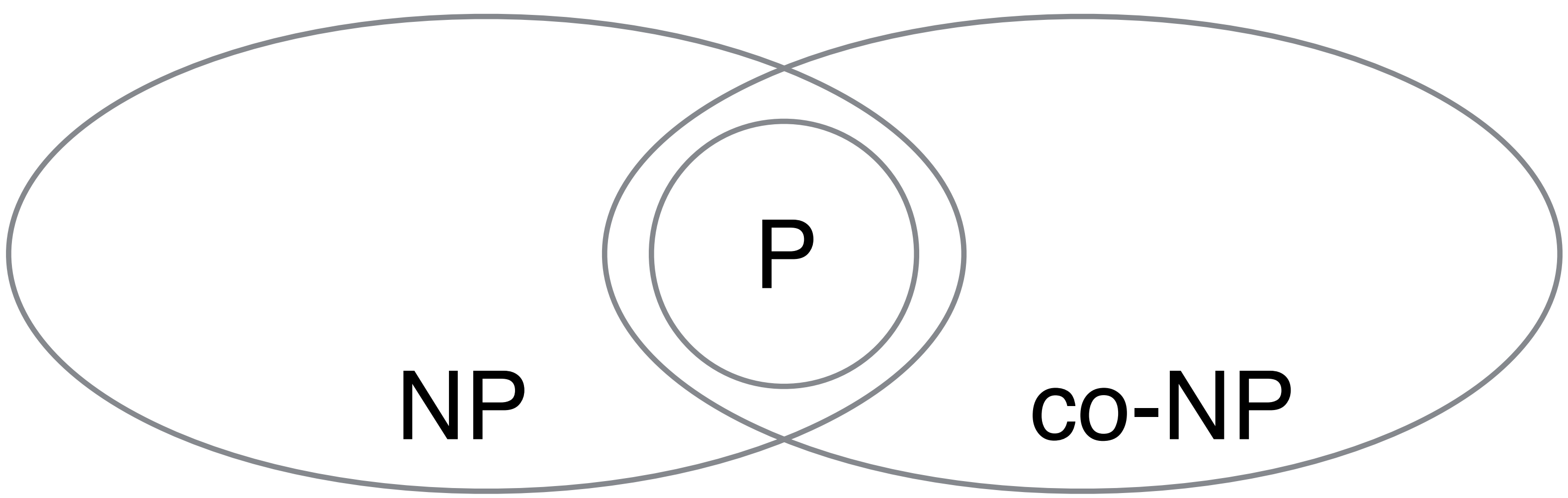
A polytime reduction from $F : I \to \{\text{yes, no}\}$ to $G : I' \to \{\text{yes, no}\}$ is a PT algorithm $I \to I'$ w/ $F(x) = \text{yes} \Leftrightarrow G(f(x)) = \text{yes}$

Reductions and hardness



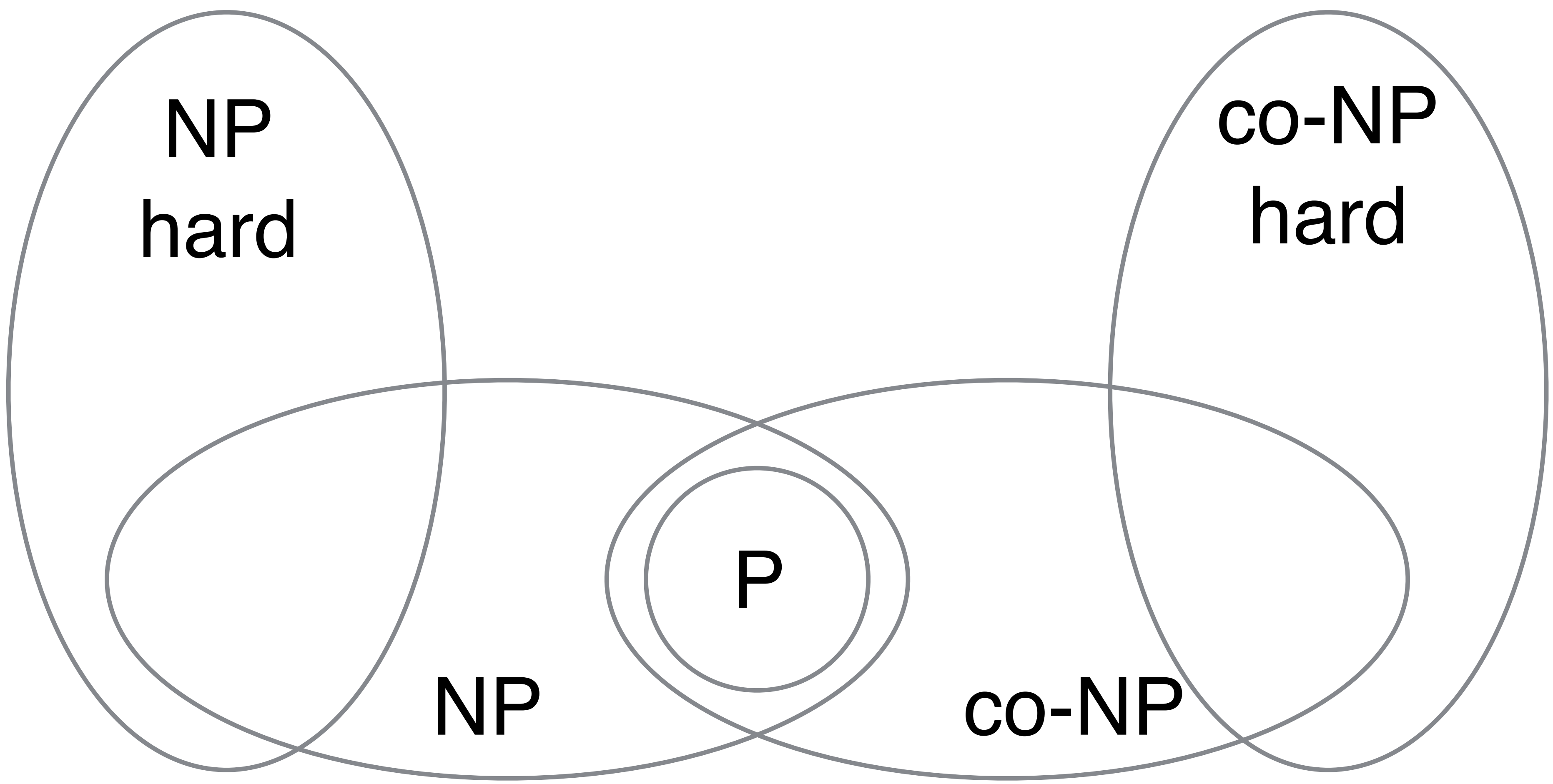
A polytime reduction from $F : I \to \{\text{yes, no}\}$ to $G : I' \to \{\text{yes, no}\}$ is a PT algorithm $I \to I'$ w/ $F(x) = \text{yes} \Leftrightarrow G(f(x)) = \text{yes}$

Reductions and hardness



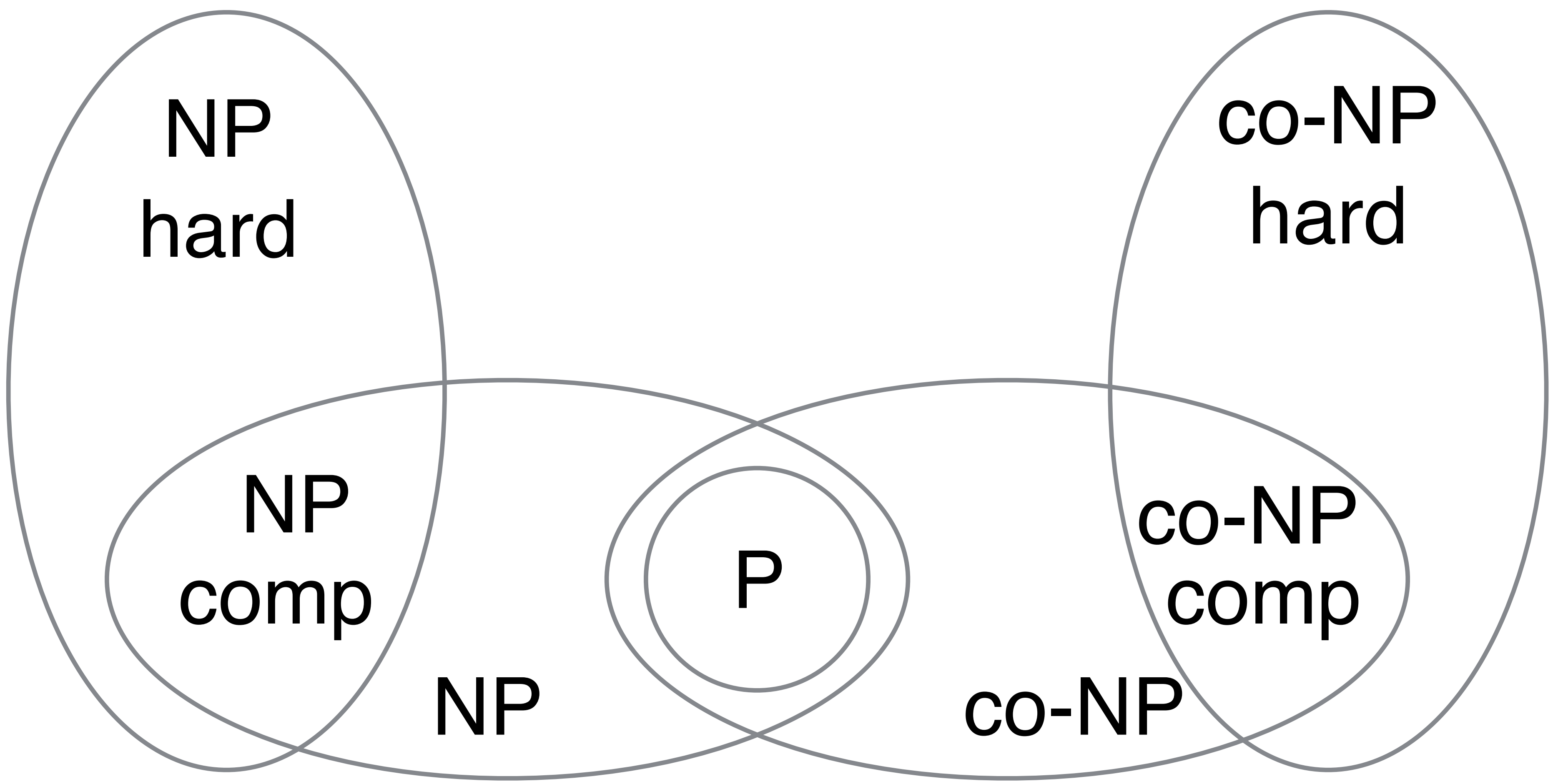
A polytime reduction from $F : I \to \{\text{yes, no}\}$ to $G : I' \to \{\text{yes, no}\}$ is a PT algorithm $I \to I'$ w/ $F(x) = \text{yes} \Leftrightarrow G(f(x)) = \text{yes}$

Reductions and hardness



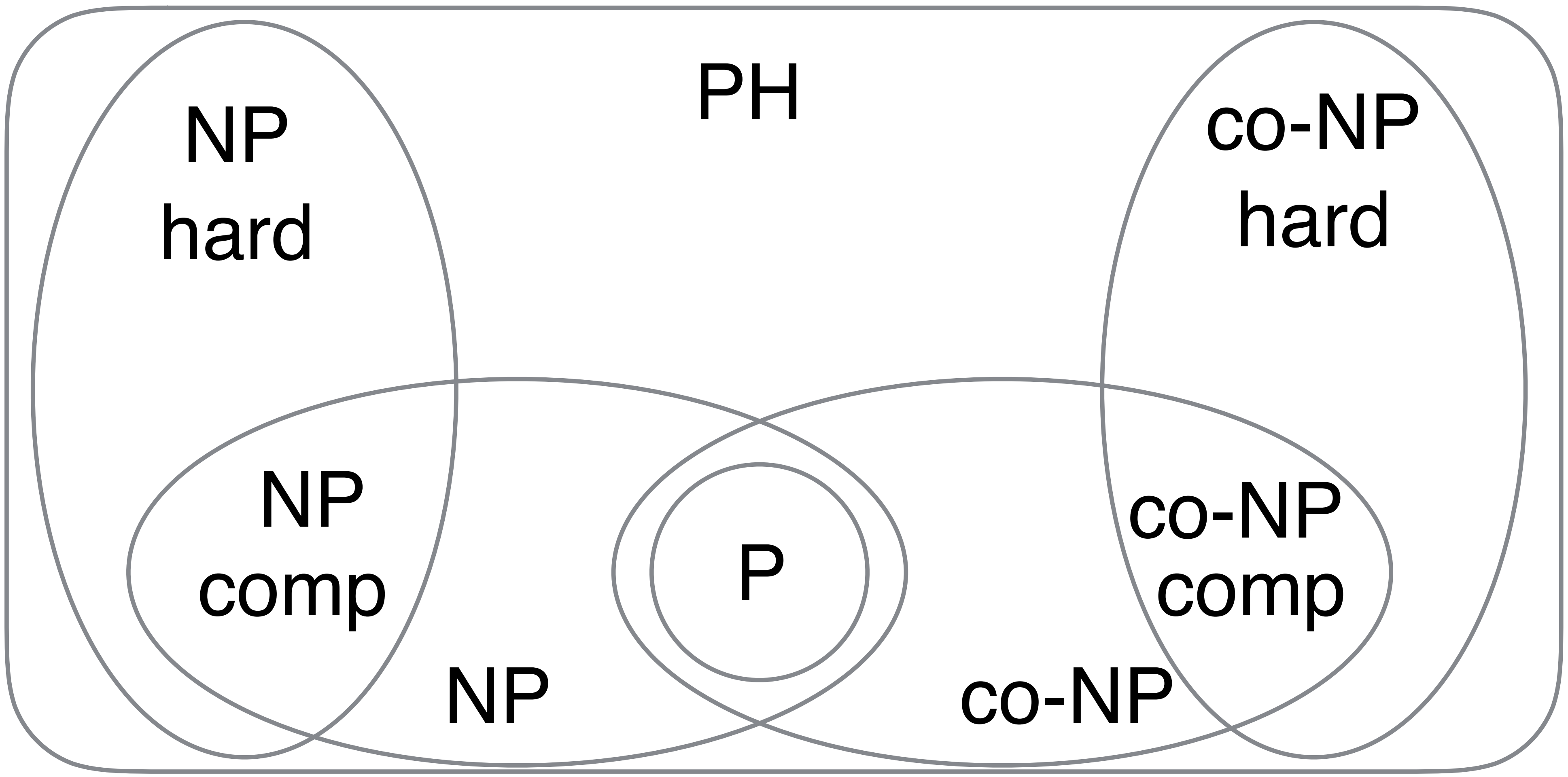
A polytime reduction from $F : I \to \{\text{yes, no}\}$ to $G : I' \to \{\text{yes, no}\}$ is a PT algorithm $I \to I'$ w/ $F(x) = \text{yes} \Leftrightarrow G(f(x)) = \text{yes}$

Reductions and hardness



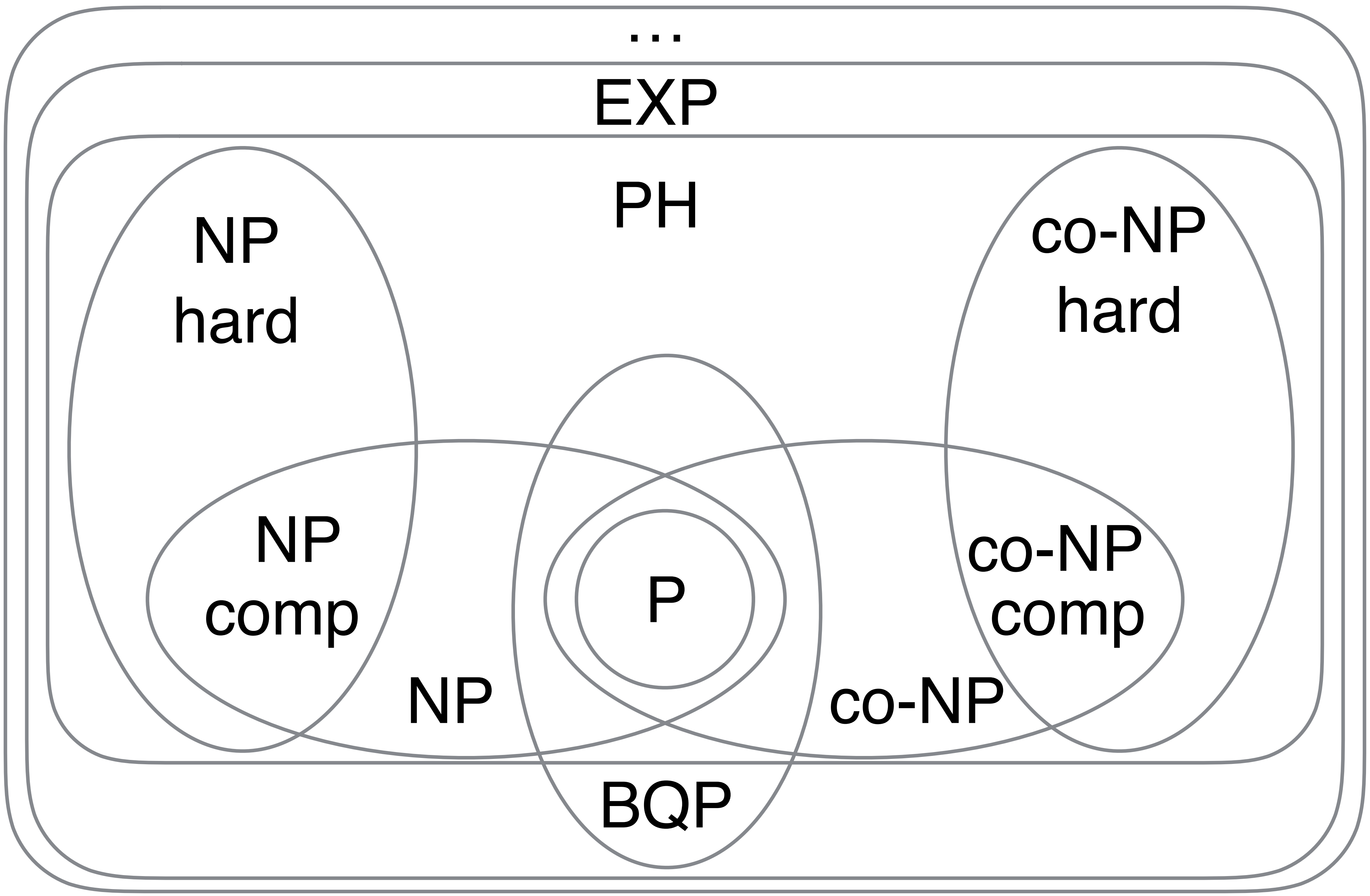
A polytime reduction from $F : I \to \{\text{yes, no}\}$ to $G : I' \to \{\text{yes, no}\}$ is a PT algorithm $I \to I'$ w/ $F(x) = \text{yes} \Leftrightarrow G(f(x)) = \text{yes}$

Reductions and hardness



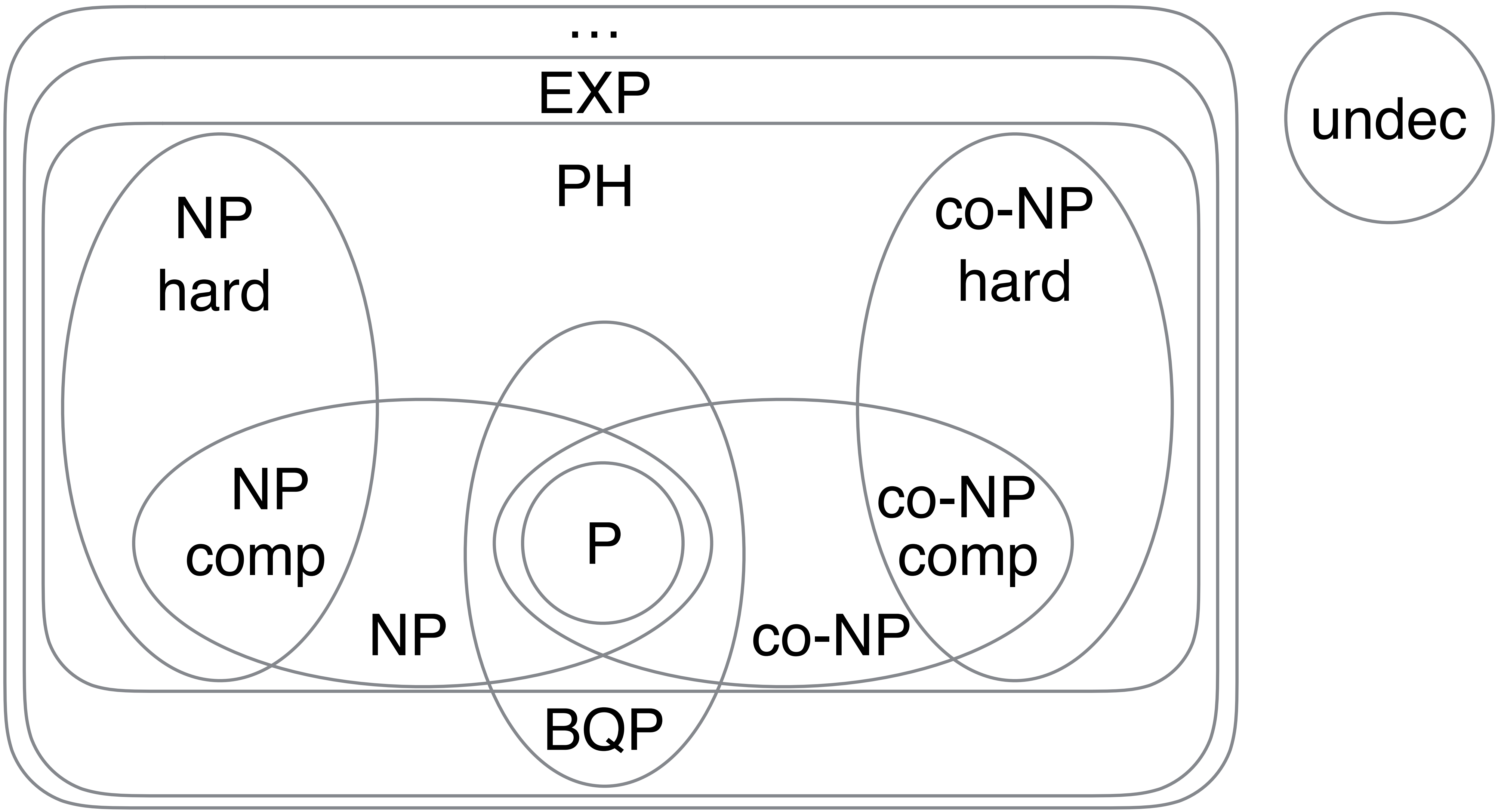
A polytime reduction from $F : I \to \{\text{yes, no}\}$ to $G : I' \to \{\text{yes, no}\}$ is a PT algorithm $I \to I'$ w/ $F(x) = \text{yes} \Leftrightarrow G(f(x)) = \text{yes}$

Reductions and hardness



A polytime reduction from $F : I \to \{\text{yes, no}\}$ to $G : I' \to \{\text{yes, no}\}$ is a PT algorithm $I \to I'$ w/ $F(x) = \text{yes} \Leftrightarrow G(f(x)) = \text{yes}$

Reductions and hardness



A polytime reduction from $F : I \to \{\text{yes, no}\}$ to $G : I' \to \{\text{yes, no}\}$ is a PT algorithm $I \to I'$ w/ $F(x) = \text{yes} \Leftrightarrow G(f(x)) = \text{yes}$

Example of NP and undecidable problems

Surprising Fact: NP hard, NP complete, and undecidable problems exist

Example of NP and undecidable problems

Surprising Fact: NP hard, NP complete, and undecidable problems exist

Subset sum (NP-complete):
Given a set of integers, does there exist a subset whose elements sum to zero? (Relevant for fine-tuning [Bousso, Polchinski `00; Arkani-Hamed,Dimopoulos,Kachru `05])

[Denef,Douglas `06]

Example of NP and undecidable problems

Surprising Fact: NP hard, NP complete, and undecidable problems exist

Subset sum (NP-complete):

Given a set of integers, does there exist a subset whose elements sum to zero? (Relevant for fine-tuning [Bousso, Polchinski `00; Arkani-Hamed,Dimopoulos,Kachru `05])

[Denef,Douglas `06]

Critical Point (NP hard):

Find the critical points of $V(\phi)$

Example of NP and undecidable problems

Surprising Fact: NP hard, NP complete, and undecidable problems exist

Subset sum (NP-complete):
Given a set of integers, does there exist a subset whose elements sum to zero? (Relevant for fine-tuning [Bousso, Polchinski `00; Arkani-Hamed,Dimopoulos,Kachru `05])

[Denef,Douglas `06]

Critical Point (NP hard):
Find the critical points of $V(\phi)$

Metastable Vacuum (co-NP hard):
Find the local minimum of $V(\phi)$, $l_i \leq \phi_i \leq u_i$

Example of NP and undecidable problems

Surprising Fact: NP hard, NP complete, and undecidable problems exist

Subset sum (NP-complete):
Given a set of integers, does there exist a subset whose elements sum to zero? (Relevant for fine-tuning [Bousso, Polchinski `00; Arkani-Hamed,Dimopoulos,Kachru `05])

[Denef,Douglas `06]

Critical Point (NP hard):
Find the critical points of $V(\phi)$

Metastable Vacuum (co-NP hard):
Find the local minimum of $V(\phi)$, $l_i \leq \phi_i \leq u_i$

Diophantine equation (undecidable):
Does a set of coupled, non-linear Diophantine eqns have a solution?

Example of NP and undecidable problems

Surprising Fact: NP hard, NP complete, and undecidable problems exist

Subset sum (NP-complete):
Given a set of integers, does there exist a subset whose elements sum to zero? (Relevant for fine-tuning [Bousso, Polchinski `00; Arkani-Hamed,Dimopoulos,Kachru `05])
[Denef,Douglas `06]

Critical Point (NP hard):
Find the critical points of $V(\phi)$

Metastable Vacuum (co-NP hard):
Find the local minimum of $V(\phi)$, $l_i \leq \phi_i \leq u_i$

Diophantine equation (undecidable):
Does a set of coupled, non-linear Diophantine eqns have a solution?

Cohomology (not NP):
Is $h^\bullet(X,V) = (h_0, h_1, h_2, h_3)$? Given h_i, we cannot check this to be true in P.

A typical workflow for constructing string models

- Choose a background geometry
- Find boundary conditions (branes, fluxes) s.t.
  - Tadpole, K-Theory, existence of unbroken SUSY somewhere $\Rightarrow$ Coupled Diophantine (undec.)
  - CC is small $\Rightarrow$ NP-complete (subset sum via BP)
- Minimize scalar potential
  - Find critical points $\Rightarrow$ NP hard
  - Check that they are minima $\Rightarrow$ co-NP hard
- Find massless spectrum
  - Compute cohomology dims $\Rightarrow$ Grobner basis (NP, double-exp)

Can ML solve NP or undec in P?

Can ML solve NP or undec in P?

- Of course not. So how does it help?

Can ML solve NP or undec in P?

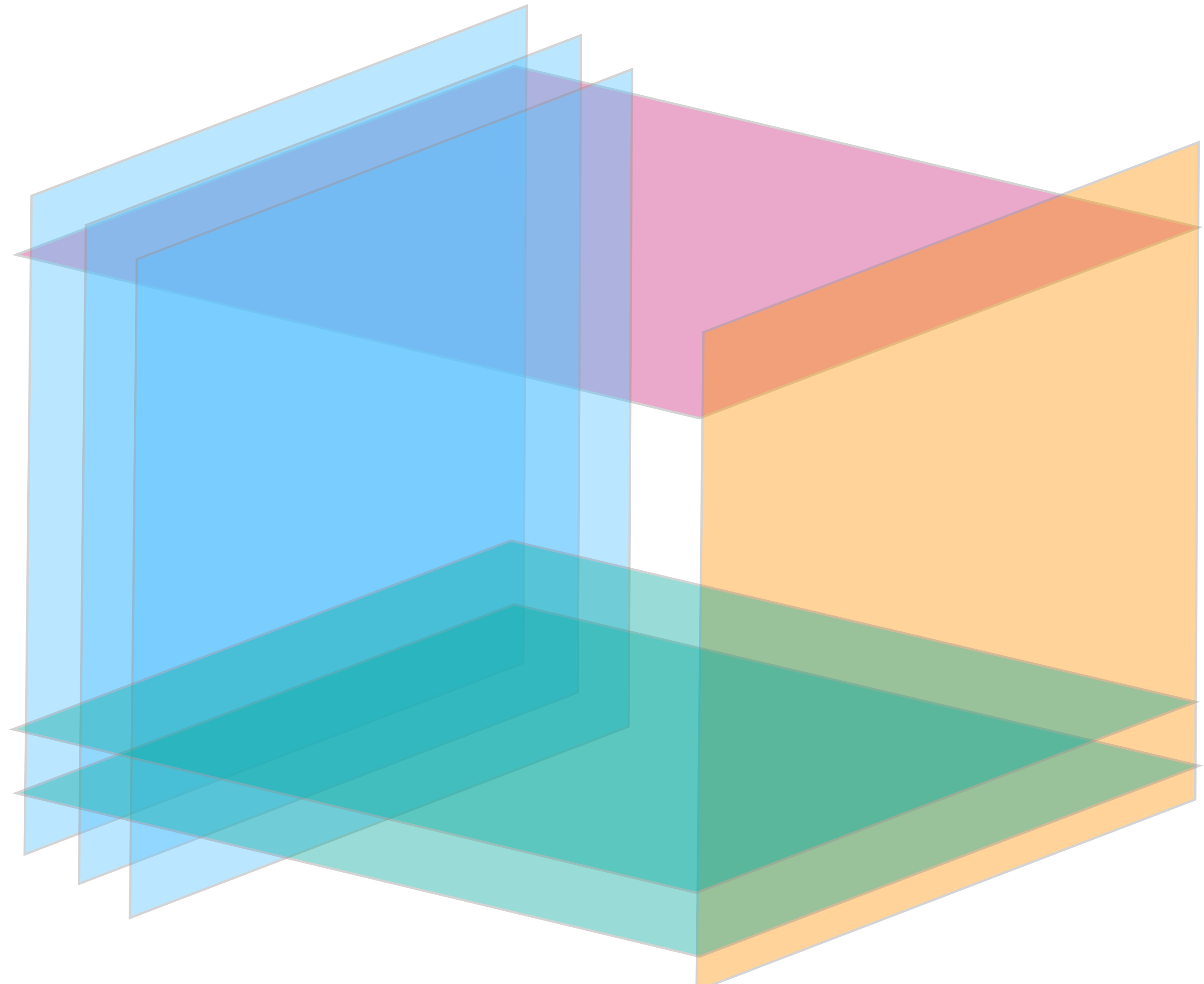
- Of course not. So how does it help?
- In practice, don’t need to solve arbitrarily large examples

Can ML solve NP or undec in P?

- Of course not. So how does it help?
- In practice, don't need to solve arbitrarily large examples
- Don't need to solve exactly (approximate a solution) and cross-check (Euler Number, anomalies, Stability)

Can ML solve NP or undec in P?

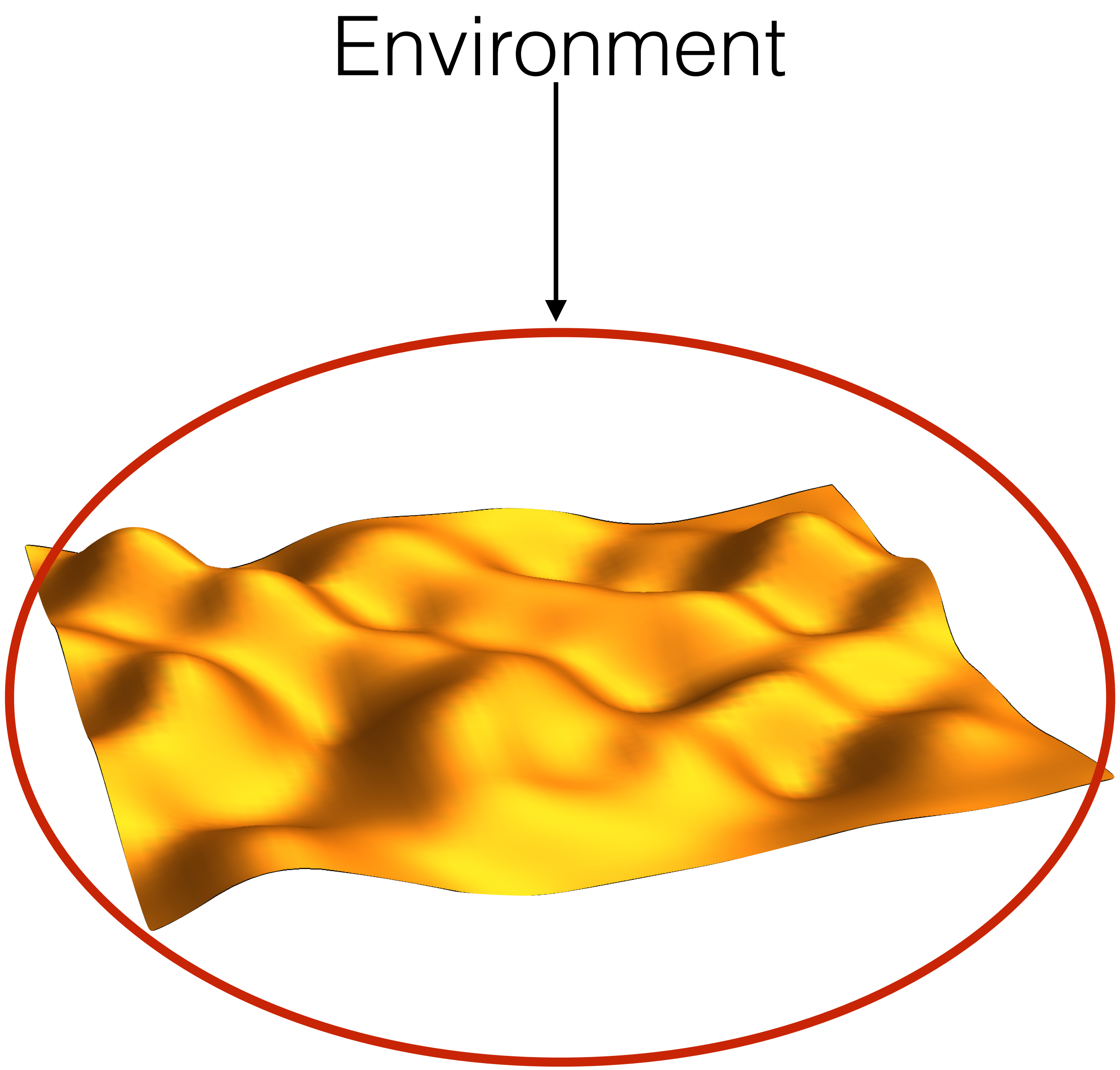
- Of course not. So how does it help?
- In practice, don't need to solve arbitrarily large examples
- Don't need to solve exactly (approximate a solution) and cross-check (Euler Number, anomalies, Stability)
- Problem might have more substructure / symmetries that simplify the computation
    - Solving general Diophantine undecidable
    - Solving quad. Diophantine like $ax_1^2 + bx_2 = c$ is NP-complete
    - Solving linear Diophantine is in P
    - Finding vacua is NP, finding near-vacua is in P



Machine learning the landscape of IIA toroidal orientifolds

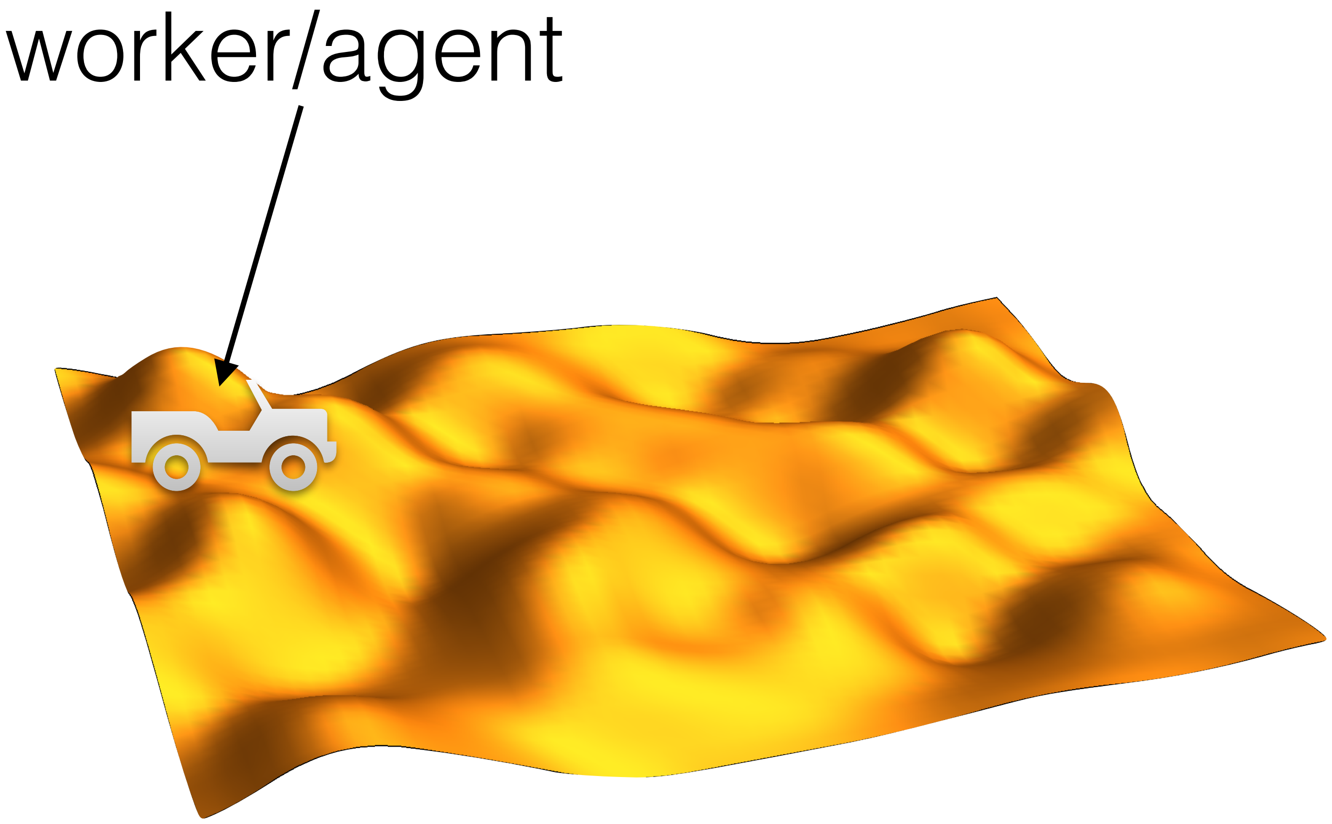
Reinforcement Learning - Basics

- Want to explore the string landscape ("environment")



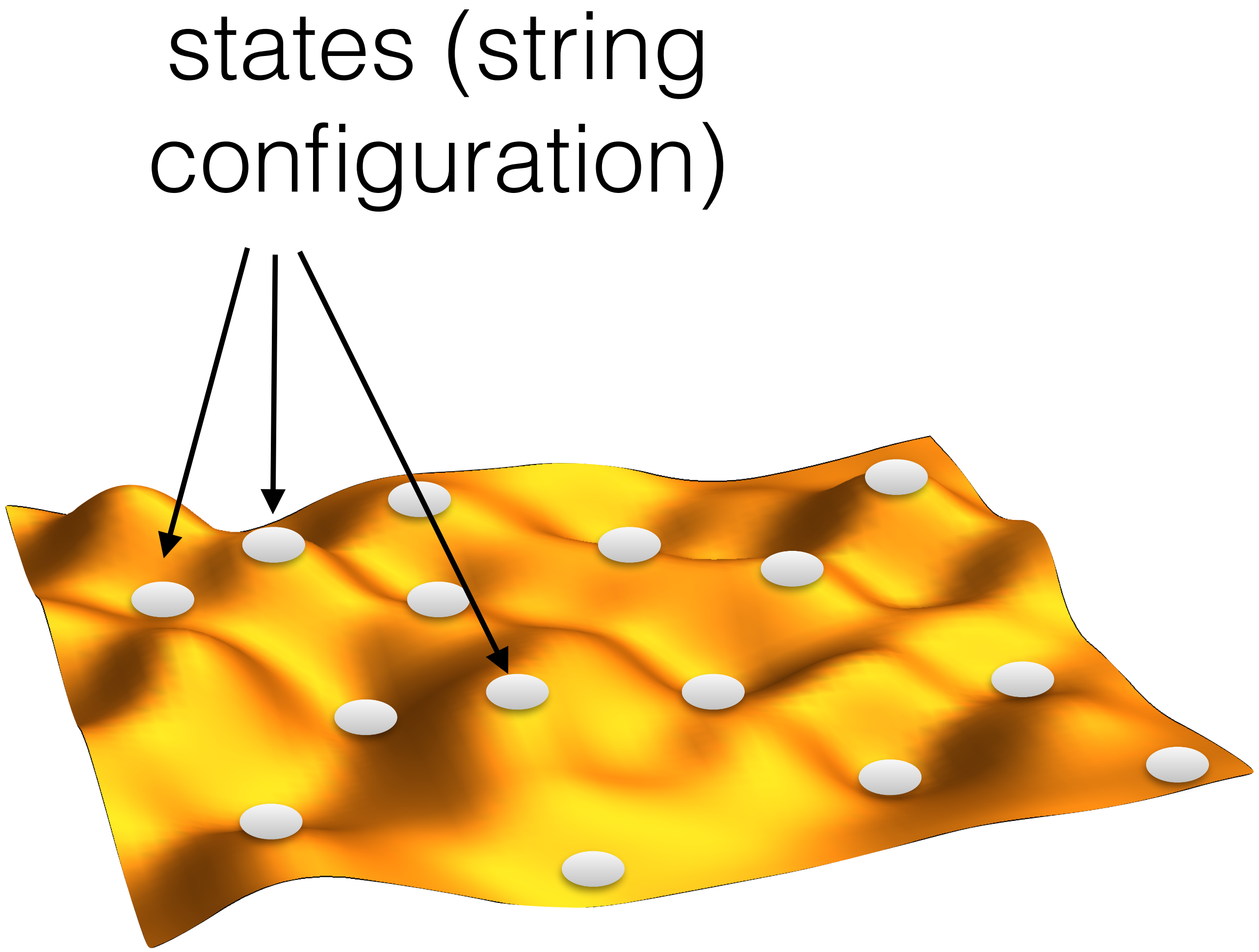
Reinforcement Learning - Basics

- Want to explore the string landscape ("environment")
- Done by "workers" that are conditioned



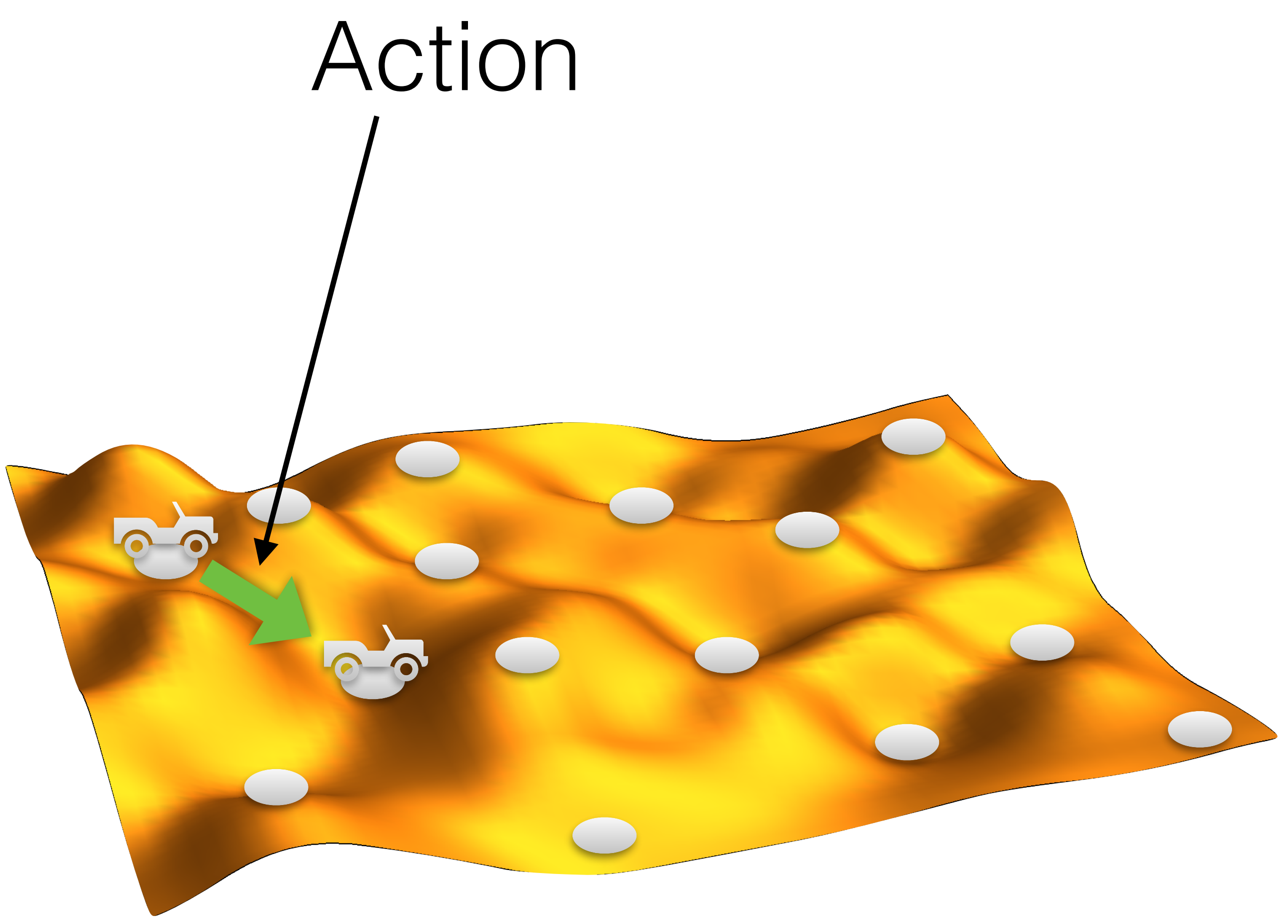
Reinforcement Learning - Basics

- Want to explore the string landscape ("environment")
- Done by "workers" that are conditioned
- At any given moment, a worker is in a specific string configuration ("state") defined by discrete topological data (branes, flux, cycles, ...)



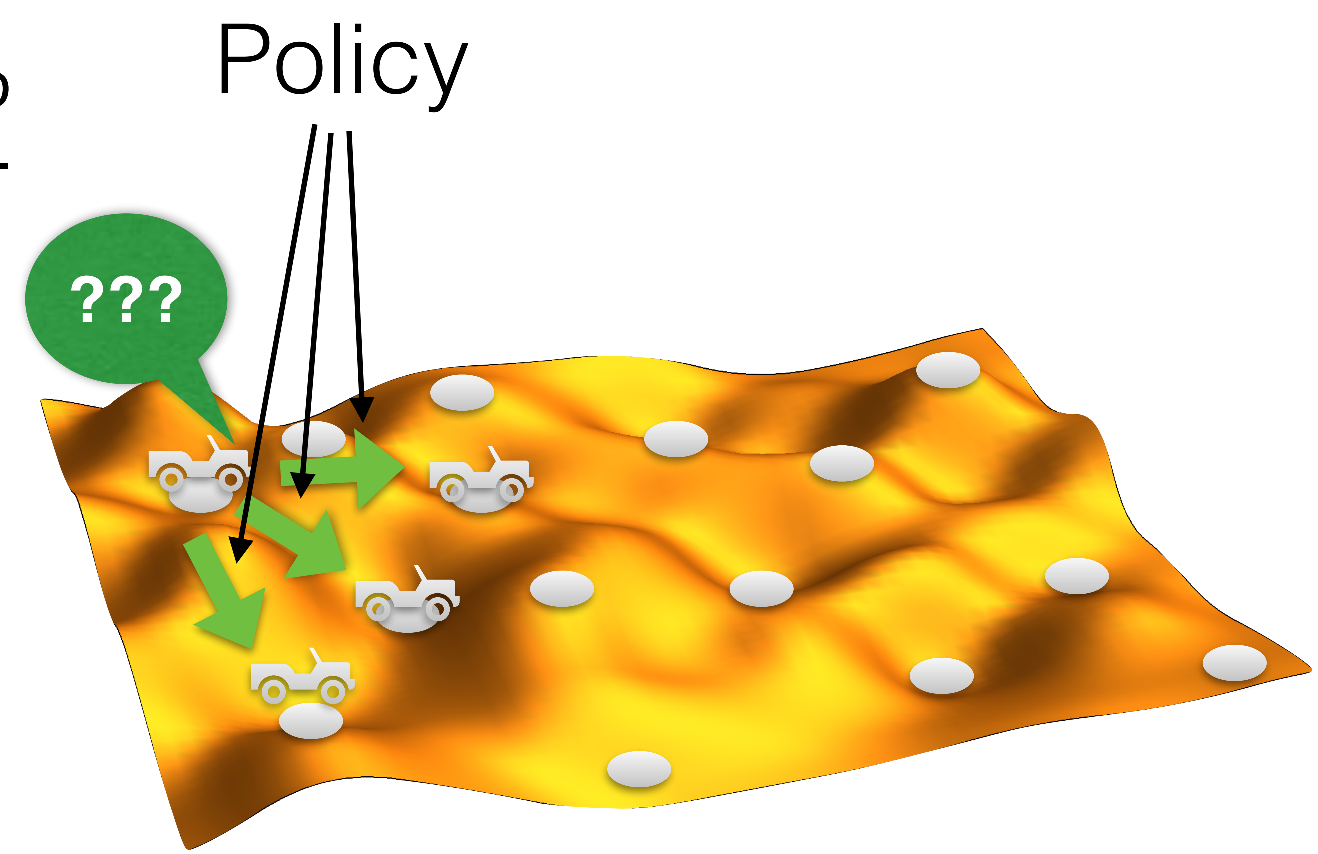
Reinforcement Learning - Basics

- Want to explore the string landscape ("environment")
- Done by "workers" that are conditioned
- At any given moment, a worker is in a specific string configuration ("state") defined by discrete topological data (branes, flux, cycles, …)
- Workers change state by taking "actions" to reach new states ("elements of the environment")



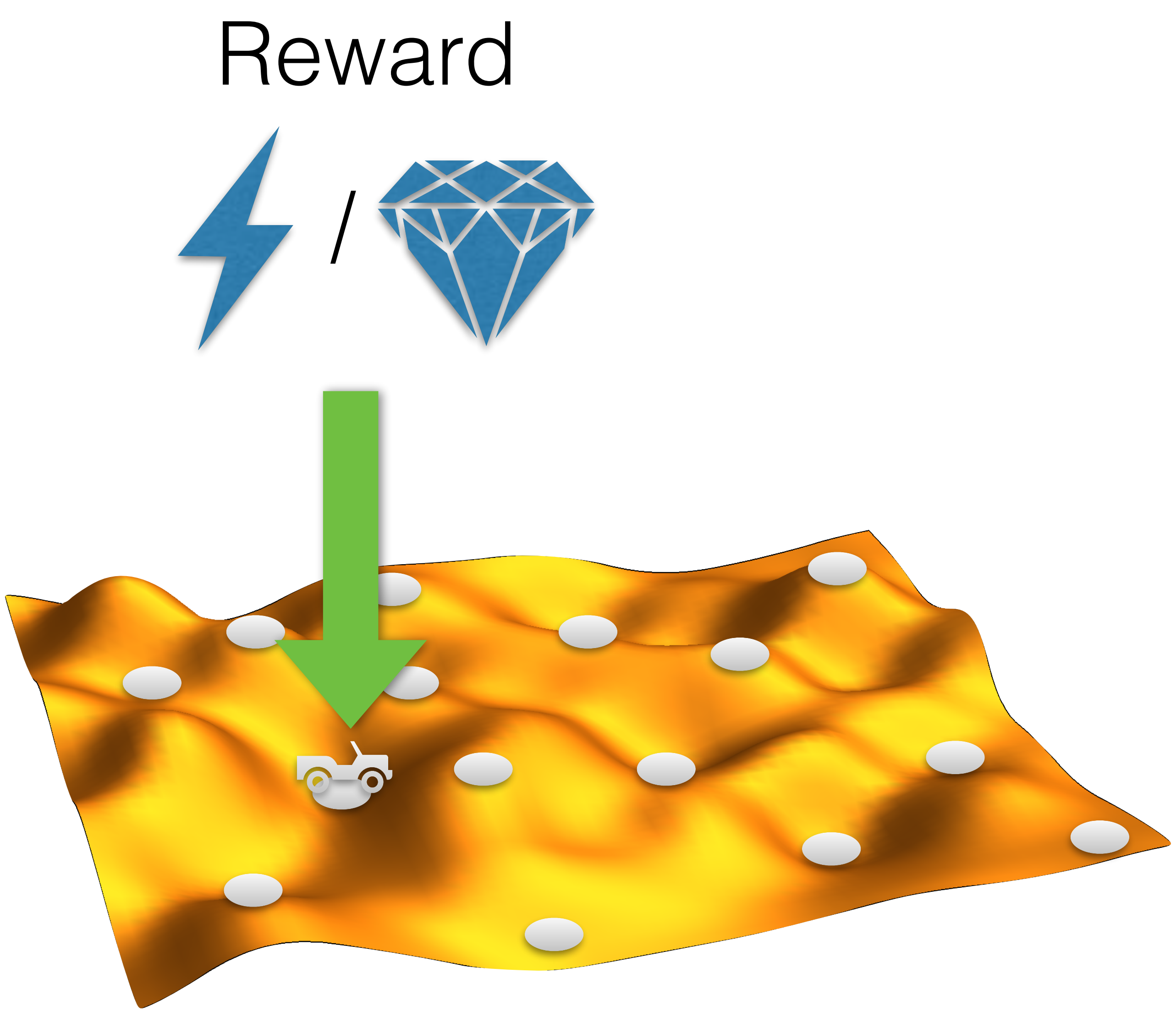
Reinforcement Learning - Basics

- Want to explore the string landscape ("environment")
- Done by "workers" that are conditioned
- At any given moment, a worker is in a specific string configuration ("state") defined by discrete topological data (branes, flux, cycles, …)
- Workers change state by taking "actions" to reach new states ("elements of the environment")
- They select these actions via some "policy"



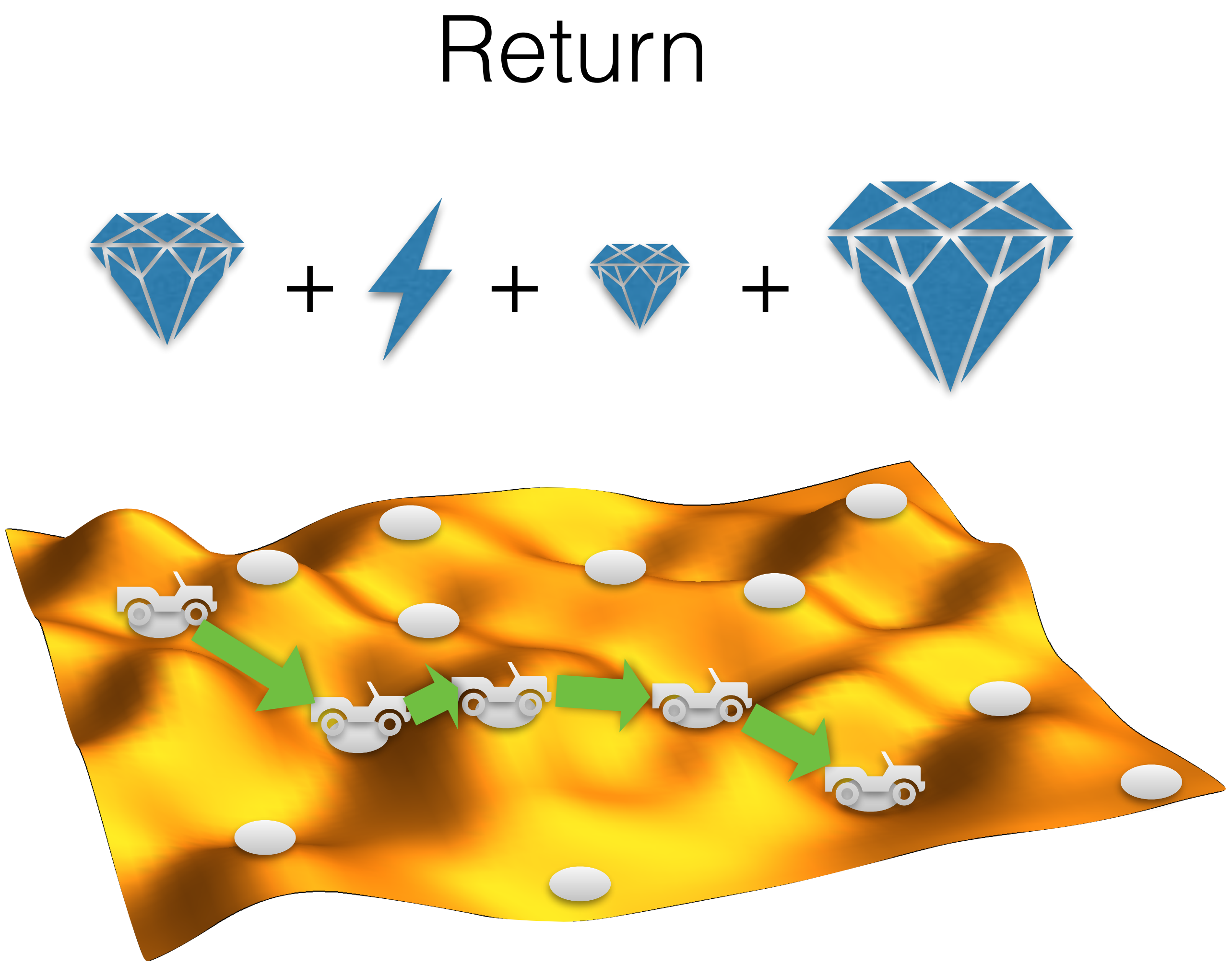
Reinforcement Learning - Basics

- Want to explore the string landscape ("environment")
- Done by "workers" that are conditioned
- At any given moment, a worker is in a specific string configuration ("state") defined by discrete topological data (branes, flux, cycles, …)
- Workers change state by taking "actions" to reach new states ("elements of the environment")
- They select these actions via some "policy"
- Depending on the chosen action they receive a pos/neg "reward"

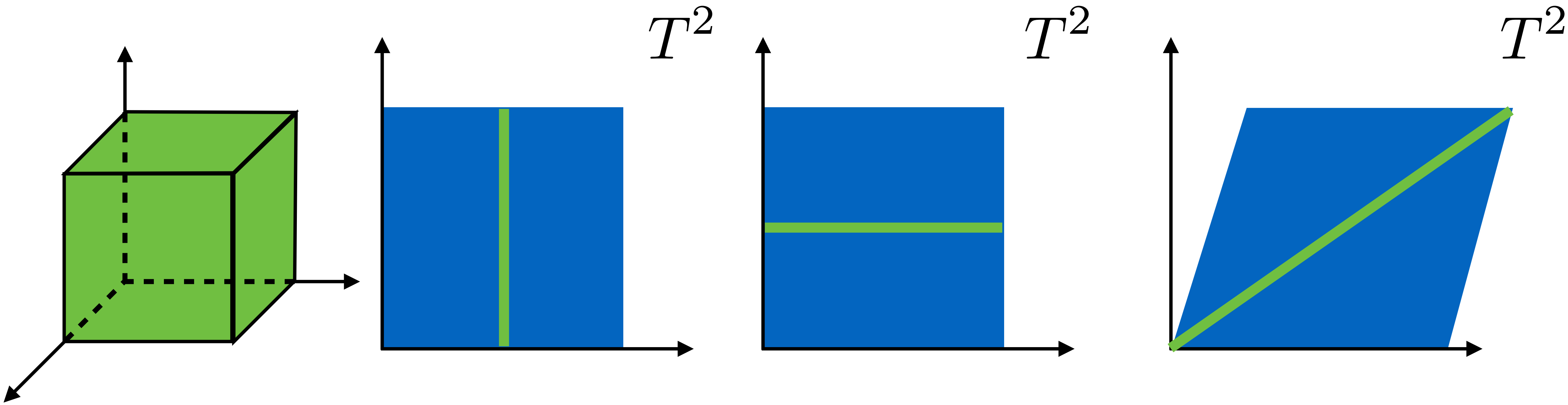


Reinforcement Learning - Basics

- Want to explore the string landscape ("environment")
- Done by "workers" that are conditioned
- At any given moment, a worker is in a specific string configuration ("state") defined by discrete topological data (branes, flux, cycles, …)
- Workers change state by taking "actions" to reach new states ("elements of the environment")
- They select these actions via some "policy"
- Depending on the chosen action they receive a pos/neg "reward"
- Via this reinforcement, the agent learns a policy that, given a state, selects an action that maximises its "return" (accumulated long-term reward)



D6 branes



- Can (have to for three generations) tilt torus (2 different complex structure choices compatible with orientifold)
- D6 brane: 4D Minkowski + a line on each torus
- Can stack multiple D6 branes on top of each other
- Brane stacks $\Leftrightarrow$ Tuple: $(N, n_1, m_1, n_2, m_2, n_3, m_3)$

D6 Branes - Consistency Conditions

- Tadpole cancellation: Balance D6 / O6 charges:

$$\sum_{a=1}^{\#\text{stacks}} \begin{pmatrix} N^a n_1^a\, n_2^a\, n_3^a \\ -N^a n_1^a\, m_2^a m_3^a \\ -N^a m_1^a\, n_2^a\, m_3^a \\ -N^a m_1^a m_2^a\, n_3^a \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 4 \\ 8 \end{pmatrix}$$

- K-Theory: Global consistency constraint:

$$\sum_{a=1}^{\#\text{stacks}} \begin{pmatrix} 2N^a m_1^a m_2^a m_3^a \\ -N^a m_1^a\, n_2^a\, n_3^a \\ -N^a n_1^a\, m_2^a\, n_3^a \\ -2N^a n_1^a\, n_2^a\, m_3^a \end{pmatrix} \mod \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

D6 Branes - Consistency Conditions

- SUSY: $\forall a = 1, \ldots, \#\ \text{stacks}$

$$m_1^a m_2^a m_3^a - j\, m_1^a n_2^a n_3^a - k\, n_1^a m_2^a n_3^a - \ell n_1^a n_2^a m_3^a = 0$$

$$n_1^a n_2^a n_3^a - j\, n_1^a m_2^a m_3^a - k\, m_1^a n_2^a m_3^a - \ell m_1^a m_2^a n_3^a > 0$$

- Pheno: $SU(3) \times SU(2) \times U(1)$ + MSSM particles

- Massless $U(1)$'s: $T_r \in \ker(\{N^k m_i^k\})$

$$i = 1, 2, 3 \quad \text{(three tori)}$$

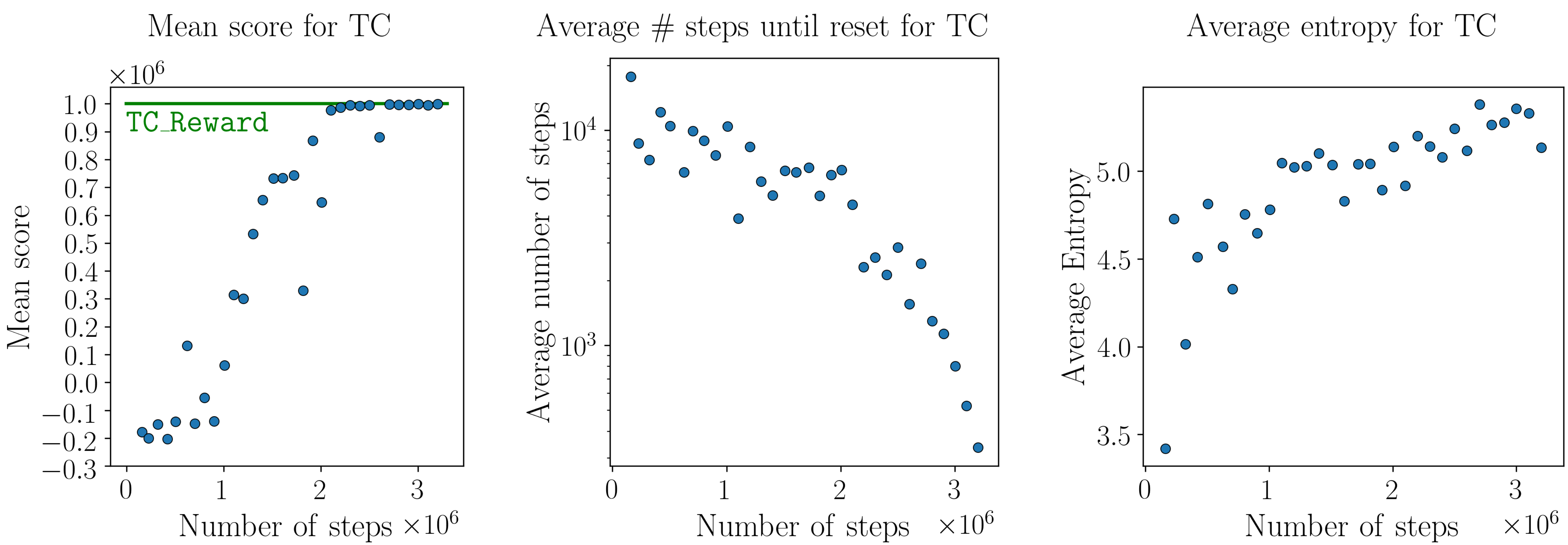
$$k = 1, \ldots, \#U \text{ brane stacks}$$

$$r = 1, \ldots, \dim(\ker(\{N^k m_i^k\}))$$

$$= k - 3 \text{ (generically)}$$

Learn TC condition

$$\sum_{a=1}^{\#\text{stacks}} \begin{pmatrix} N^a n_1^a\, n_2^a\, n_3^a \\ -N^a n_1^a\, m_2^a m_3^a \\ -N^a m_1^a\, n_2^a\, m_3^a \\ -N^a m_1^a m_2^a\, n_3^a \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 4 \\ 8 \end{pmatrix}$$



Learn TC+K+SUSY condition

Tadpole cancellation:

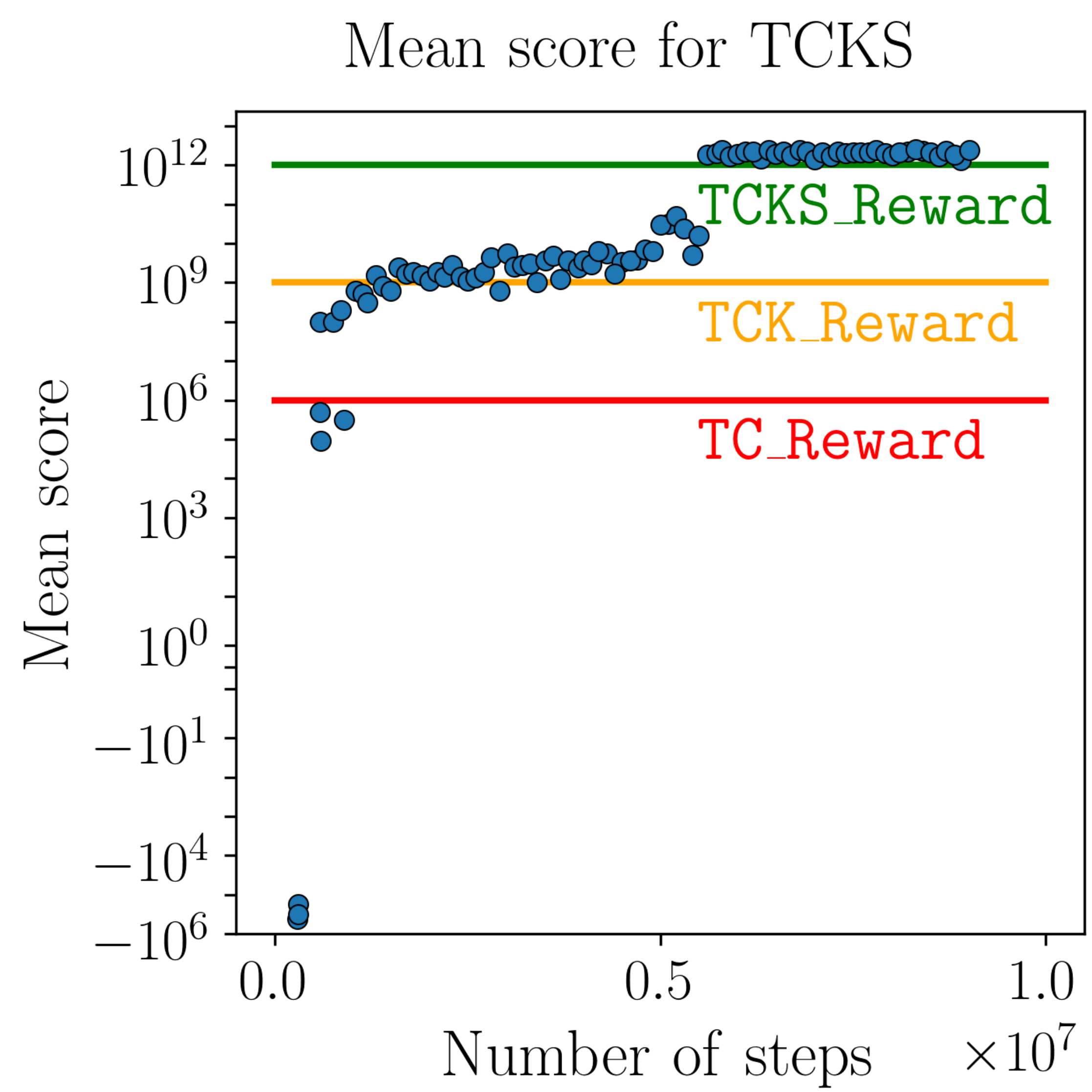
$$\sum_{a=1}^{\#\text{stacks}} \begin{pmatrix} N^a\, n_1^a\; n_2^a\; n_3^a \\ -N^a\, n_1^a\, m_2^a m_3^a \\ -N^a m_1^a\, n_2^a\, m_3^a \\ -N^a m_1^a m_2^a\, n_3^a \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 4 \\ 8 \end{pmatrix}$$

K-Theory:

$$\sum_{a=1}^{\#\text{stacks}} \begin{pmatrix} 2N^a m_1^a m_2^a m_3^a \\ -N^a m_1^a\, n_2^a\; n_3^a \\ -N^a\, n_1^a\, m_2^a\, n_3^a \\ -2N^a\, n_1^a\; n_2^a\, m_3^a \end{pmatrix} \bmod \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

SUSY:

$$m_1^a m_2^a m_3^a - j\, m_1^a n_2^a n_3^a - k\, n_1^a m_2^a n_3^a - \ell n_1^a n_2^a m_3^a = 0$$
$$n_1^a n_2^a n_3^a - j\, n_1^a m_2^a m_3^a - k\, m_1^a n_2^a m_3^a - \ell m_1^a m_2^a n_3^a > 0$$



Learn SUSY+TC condition

K-Theory:

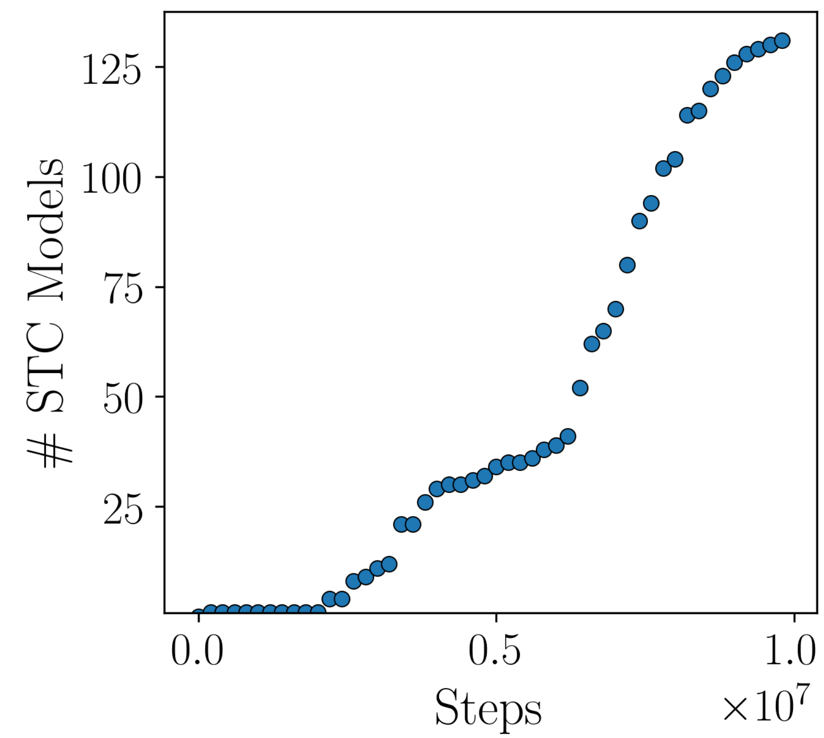
$$\sum_{a=1}^{\#\text{stacks}} \begin{pmatrix} 2N^a m_1^a m_2^a m_3^a \\ -N^a m_1^a\, n_2^a\, n_3^a \\ -N^a\, n_1^a\, m_2^a\, n_3^a \\ -2N^a\, n_1^a\, n_2^a\, m_3^a \end{pmatrix} \bmod \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

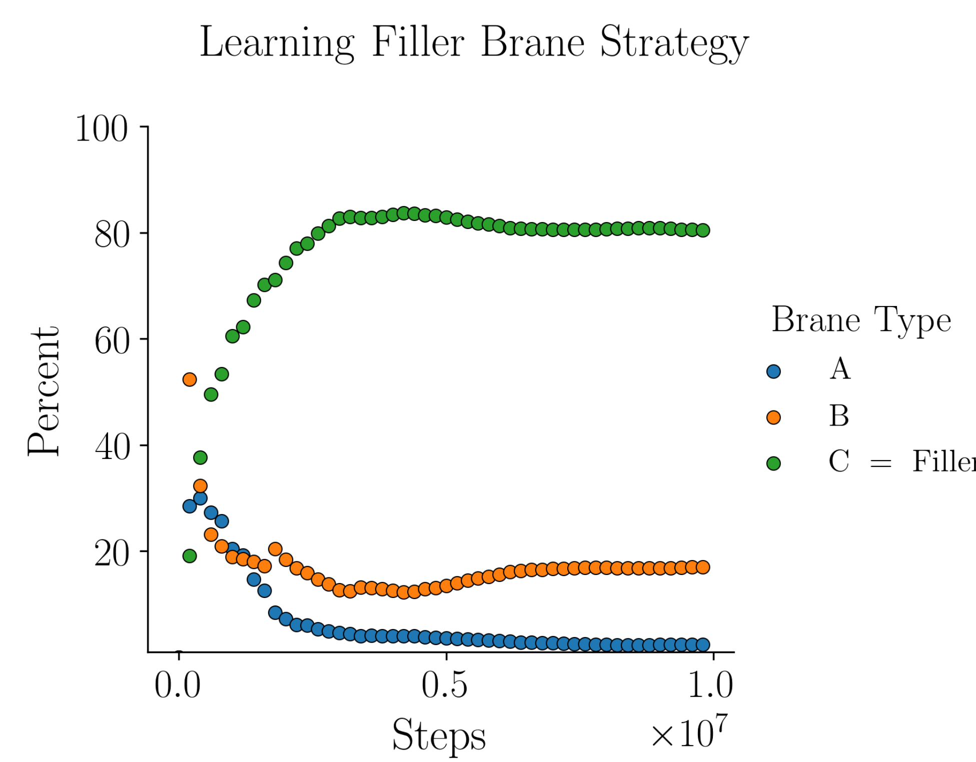
SUSY:

$$m_1^a m_2^a m_3^a - j\, m_1^a n_2^a n_3^a - k\, n_1^a m_2^a n_3^a - \ell n_1^a n_2^a m_3^a = 0$$
$$n_1^a n_2^a n_3^a - j\, n_1^a m_2^a m_3^a - k\, m_1^a n_2^a m_3^a - \ell m_1^a m_2^a n_3^a > 0$$

Tadpole cancellation:

$$\sum_{a=1}^{\#\text{stacks}} \begin{pmatrix} N^a\, n_1^a\, n_2^a\, n_3^a \\ -N^a\, n_1^a\, m_2^a m_3^a \\ -N^a m_1^a\, n_2^a\, m_3^a \\ -N^a m_1^a m_2^a\, n_3^a \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 4 \\ 8 \end{pmatrix}$$





Conclusions

- Finding viable vacua requires solving nested hard and undecidable problems
- By finding structures and/or making approximations you can tackle these problems
- For toroidal orientifold example we found
  - ML (RL) finds strategies to solve string consistency constraints
  - ML recovers human-derived strategies and finds new ones

Advertisements

Data science applications to string theory
~10/2019

Machine learning and its applications
October 7-11, 2019, BCTP Bonn

bctp

Bethe Forum on ML in Physics
May 2020, Bonn University

SIMONS CENTER FOR GEOMETRY AND PHYSICS

Neural networks and the Data Science Revolution: from theoretical physics to neuroscience, and back
January 6-31, 2020, SCGP Stony Brook

Thank you