Machine learning in string theory

Fabian Ruehle (University of Oxford)

Geometry and Strings

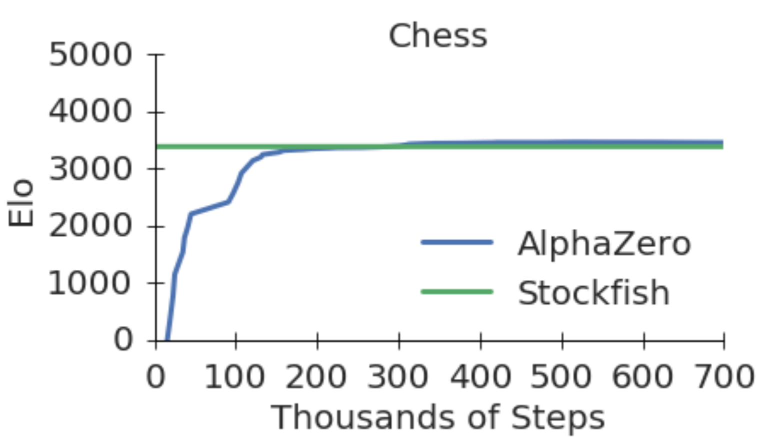
23/07/2018

Based on:

[Brent Nelson, Jim Halverson, Fabian Ruehle]

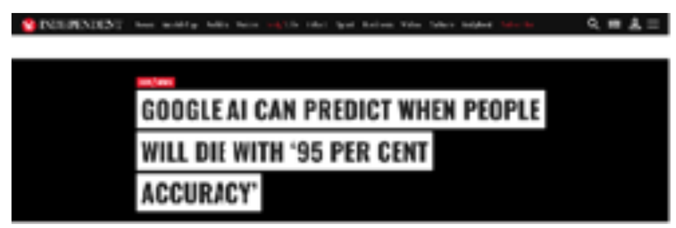
[Brent Nelson, Jim Halverson, Cody Long, Fabian Ruehle]

Motivation - ML in Science and Society



[Silver et al. `17]

Motivation - ML in Science and Society



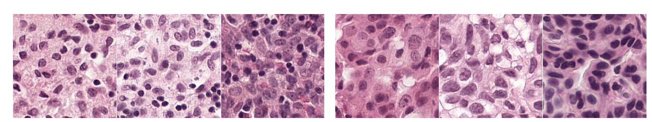


Fig. 1. Left: three tumor patches and **right**: three challenging normal patches.

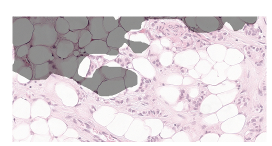


Fig. 2. Difficulty of pixel-accurate annotations for scattered tumor cells. Ground truth annotation is overlaid with a lighter shade. Note that the tumor annotations include both tumor cells and normal cells *e.g.*,white space representing adipose tissue (fat).

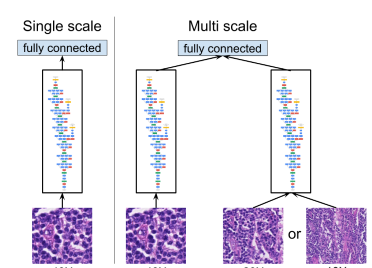
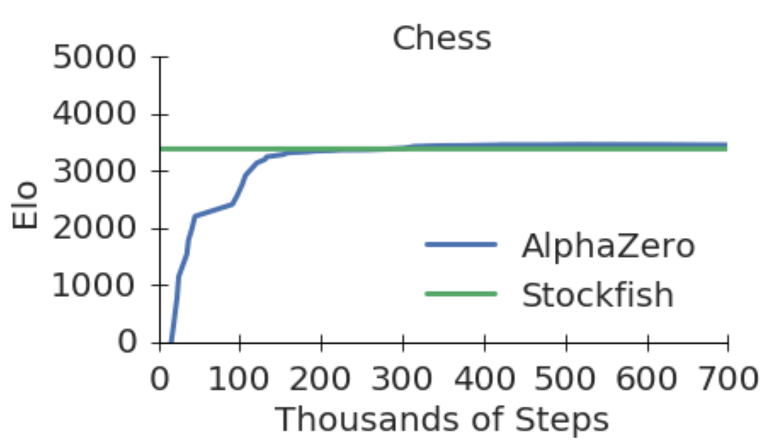


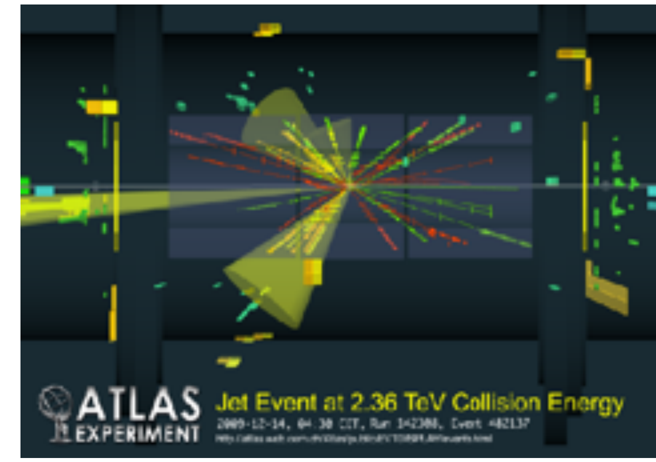
Fig. 3. The three colorful blocks represent Inception (V3) towers up to the second-last layer (PreLogit). *Single scale* utilizes one tower with input images at 40X magnification; *multi-scale* utilizes multiple (*e.g.*,2) input magnifications that are input to separate towers and merged.

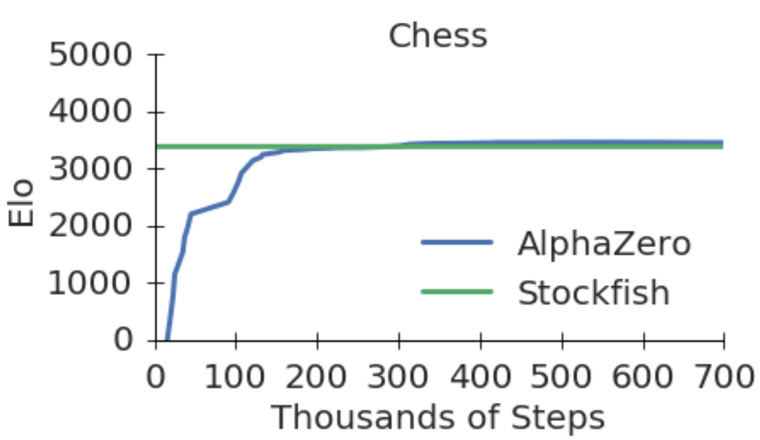
[Liu et al.`17]



[Silver et al.`17]

Motivation - ML in Science and Society





[Silver et al.`17]

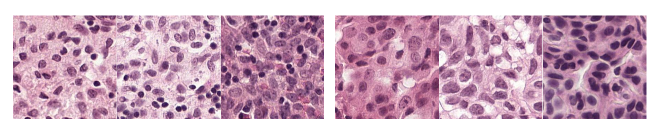


Fig. 1. Left: three tumor patches and **right**: three challenging normal patches.

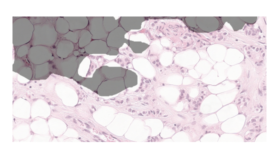


Fig. 2. Difficulty of pixel-accurate annotations for scattered tumor cells. Ground truth annotation is overlaid with a lighter shade. Note that the tumor annotations include both tumor cells and normal cells *e.g.*,white space representing adipose tissue (fat).

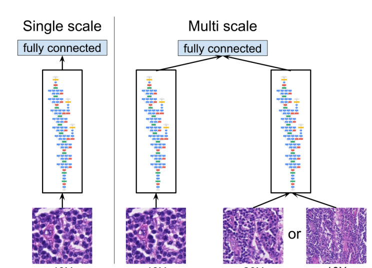
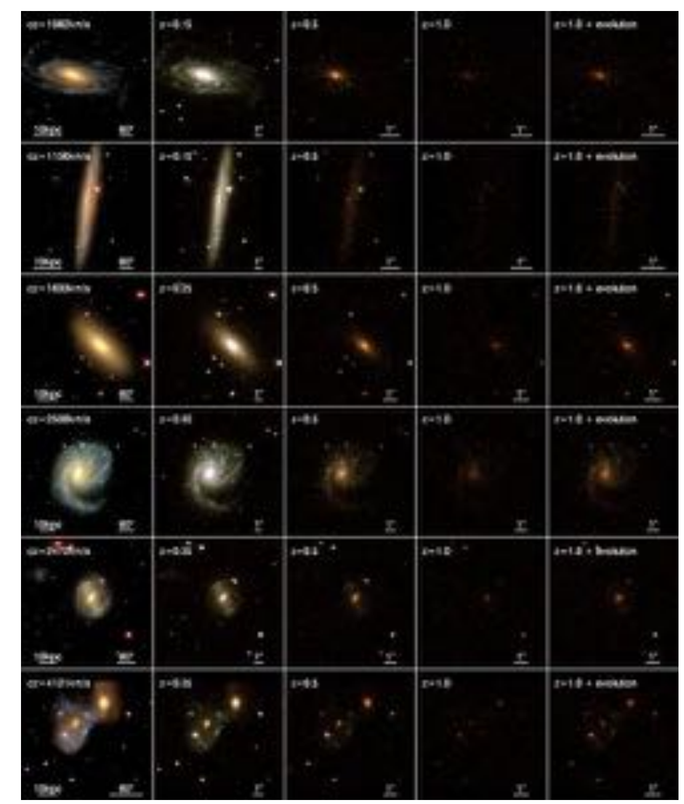


Fig. 3. The three colorful blocks represent Inception (V3) towers up to the second-last layer (PreLogit). *Single scale* utilizes one tower with input images at 40X magnification; *multi-scale* utilizes multiple (*e.g.*,2) input magnifications that are input to separate towers and merged.

[Liu et al.`17]



[Zooniverse`18; picture from Barden et al`08]

Motivation - ML in String Theory

- Possible applications of ML in string theory
  - Find string models in the landscape
  - Find generic / common features of string-derived model and extract string theory predictions from the landscape
  - Find patterns in mathematics of string theory
  - Use machine learning / AI to perform computation intensive work [FR'17]
  - …
- Can we use machine learning to study the landscape? [He'17; Krefl, Seong'17; FR'17; Carifio, Halverson, Krioukov, Nelson'17]

Motivation - ML in String Theory

- Possible applications of ML in string theory
  - Find string models in the landscape
    [Halverson, Nelson, FR] [Carifio, Cunningham, Halverson, Krioukov, Long, Nelson][Halverson, Nilles, FR, Vaudrevange] [Faraggi et al] [Sven's talk]
  - Find generic / common features of string-derived model and extract string theory predictions from the landscape
    [He] [Vaudrevange et al] [Bull, He, Jejjala, Mishra]
  - Find patterns in mathematics of string theory
    [Krefl, Song] [Cole, Shiu] [Taylor, Wang] [Halverson, Long, FR, Tian] [Halverson, Long, FR, Nelson] [Liu] [You, Yang, Qi] [Hashimoto, Sugishita, Tanaka, Tomiya] [Jinno]
  - Use machine learning / AI to perform computation intensive work [FR'17] [Sven's talk]
  - …
- Can we use machine learning to study the landscape?
  [He'17; Krefl, Seong'17; FR'17; Carifio, Halverson, Krioukov, Nelson'17]

Motivation - ML in String Theory

- Possible applications of ML in string theory
  - Find string models in the landscape
    [Halverson, Nelson, FR] [Carifio, Cunningham, Halverson, Krioukov, Long, Nelson][Halverson, Nilles, FR, Vaudrevange] [Faraggi et al] [Sven's talk]
  - Find generic / common features of string-derived model and extract string theory predictions from the landscape
    [He] [Vaudrevange et al] [Bull, He, Jejjala, Mishra]
  - Find patterns in mathematics of string theory
    [Krefl, Song] [Cole, Shiu] [Taylor, Wang] [Halverson, Long, FR, Tian] [Halverson, Long, FR, Nelson] [Liu] [You, Yang, Qi] [Hashimoto, Sugishita, Tanaka, Tomiya] [Jinno]
  - Use machine learning / AI to perform computation intensive work [FR'17] [Sven's talk]
  - …
- Can we use machine learning to study the landscape?
  [He'17; Krefl, Seong'17; FR'17; Carifio, Halverson, Krioukov, Nelson'17]

Motivation - ML in String Theory

4D string theories highly non-unique

- Different choices lead to 10^{755} or more string vacua (Go has 10^{172} states)
  [Douglas `03; Douglas, Sen `04; Halverson, Long, Sung `17; Taylor, Wang `15-`17]
- Number huge but seems finite
  [Reid `87; Douglas, Taylor `07; Buchbinder, Constantin, Lukas `14; Groot Nibbelink, Loukas, FR, Vaudrevange `15; Di Cerbo, Svaldi `16]
- Most of these vacua do not correspond to our universe
- Problem: We know the phenomenological properties a string theory that describes our universe has to have, but we lack a vacuum selection mechanism

Motivation - ML in String Theory

When choosing a string background (geometry, flux):

- Need to ensure mathematical/physical consistency
  - Tadpole and anomaly cancellation
  - Solution is actual vacuum (D- and F-flat)
- Need to ensure physically desirable features
  - Gauge algebra of the SM: $\mathrm{SU}(3) \times \mathrm{SU}(2) \times \mathrm{U}(1)_Y$
  - Three families of quarks and leptons, one Higgs pair
  - Absence of exotics, realistic Yukawas
  - Realistic cosmological constant

Motivation - ML in String Theory

- Mathematical constraints: Often collection of non-linear, coupled Diophantic equations
- Physical constraints: Further constrains Diophantic solutions in non-obvious way
- Upshot:
  - For a given configuration we can check its viability easily, but we have no idea how to find a good configuration in the first place
- To traverse vacua: Use Reinforcement Learning, a semi-supervised approach to Machine Learning

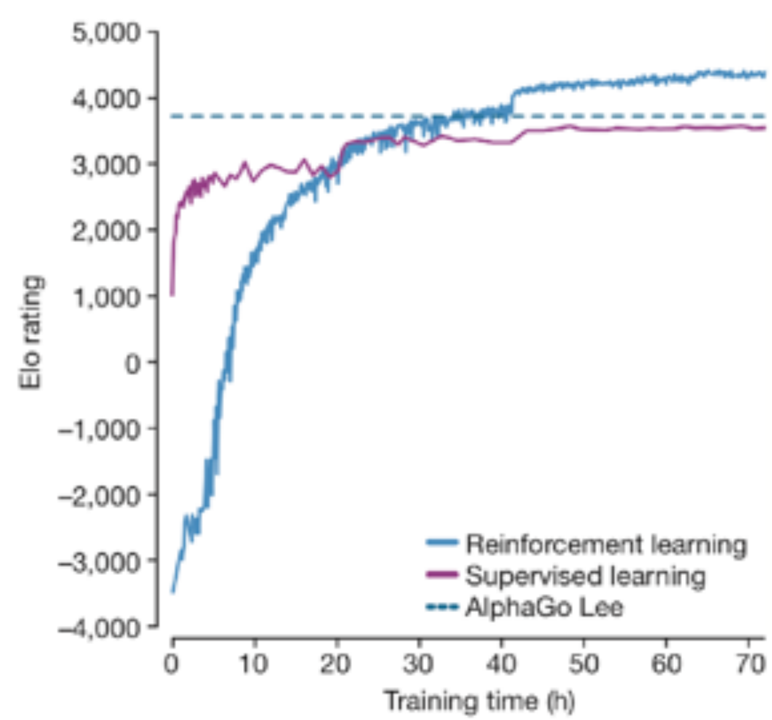
Outline

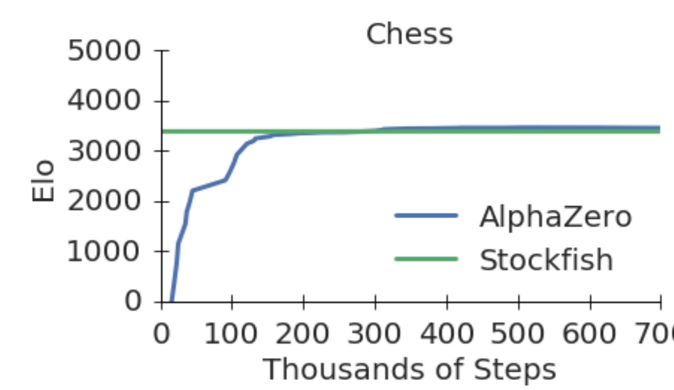
- Reinforcement Learning (RL)
  - Introduction to RL
  - Interlude: NNs + Tree searches
  - Implementation
- Example applications
  - Finding vacua in Type IIA/B intersecting brane models
  - Finding weak coupling limits in the F-Theory landscape
- Conclusion

Reinforcement learning

Reinforcement Learning - Idea

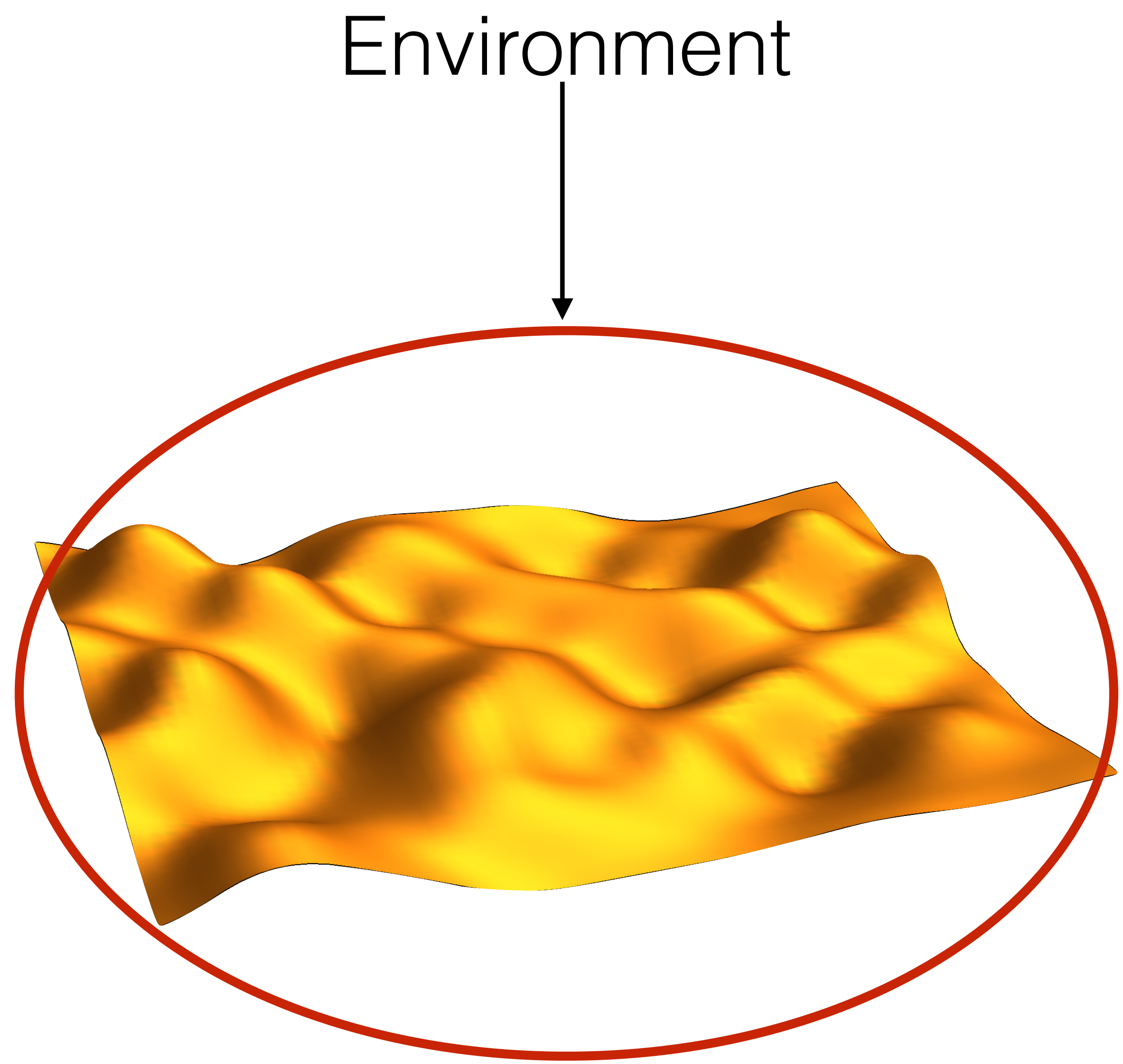
- Basic textbooks/literature [Barton, Sutton ’98 ‘17]
- Based on behavioural psychology: train individual by
  - Rewarding “good” behavior
  - Punishing “bad” behavior
- Used e.g. in Go [Silver et. al. ’16 ‘17]





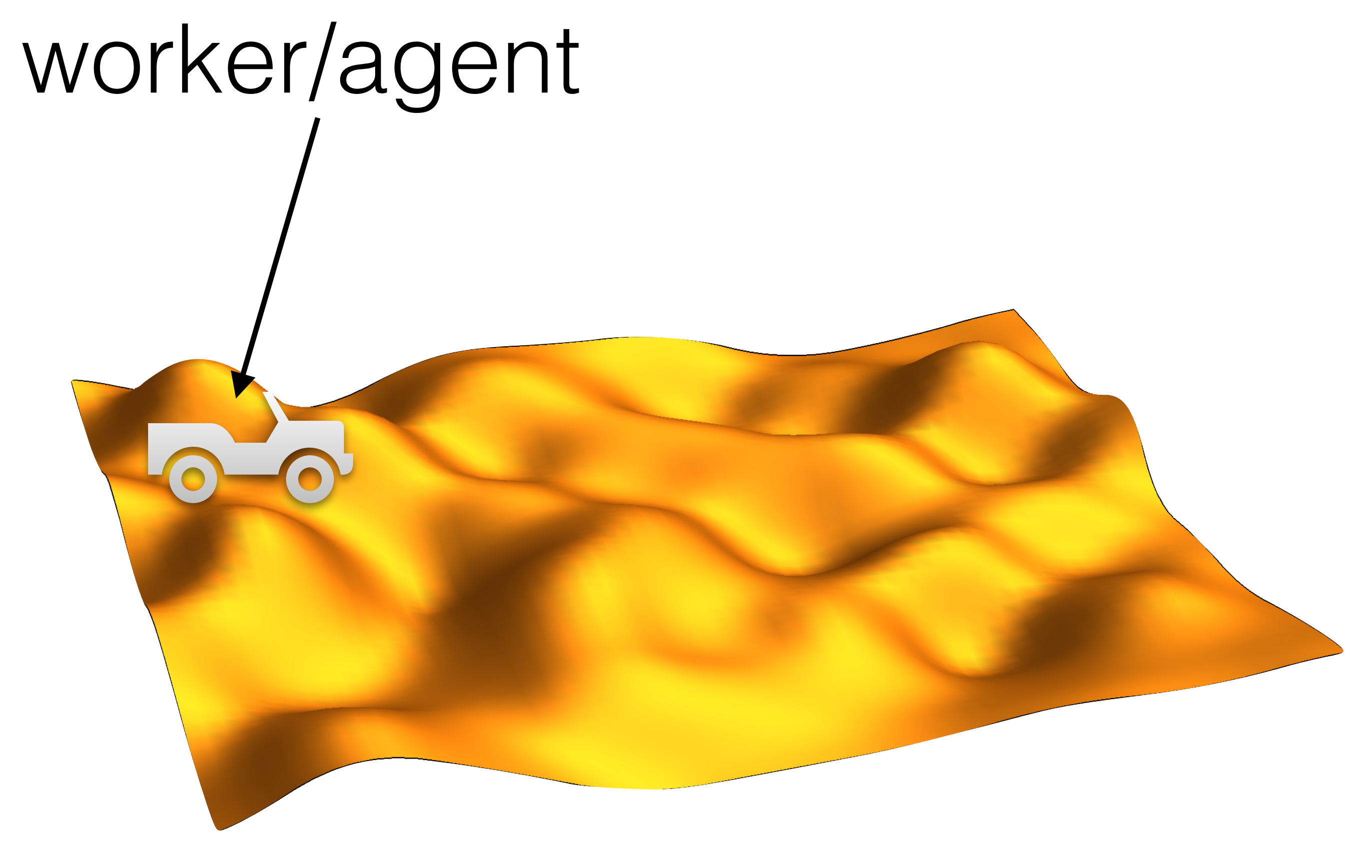
Reinforcement Learning - Vocabulary

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



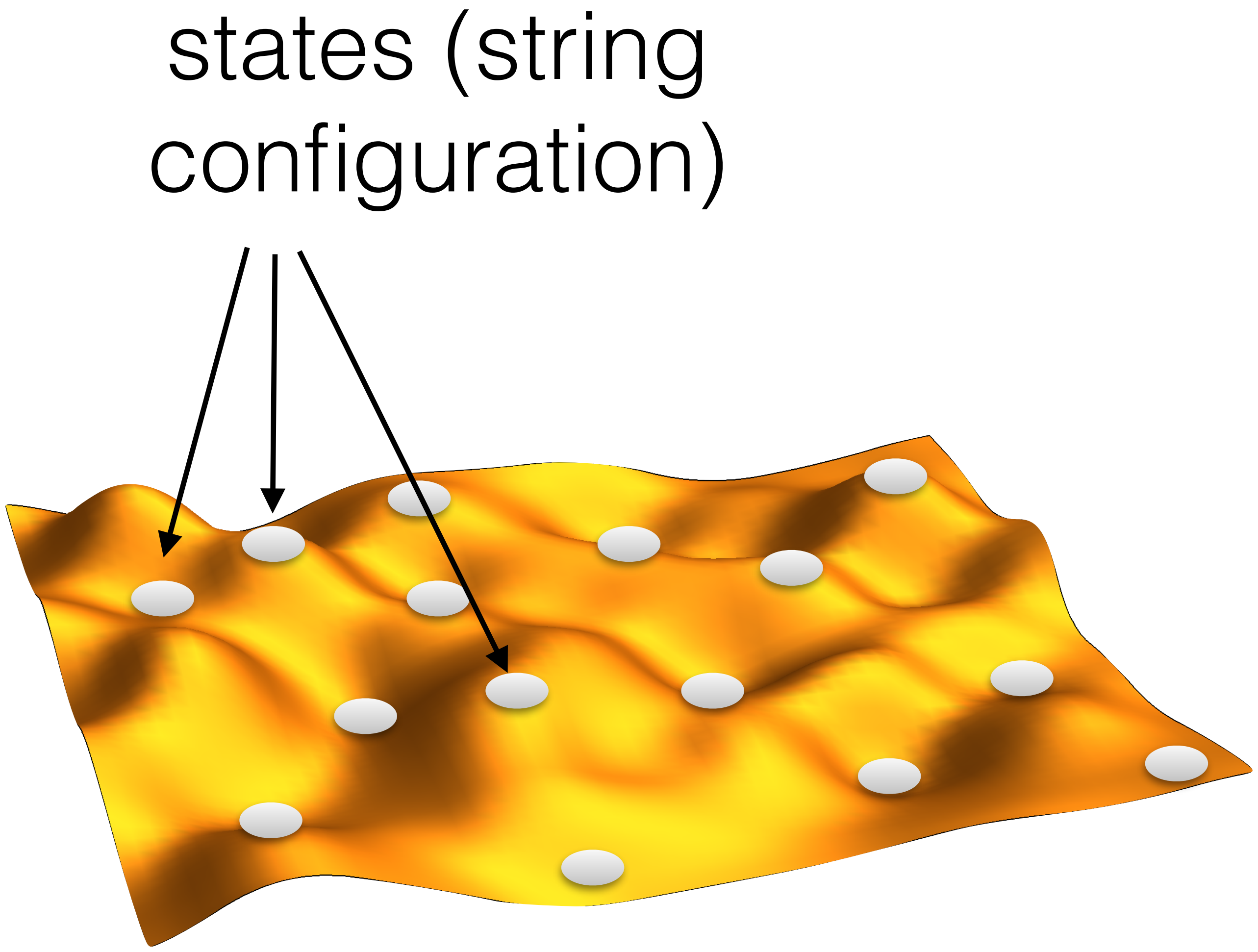
Reinforcement Learning - Vocabulary

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



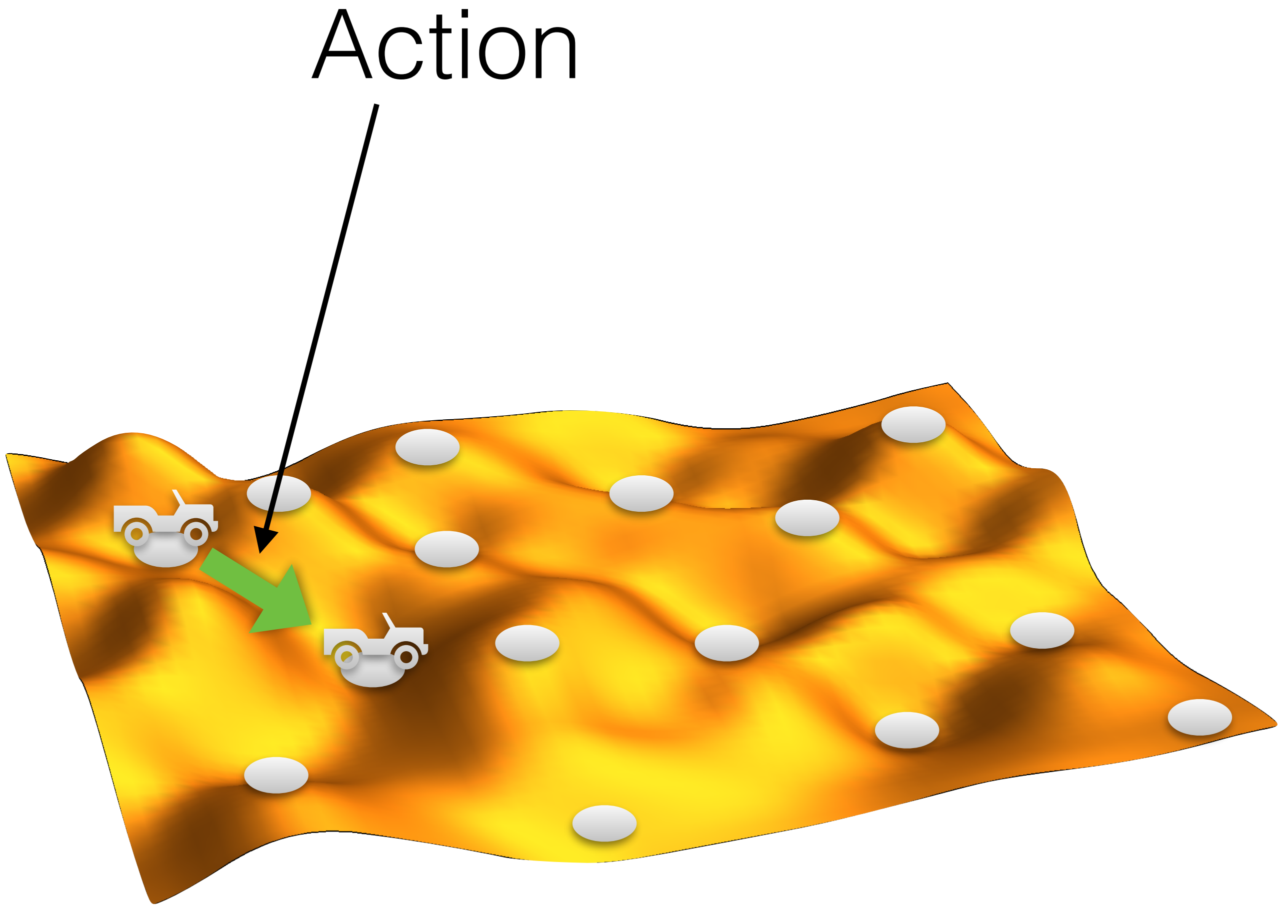
Reinforcement Learning - Vocabulary

- 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, …)



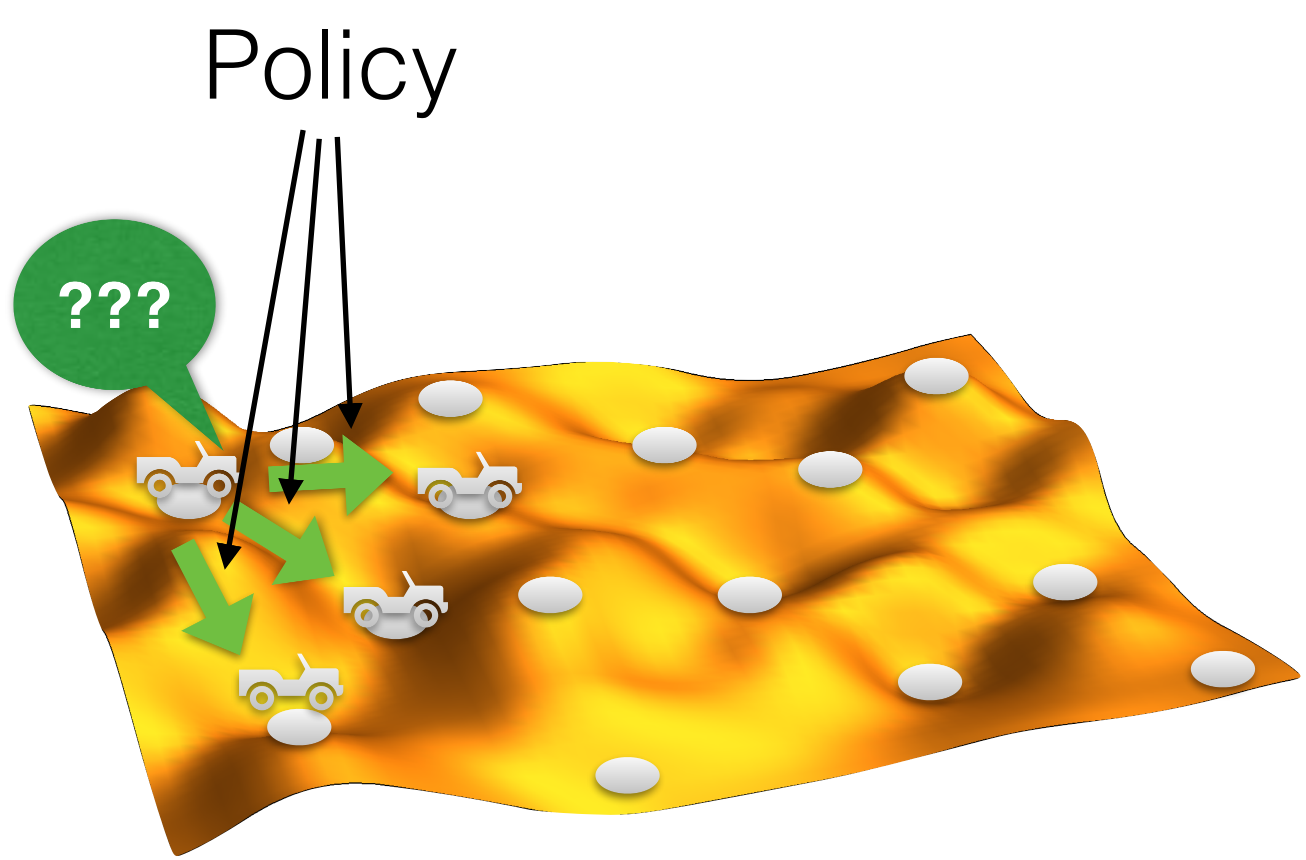
Reinforcement Learning - Vocabulary

- 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")



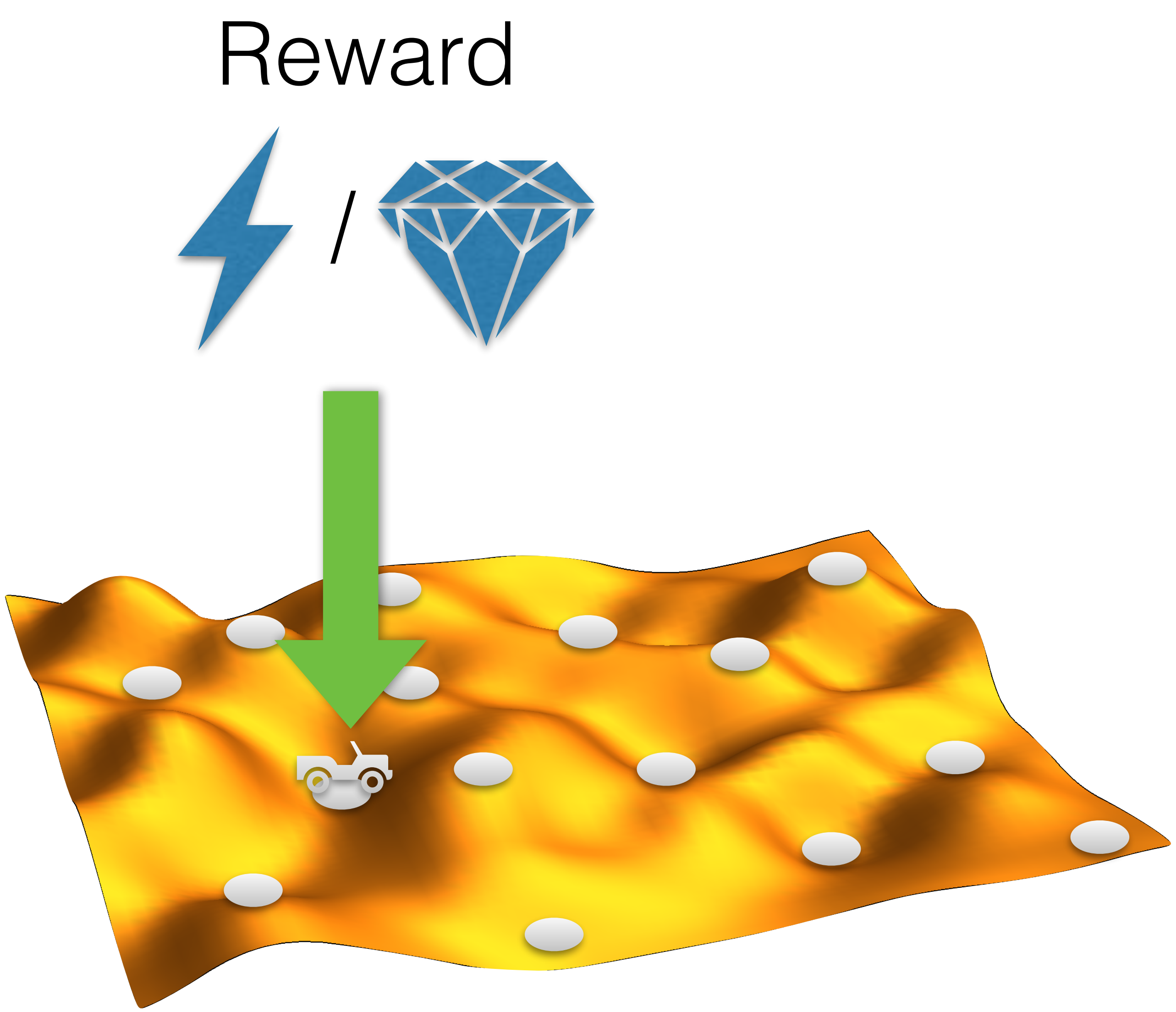
Reinforcement Learning - Vocabulary

- 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"



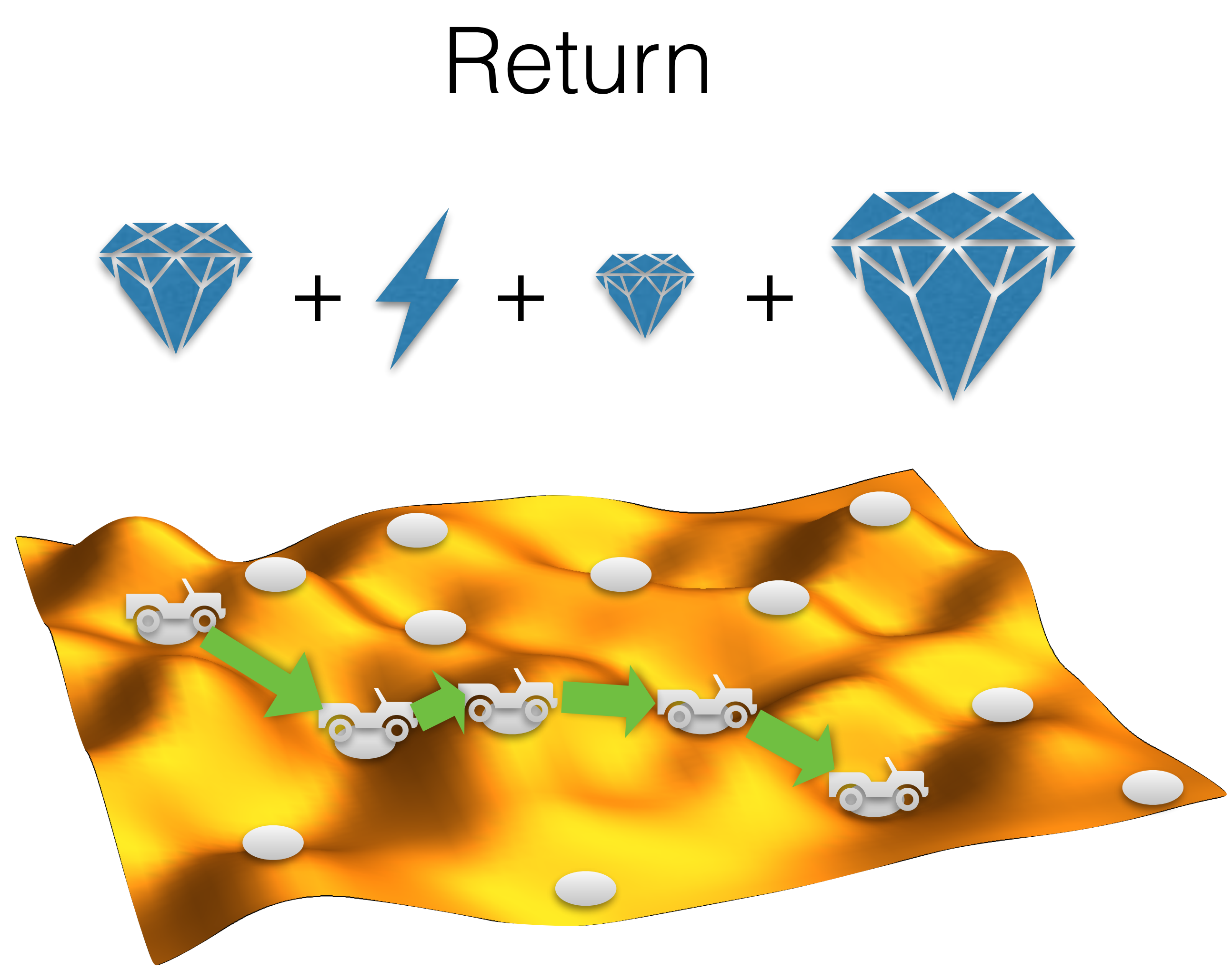
Reinforcement Learning - Vocabulary

- 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"



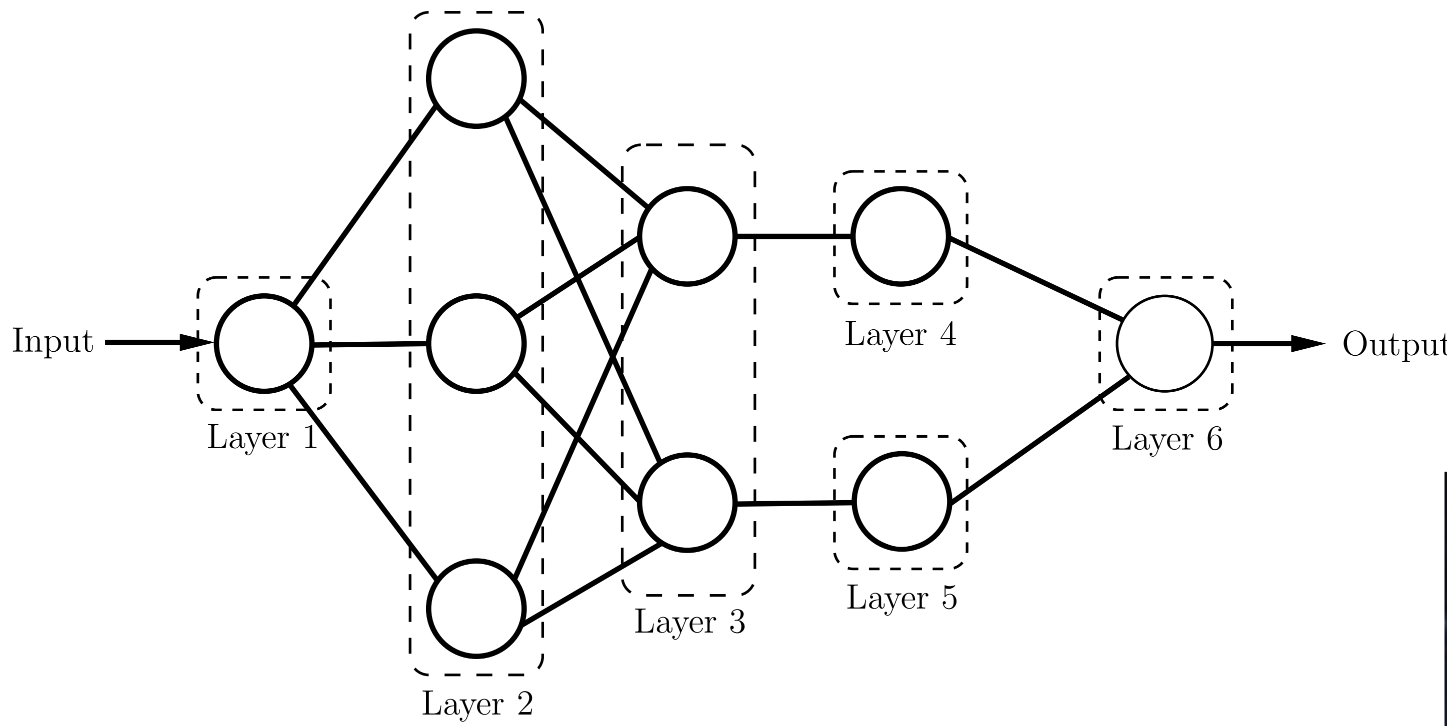
Reinforcement Learning - Vocabulary

- 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)



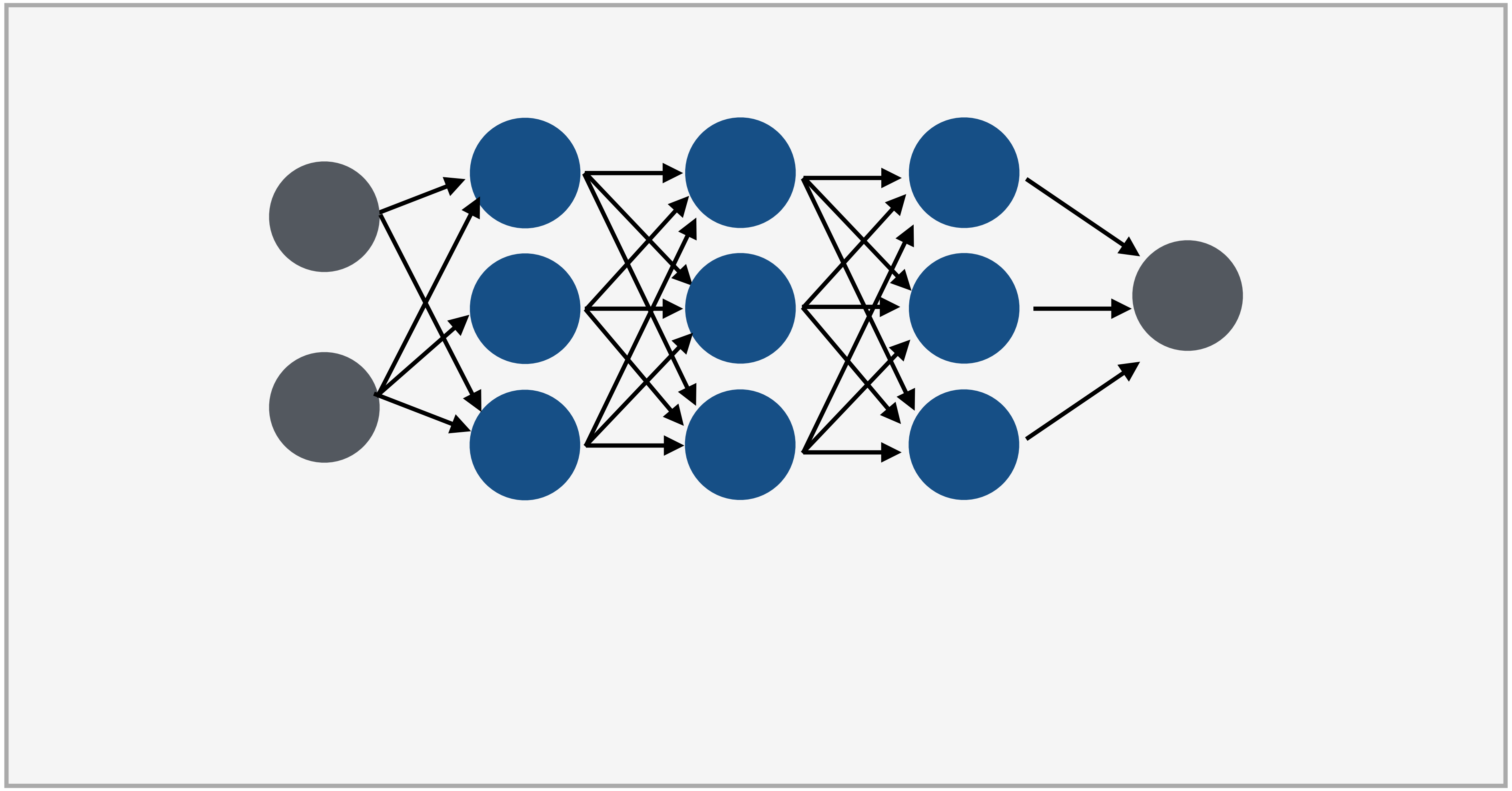
Reinforcement Learning - Prediction Problem

- In order to maximize long-term return, we need to predict:
  1. how beneficial is a given state
  2. how high will the reward of future actions be
- In order to predict this, we use neural networks that learn to make good predictions based on previous experience

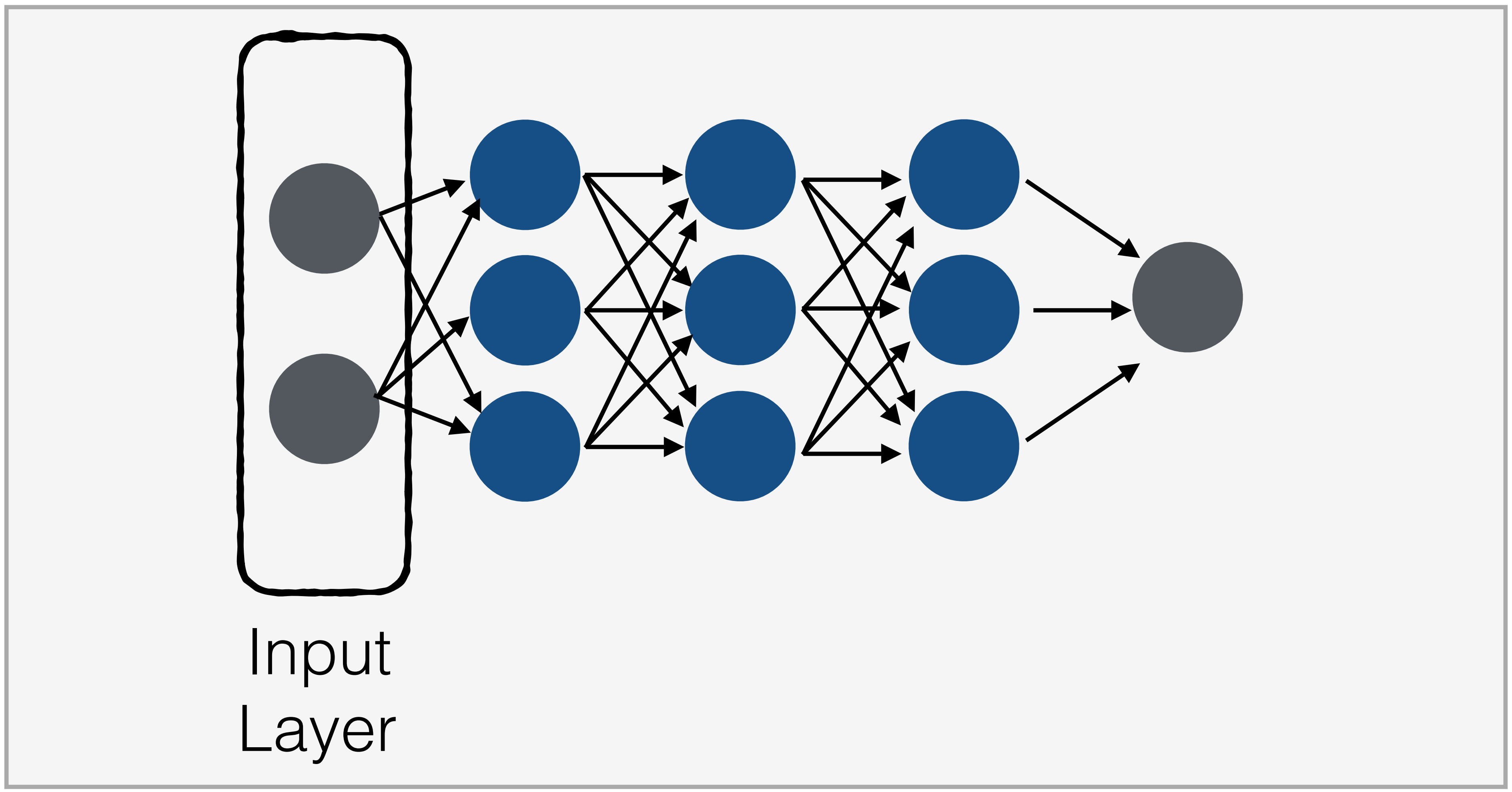


Interlude: Neural Networks

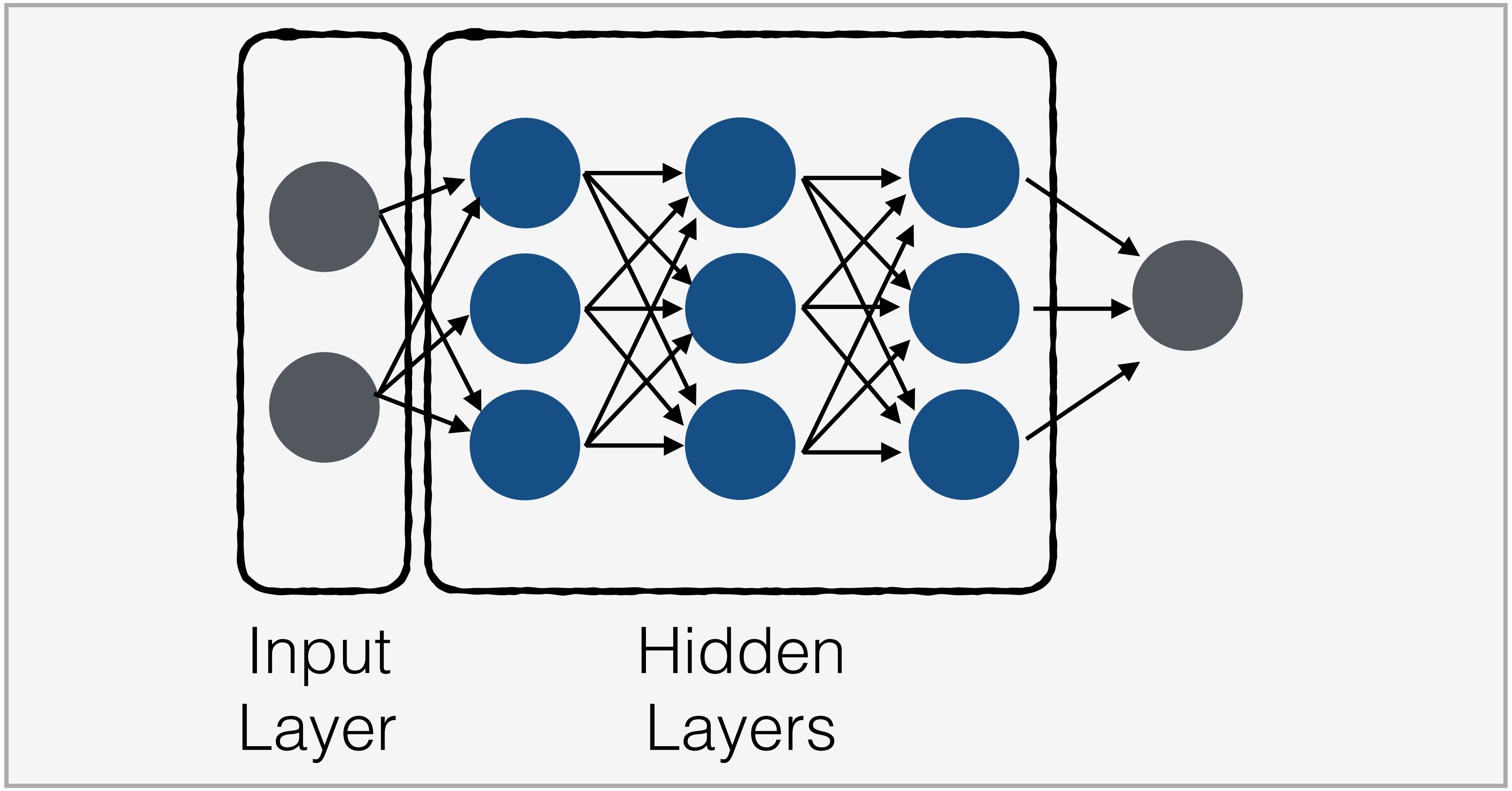
Neural Networks 101



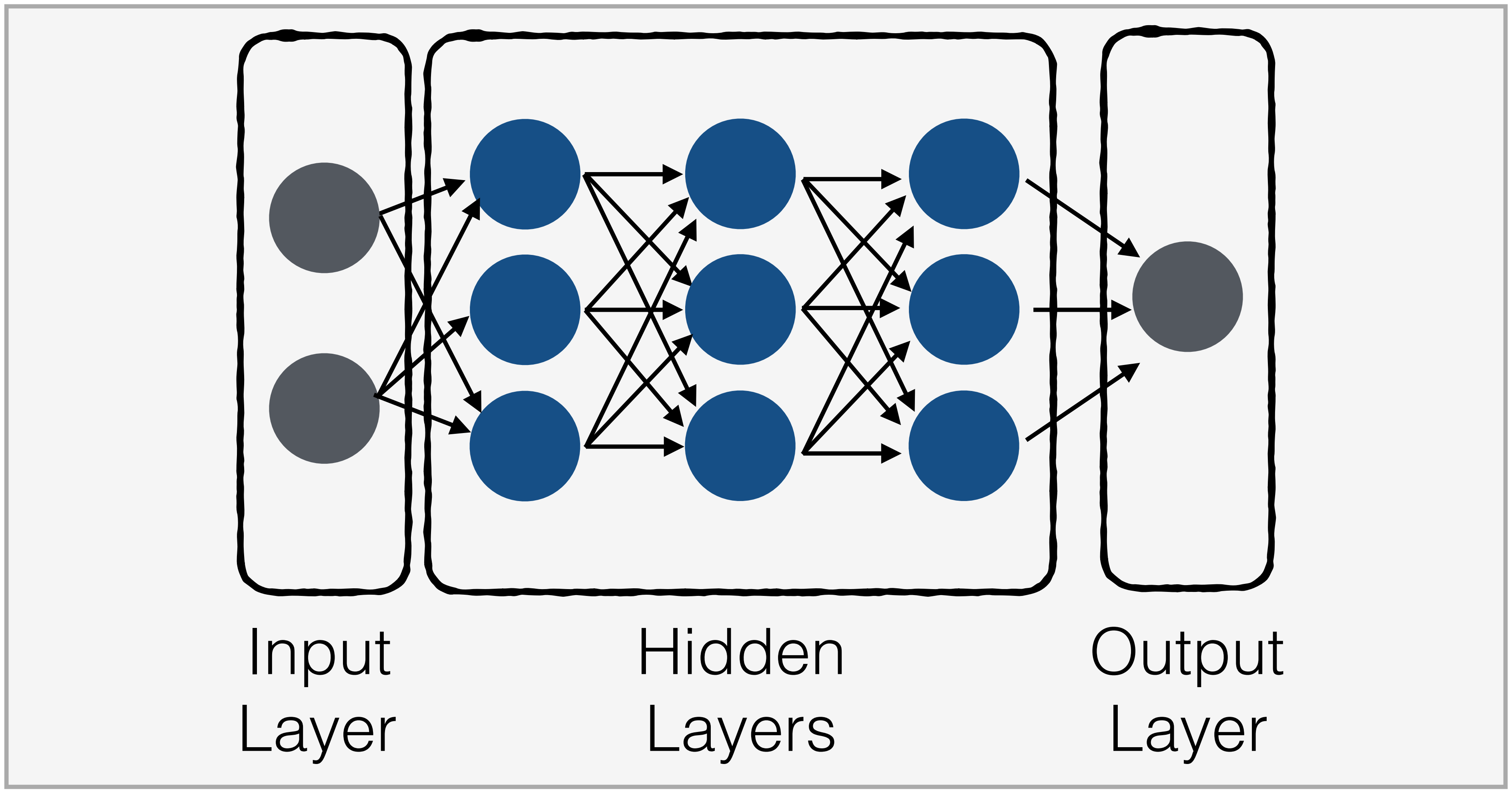
Neural Networks 101



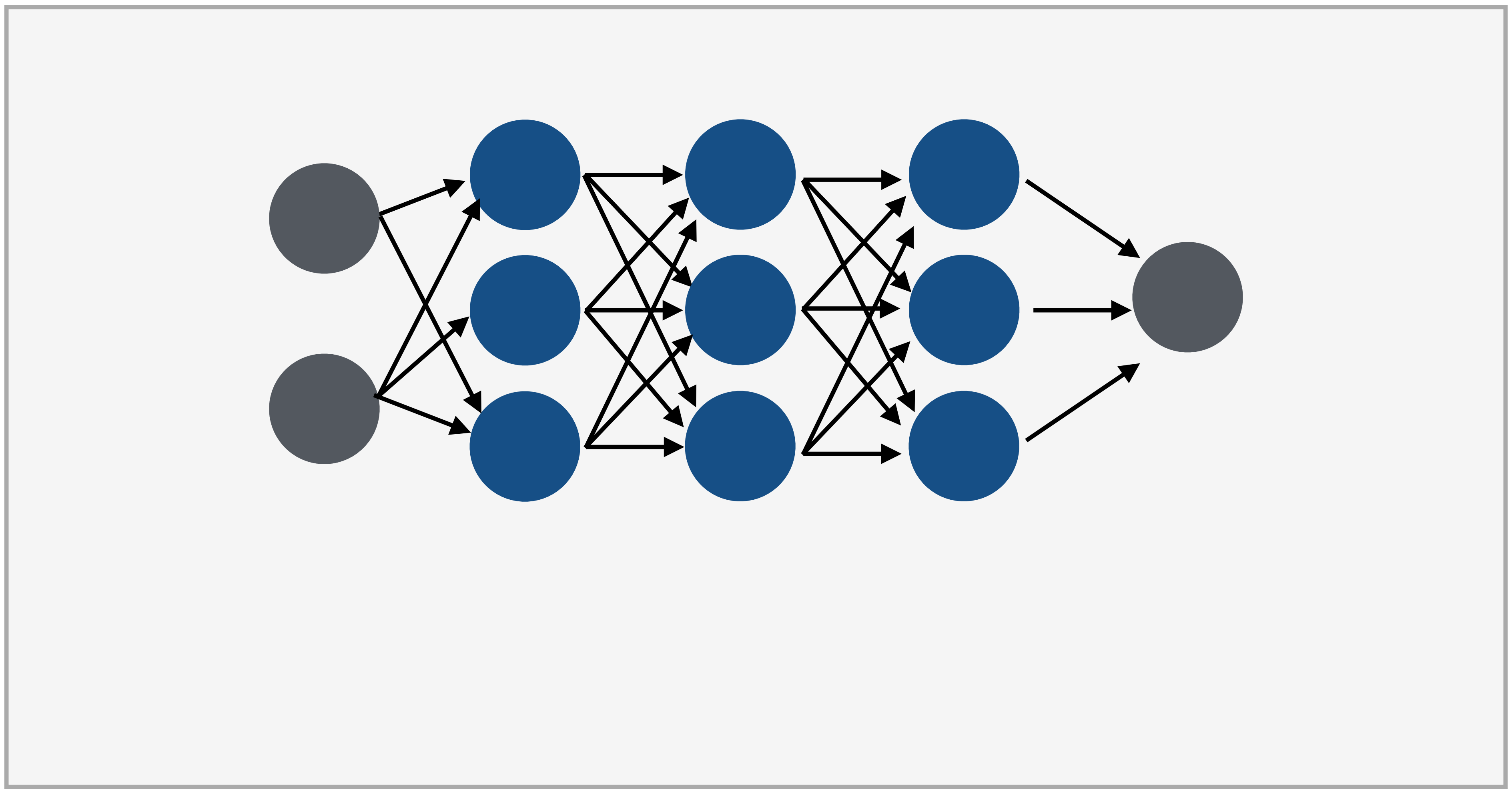
Neural Networks 101



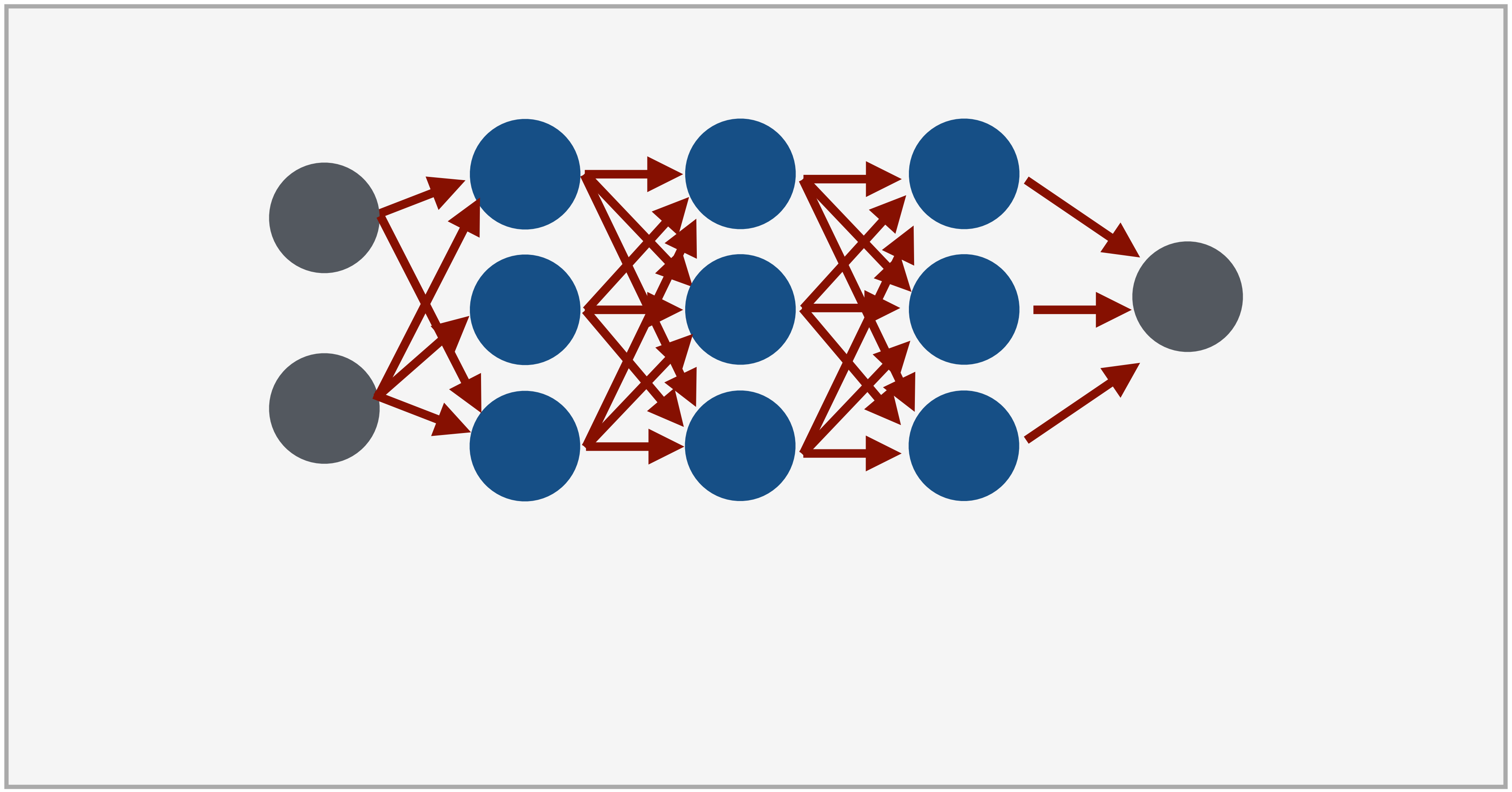
Neural Networks 101



Neural Networks 101

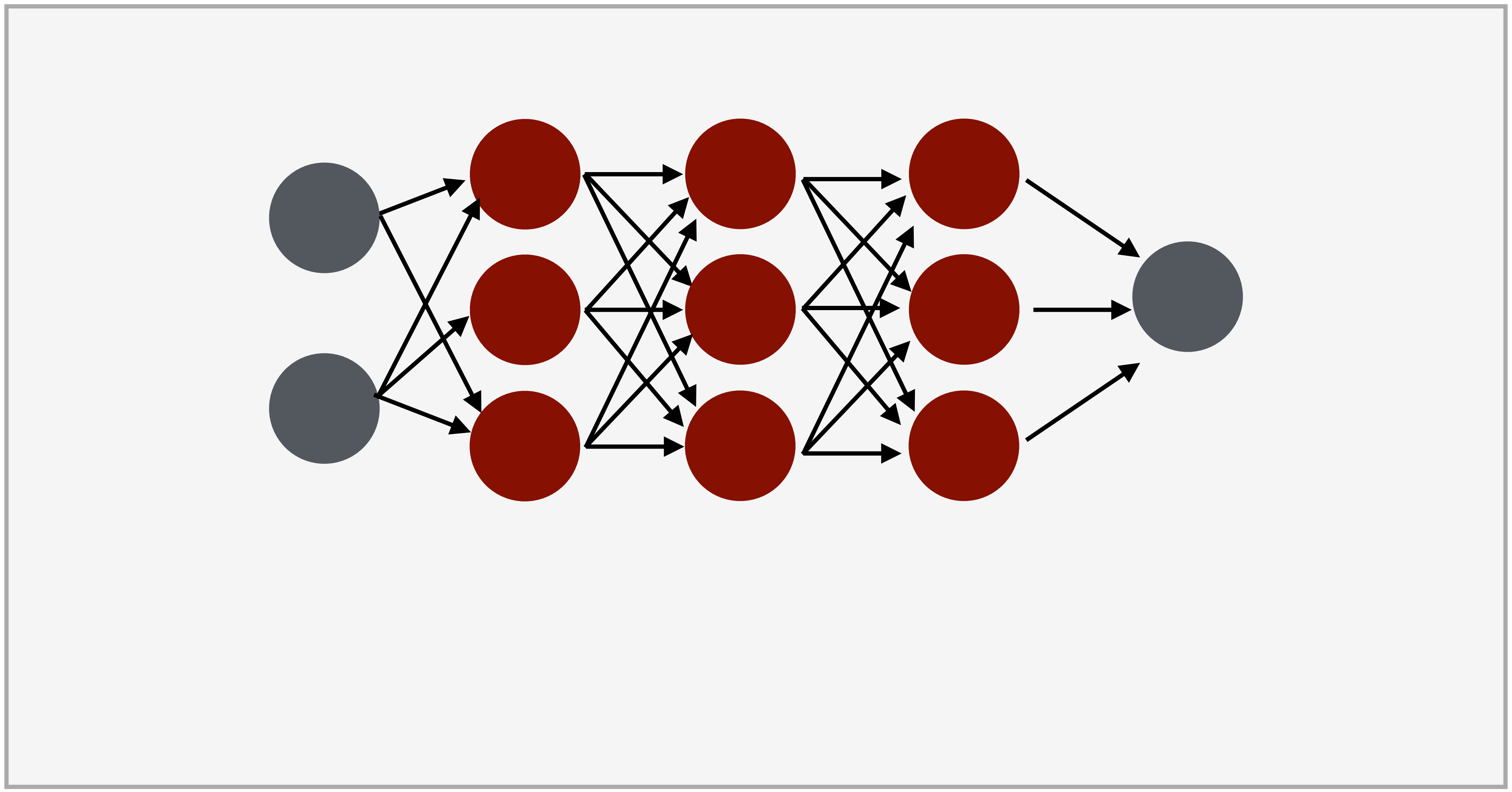


Neural Networks 101



- **Connections**: Matrix Multiplication
- Nodes: Apply some activation function f

Neural Networks 101

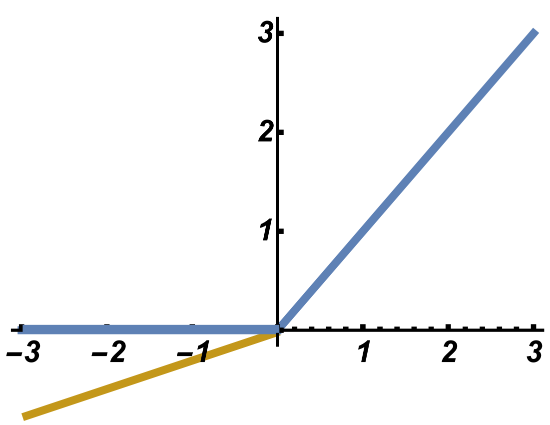


- Connections: Matrix Multiplication
- **Nodes**: Apply some activation function f

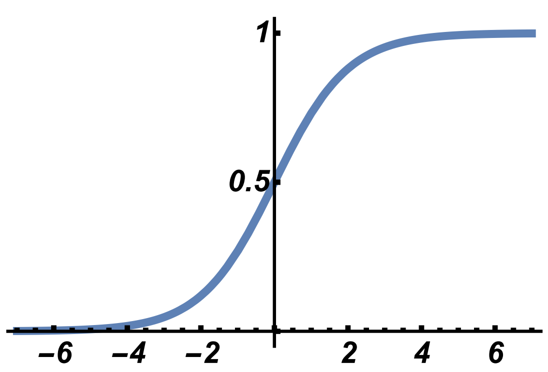
Neural Networks 101

- Connection between layers : Linear transformations L_i:
  Matrix multiplication $v^i_{\text{out}} = A^i v^i_{\text{in}} + b^i$
- Each layer applies a function (activation function) to its input to compute its output. Common choices are

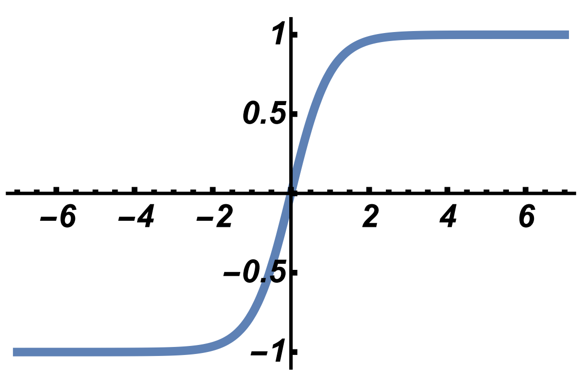
(leaky) ReLu



Logistic Sigmoid



Tanh

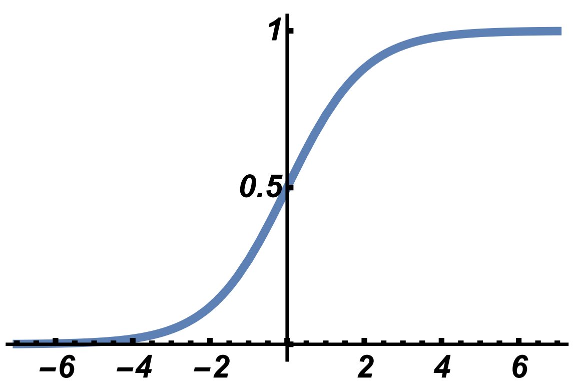


- Typical NN: $\mathbb{R}^M \to \mathbb{R}^N$

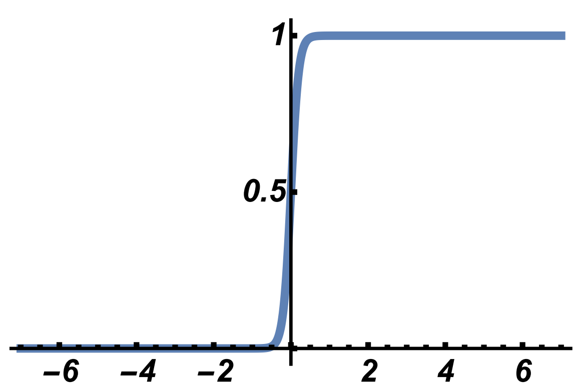
$$v \mapsto f_n \circ L_n \circ \ldots \circ f_0 \circ L_0$$

Neural Networks 101

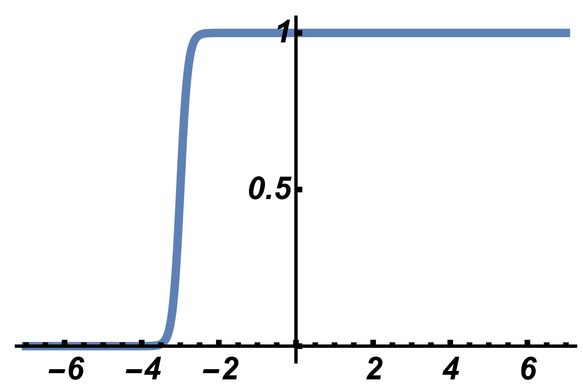
- Look at simplest case: 1 layer, 1 node, logistic sigma function $x_{\text{out}} = (1 + \exp(a x_{\text{in}} + b))^{-1}$
  - a : Steepness of step (step function for $a \to \infty$)
  - b : Position of step: (intersects y-axis at $y = 1/2$ for $b = 0$)



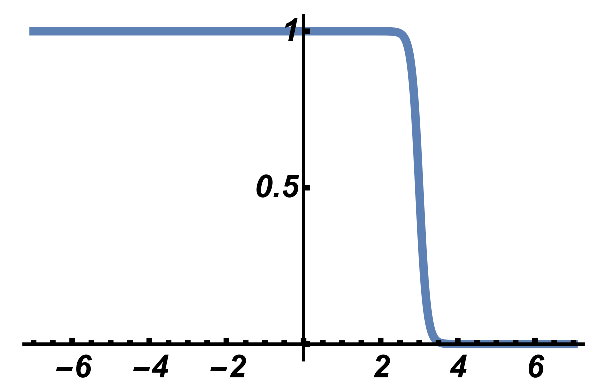
$a = 1, b = 0$



$a = 10, b = 0$

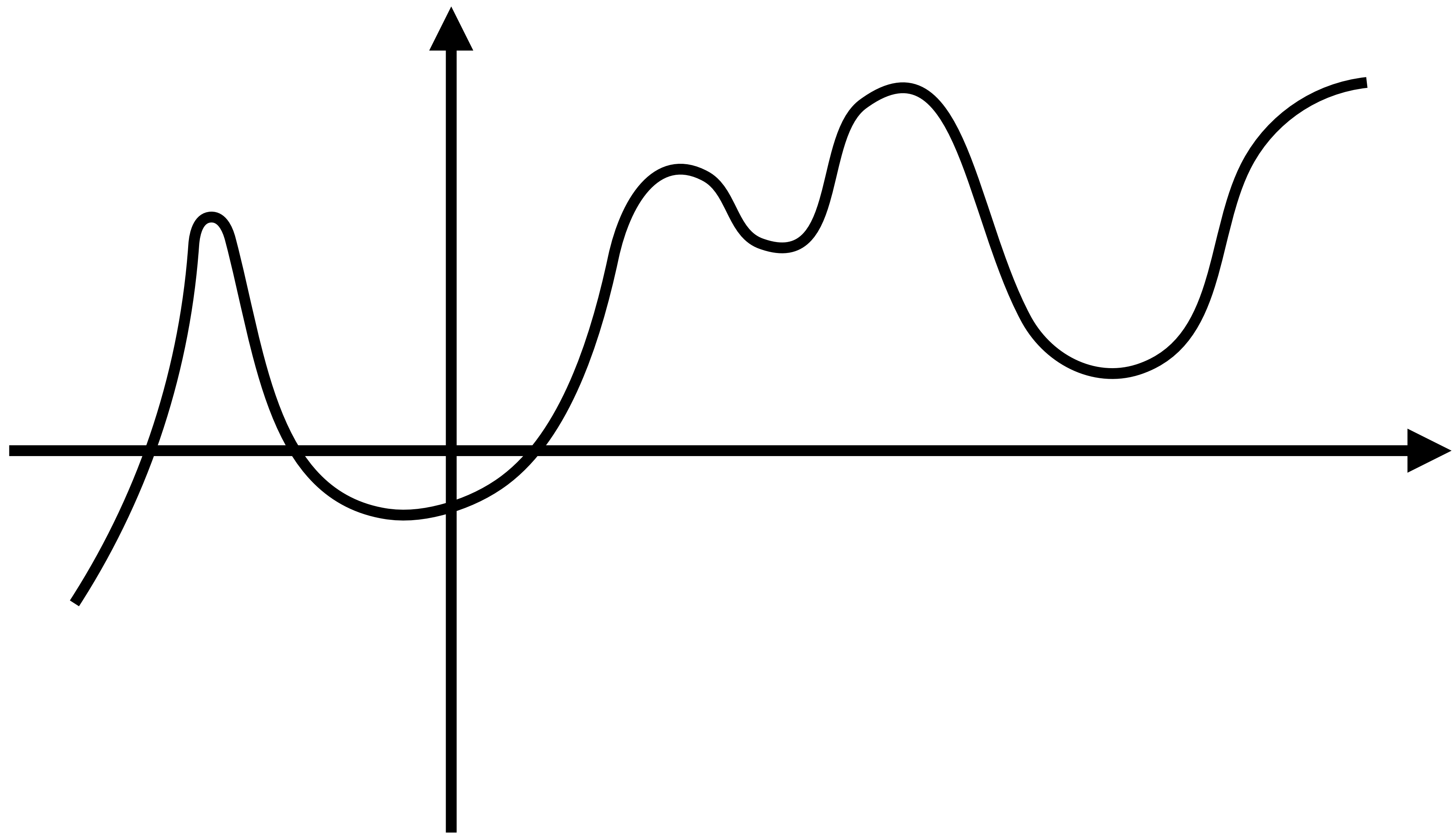


$a = 10, b = -30$

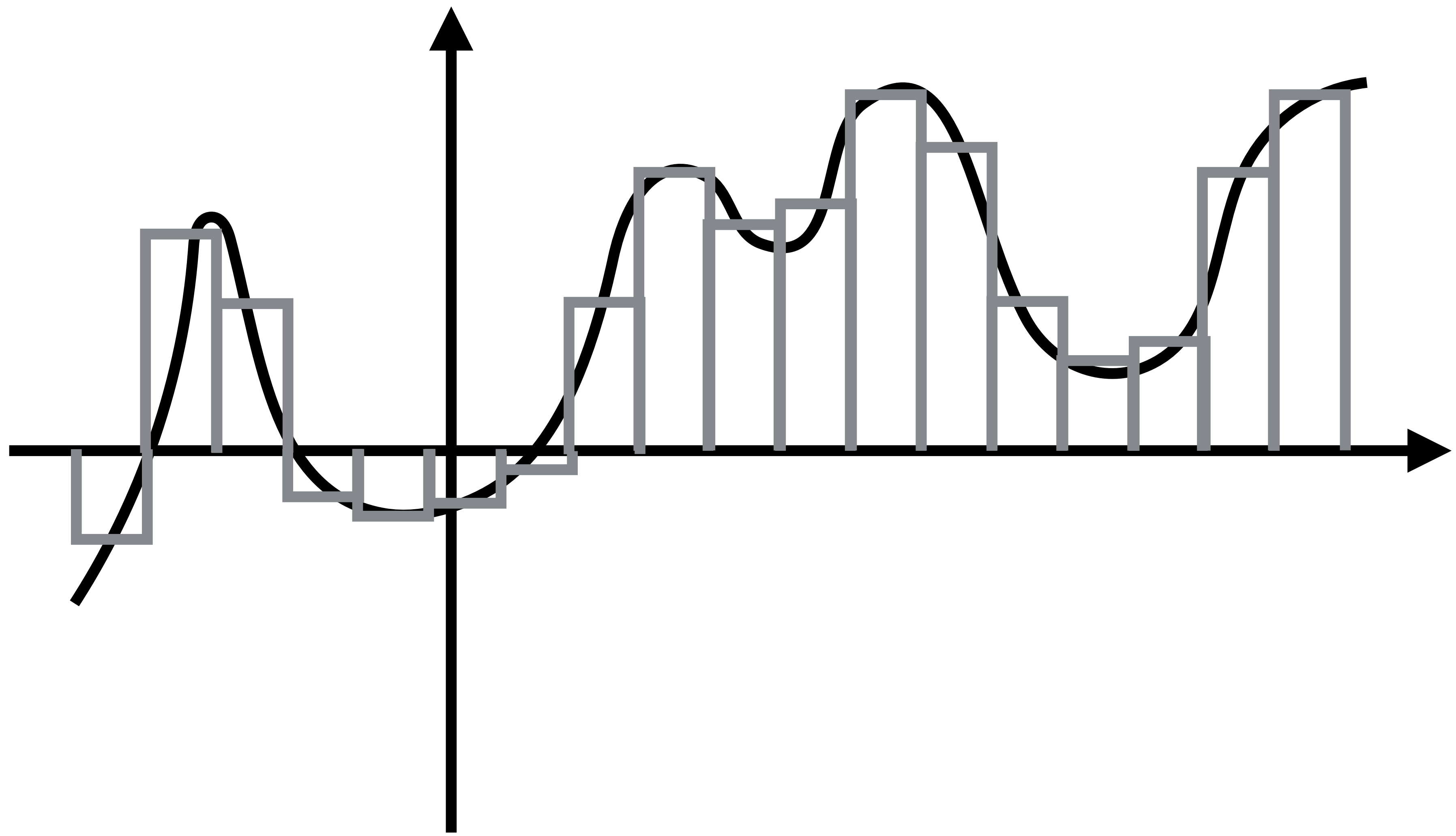


$a = -10, b = 30$

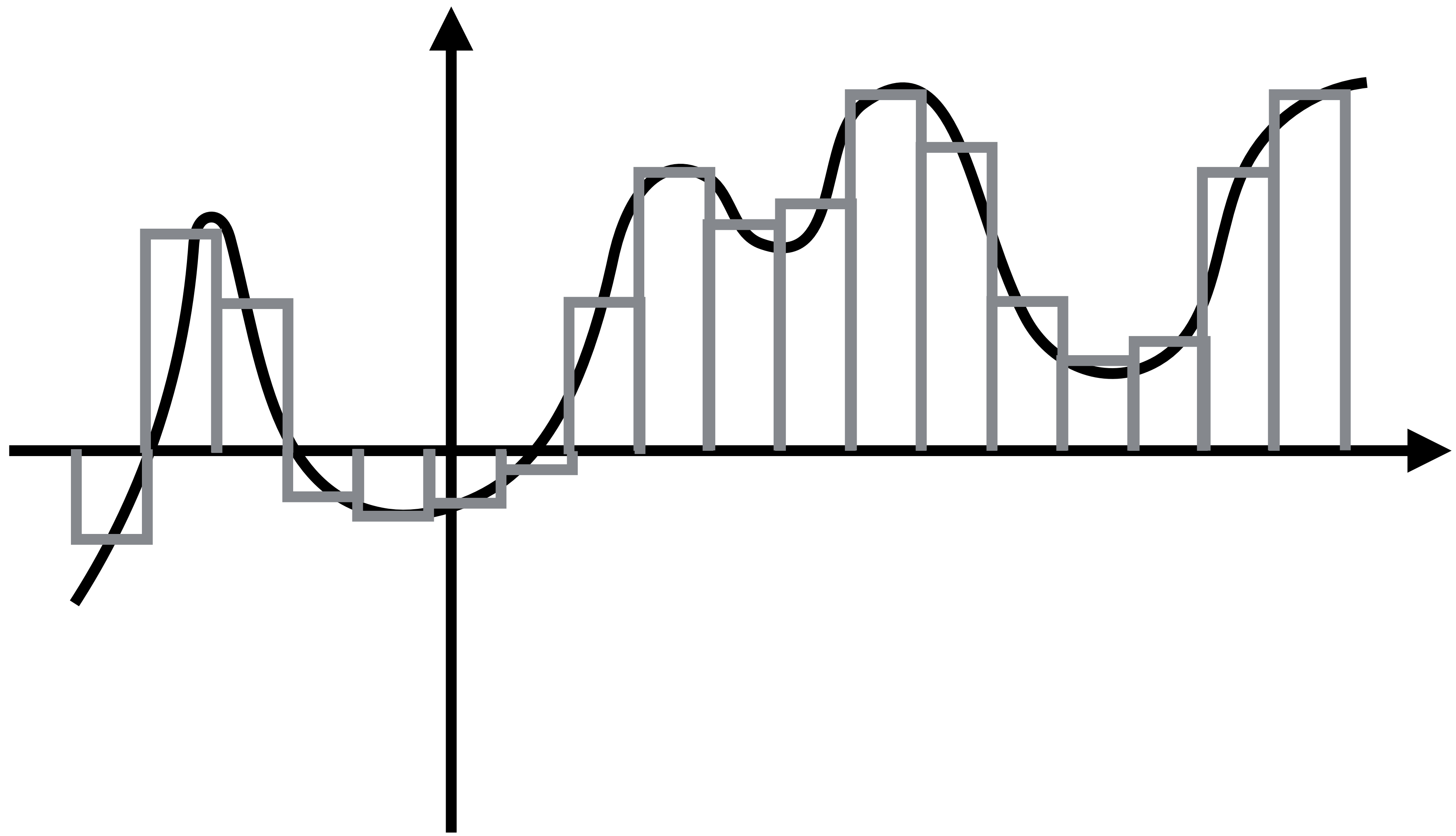
(B) Using NN to approximate functions



(B) Using NN to approximate functions



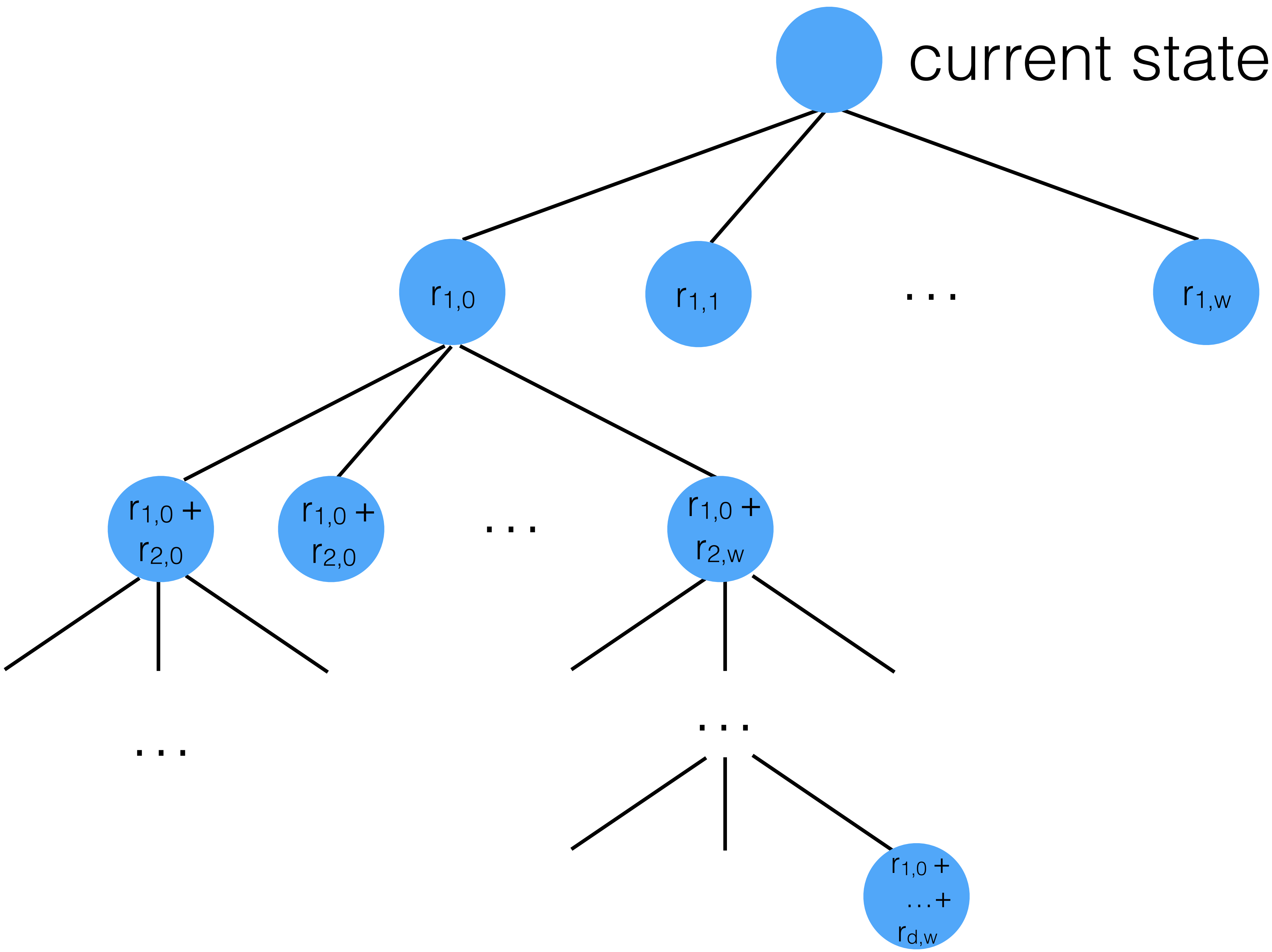
(B) Using NN to approximate functions



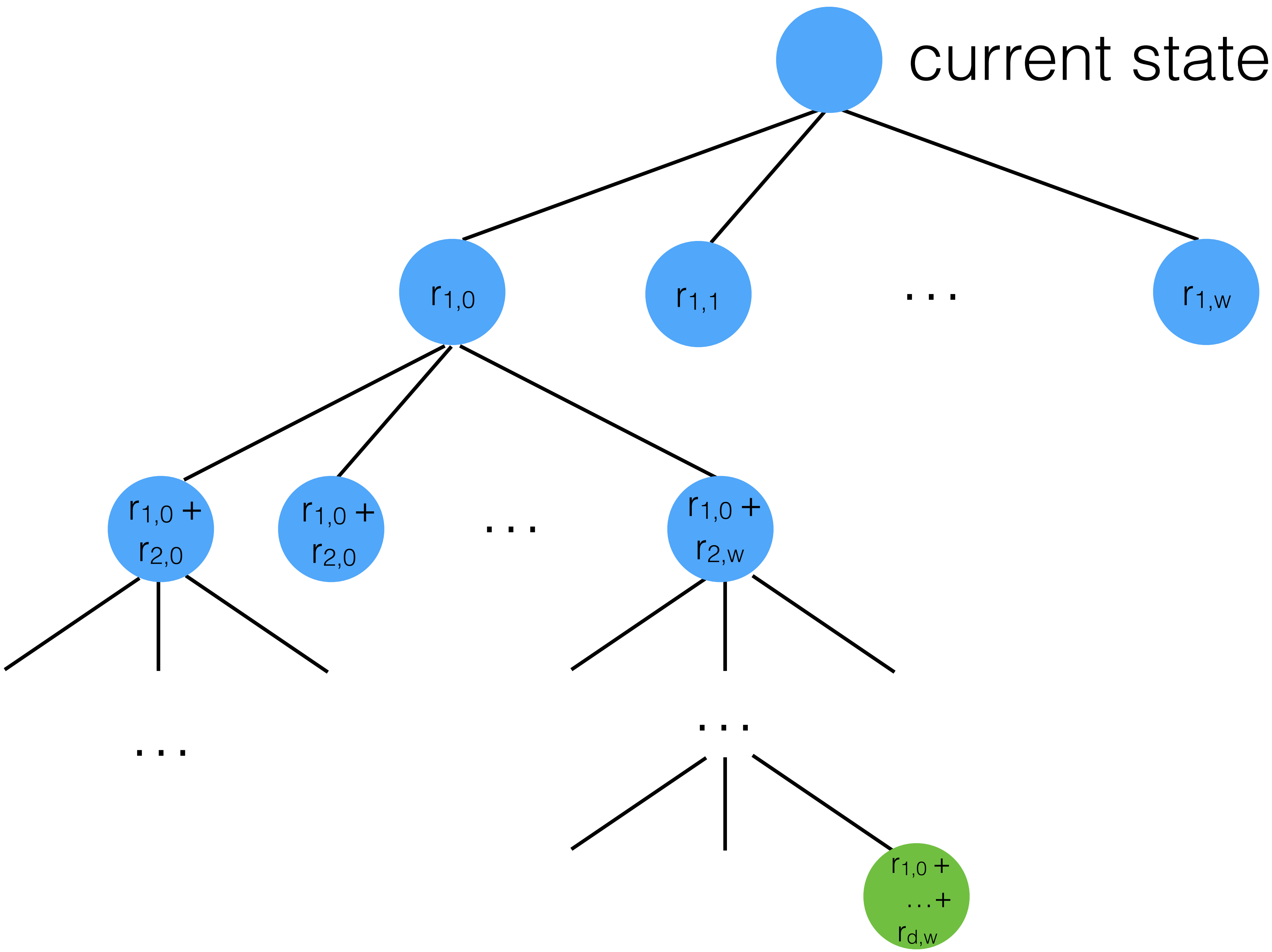
- More nodes $\Rightarrow$ more steps $\Rightarrow$ approximate any function (with one layer) [Cybenko '89; Hornik '91; Nielsen'15]

Interlude: Tree Searches

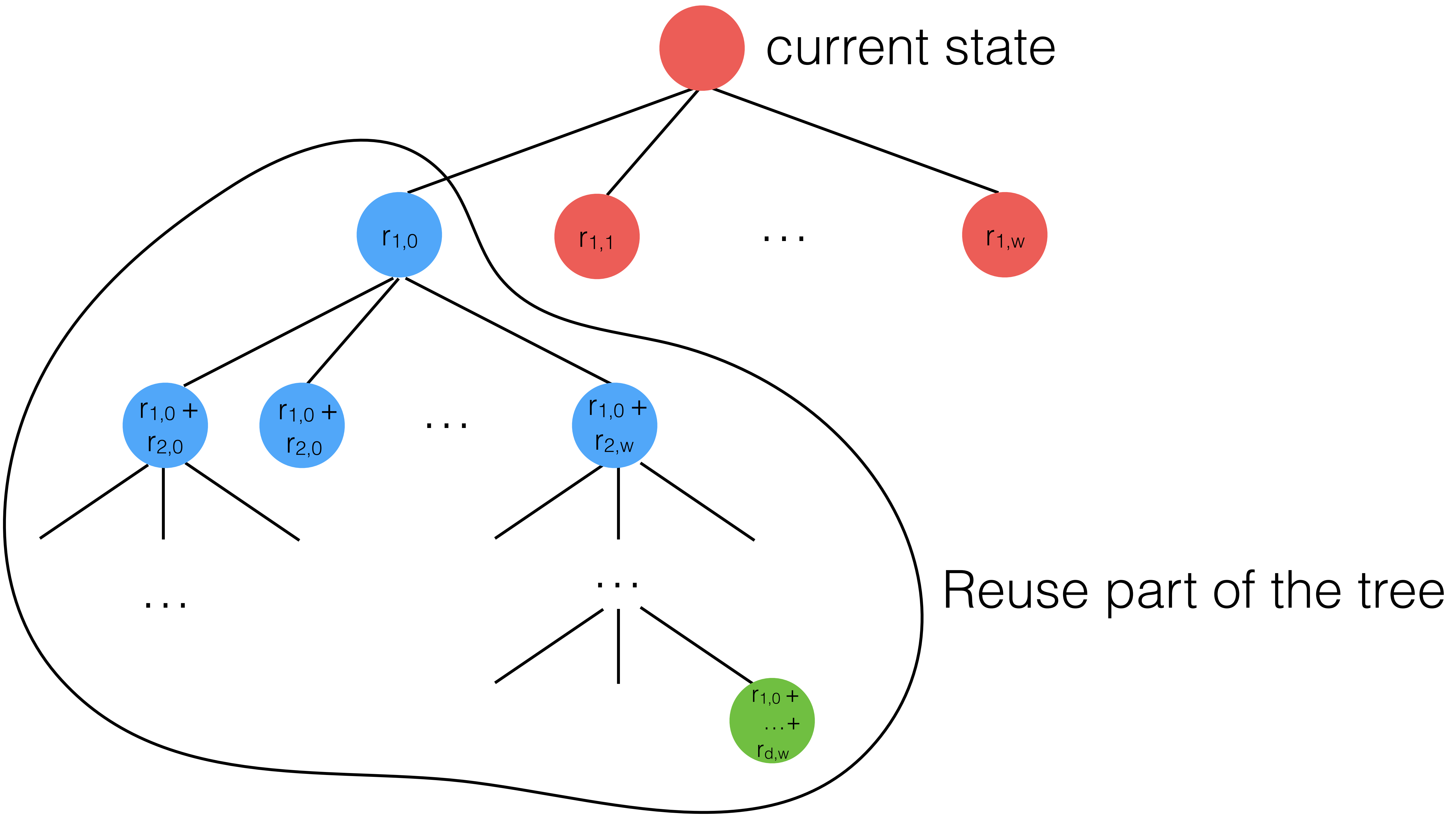
Reinforcement Learning - Tree search



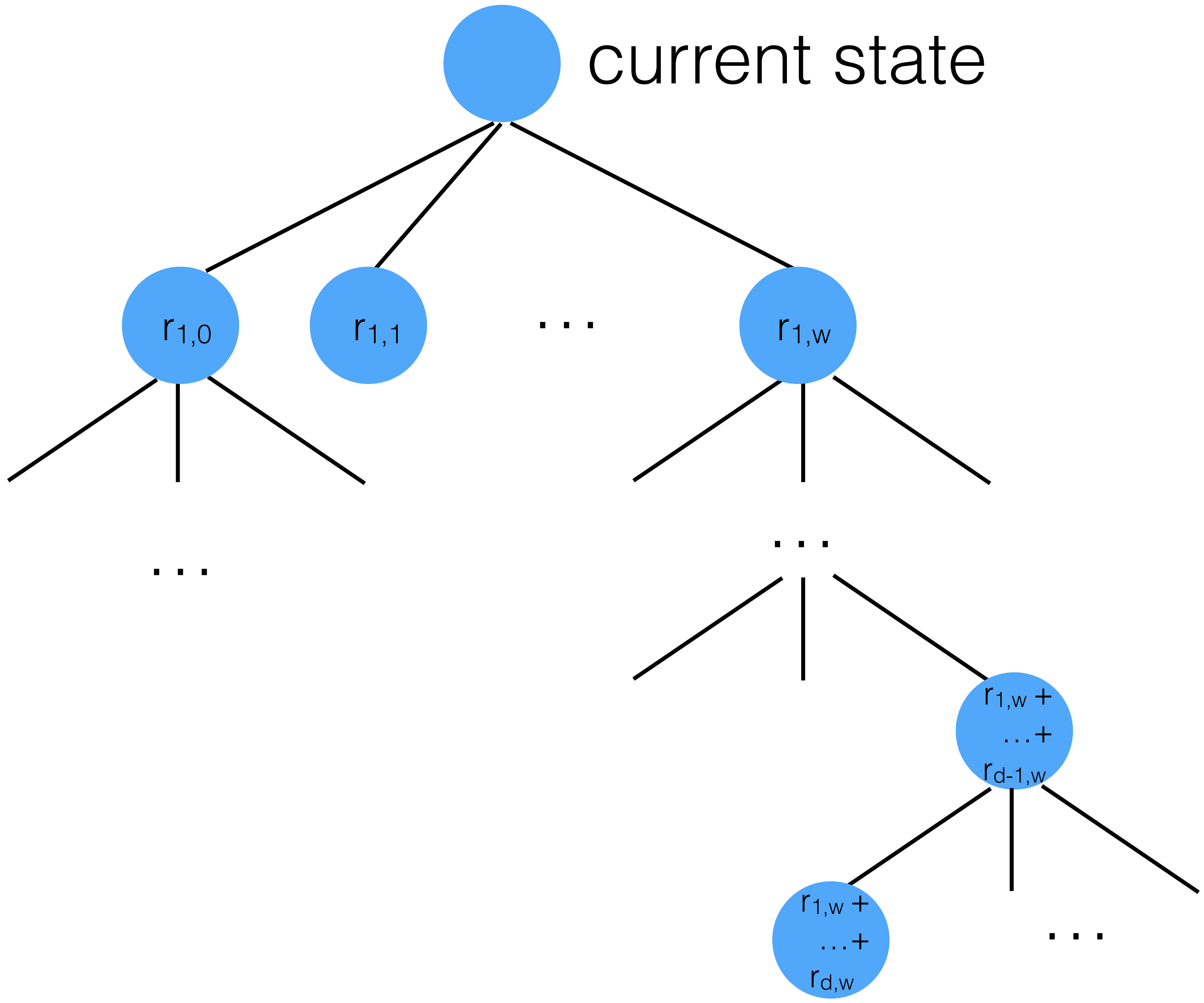
Reinforcement Learning - Tree search



Reinforcement Learning - Tree search



Reinforcement Learning - Tree search



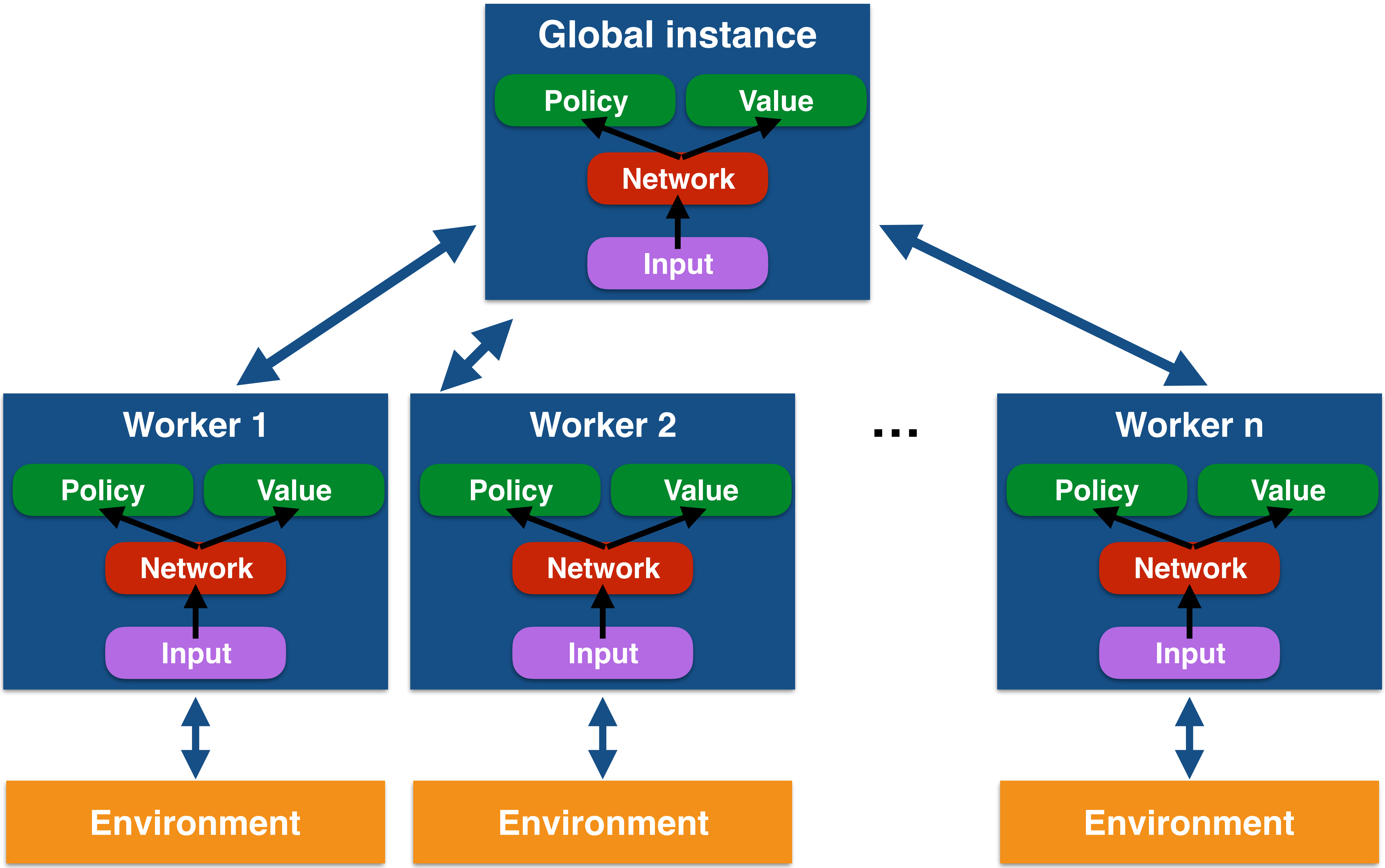
Reinforcement Learning - Details

- Commonly used policies:
  - Greedy: Choose the action that maximizes the action value function: $\pi'(s) = \text{argmax } q(s, a)$
  - ϵ greedy: Choose best action in $1 - \epsilon$ cases and a random action in ϵ cases
  - Draw next action from probability distribution
    $$\pi'(s) = \text{argmax}[\log(q(s, a)) + \text{gumbel}(q(s, a))]$$
    [Mnih et al '16]
  - Perform tree search
- We use ChainerRL implementation of A3C (Asynchronous advantage actor-critic) possibly combined with tree search

Reinforcement Learning - A3C

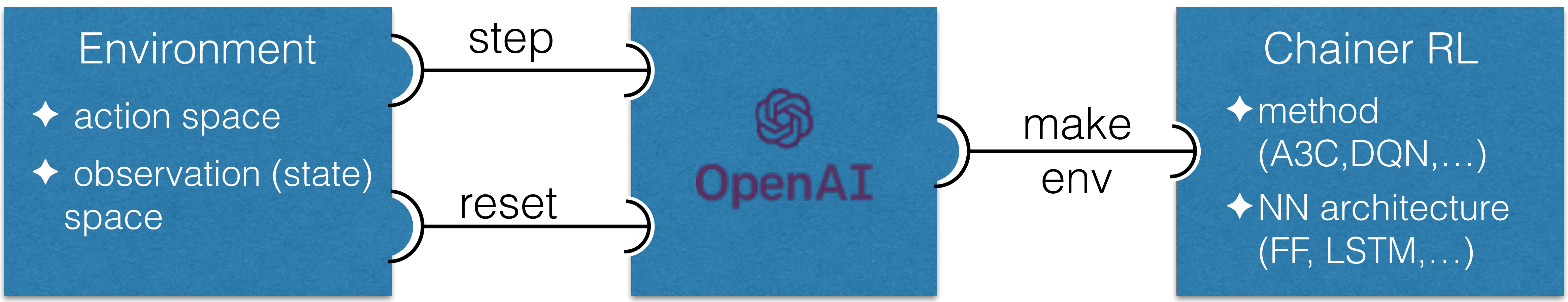
- **Asynchronous**: Have *n* workers explore the environment simultaneously and asynchronously, report back to overseer
    - improves training stability (experience of workers separated)
    - improves exploration
- **Advantage**: Use advantage to update policy
- **Actor-critic**: To maximize return we need to select the best next state (actor) based on an estimate of the value of the state (critic). Both is done using NNs.

Reinforcement Learning - A3C



Reinforcement Learning - Implementation

- Open AI Gym: Interface between agent (RL) and environment (string landscape) [Brockman et al '16]
  - We provide the environment
  - We use ChainerRL's implementation of A3C for the agent

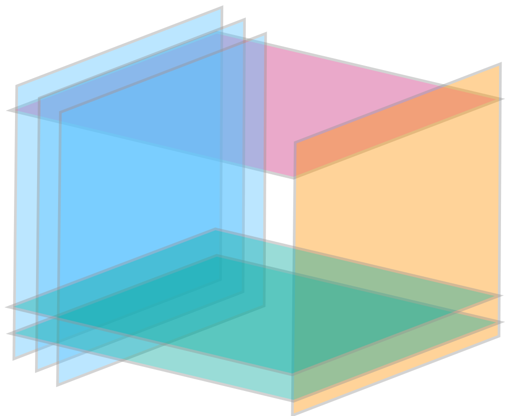


Chainer RL

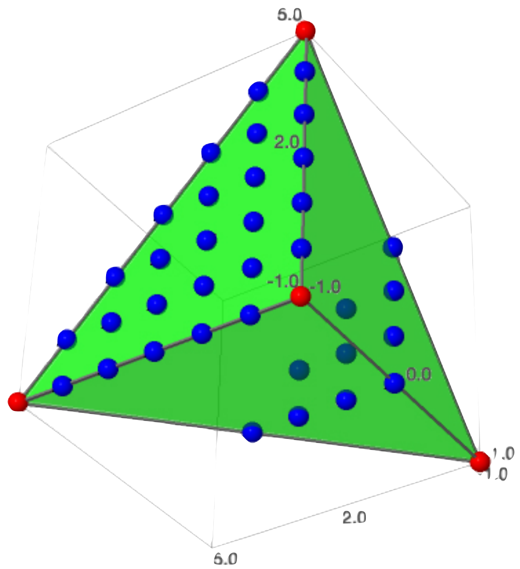
- step:
  - go to new state
  - return (new_state, reward, done, comment)
- reset:
  - reset episode
  - return start_state

- make environment
- specify RL method (A3C)
- specify policy NN (FF,LSTM)

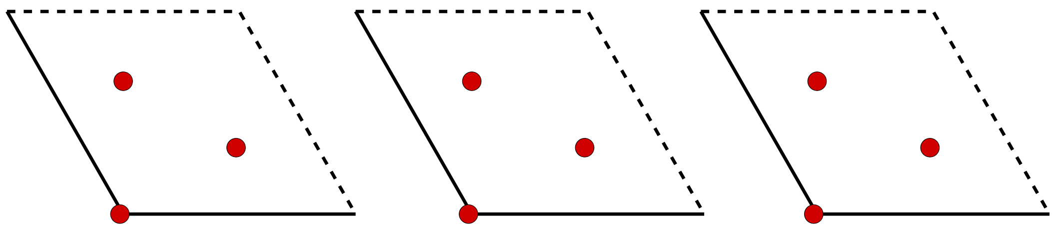
Reinforcement Learning - Versatile Applications



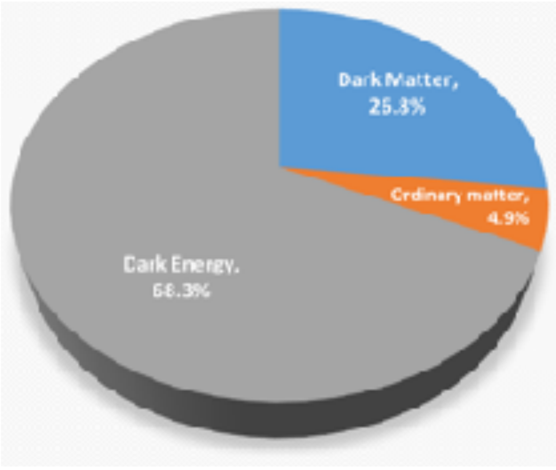
Type II Intersecting branes
Orientifolds of toroidal orbifolds



Sen limits in F-Theory

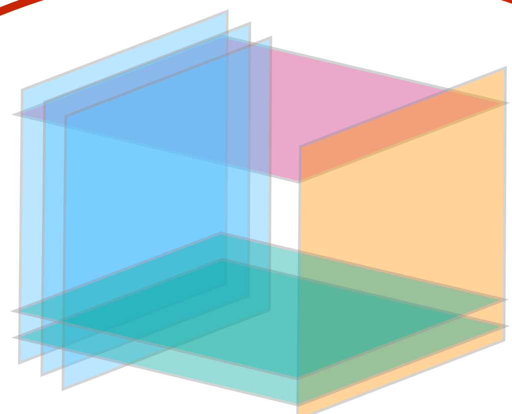


Realistic vacua
of Heterotic Orbifolds

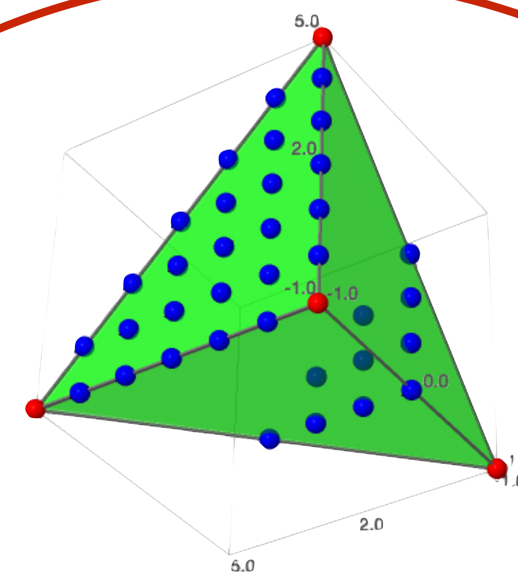


Finding small CC in
Bousso-Polchinski

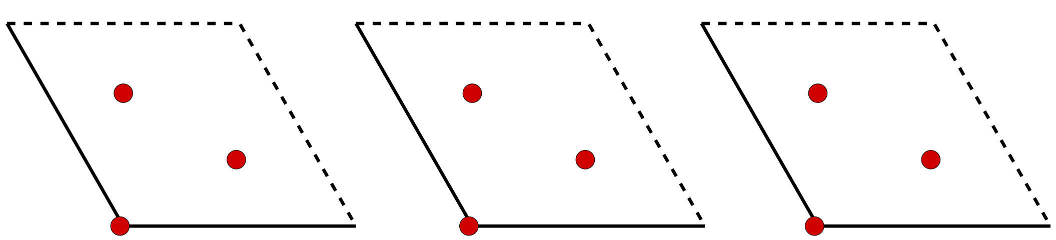
Reinforcement Learning - Versatile Applications



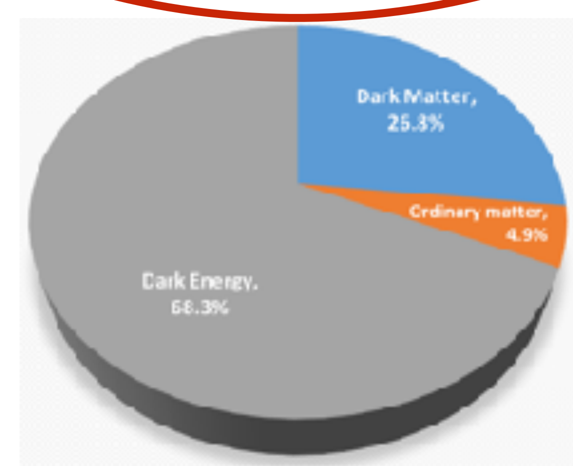
Type II Intersecting branes
Orientifolds of toroidal orbifolds



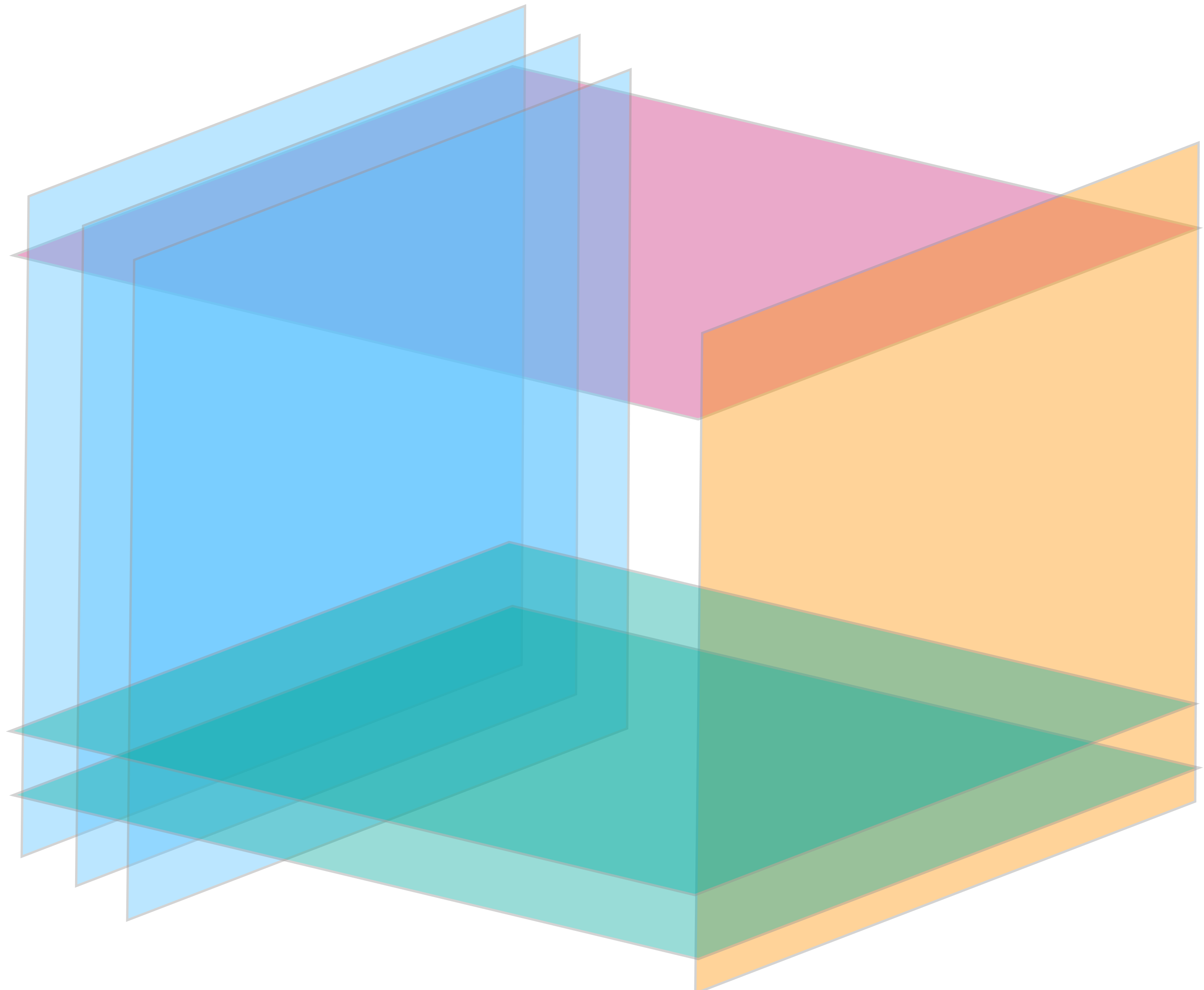
Sen limits in F-Theory



Realistic vacua
of Heterotic Orbifolds



Finding small CC in
Bousso-Polchinski

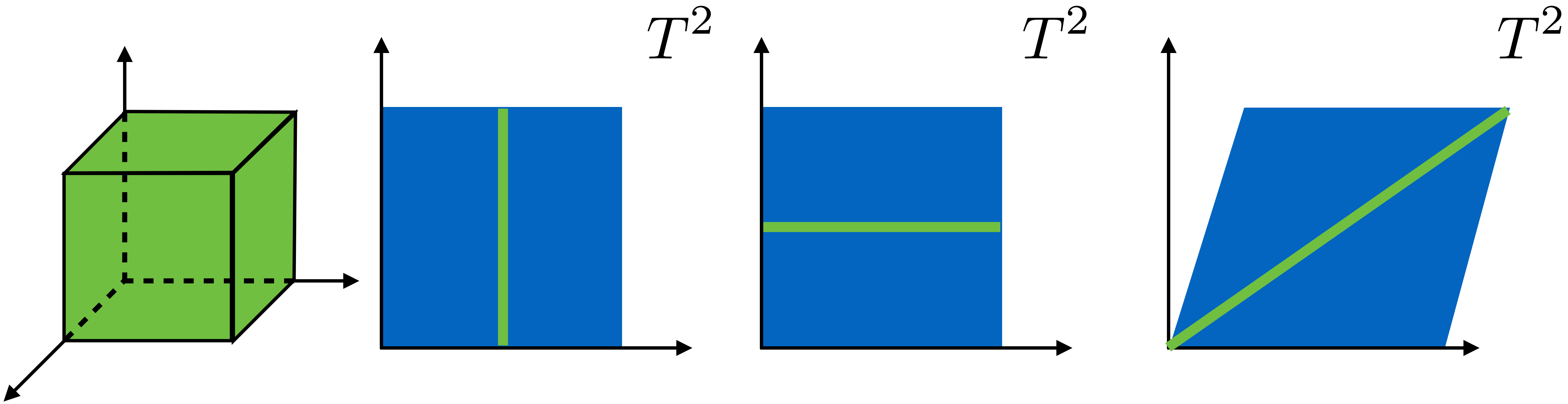


Type II Orientifolds

IIA Orientifolds

- Why this setup?
  - Well studied [Blumenhagen,Gmeiner,Honecker,Lust,Weigand '04'05; Douglas, Taylor '07, ...]
  - Comparatively simple
  - Number of (well-defined) solutions known to be finite: [Douglas, Taylor '07]
    - ✦ Use symmetries to relate different vacua
    - ✦ Combine consistency conditions to rule out combinations
  - BUT: Number of possibilities so large that not a single "interesting" solution could be found despite enormous random scans (estimated to 1:10^9)
  - Interesting to study with big data / AI methods

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: $\forall a = 1, \dots, \#\ \text{stacks}$

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

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

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

$$i = 1, 2, 3 \quad \text{(three tori)}$$
$$k = 1, \dots, \#U \ \text{brane stacks}$$
$$r = 1, \dots, \dim(\ker(\{N^k m_i^k\}))$$
$$= k - 3 \ \text{(generically)}$$

TypeII RL - Model the environment

- State space: $s_t \in S, \quad |S| = N_{\max}^{N_S} \binom{N_B}{N_S}$

$$s_t = [(N^1, n_1^1, m_1^1, n_2^1, m_2^1, n_3^1, m_3^1), (N^2, n_1^2, \ldots), \ldots]$$

- Action space: Two approaches

  - Construct collection of winding number 6-tuples.
    Actions can add/remove branes from the brane stacks or exchange entire 6-tuples from pool of constructed stacks
    $$A = \{N^a \to N^a \pm 1, \text{ add stack } (N, n_1, \ldots), \\ \text{remove stack } (N, n_1, \ldots)\}$$

  - Start with all winding numbers zero. Actions can add/remove branes from the brane stacks or add ± 1 to any winding number in any stack
    $$A = \{N^a \to N^a \pm 1, \; n_i^a \to n_i^a \pm 1, \; m_i^a \to n_i^a \pm 1\}$$

TypeII RL - Model the environment

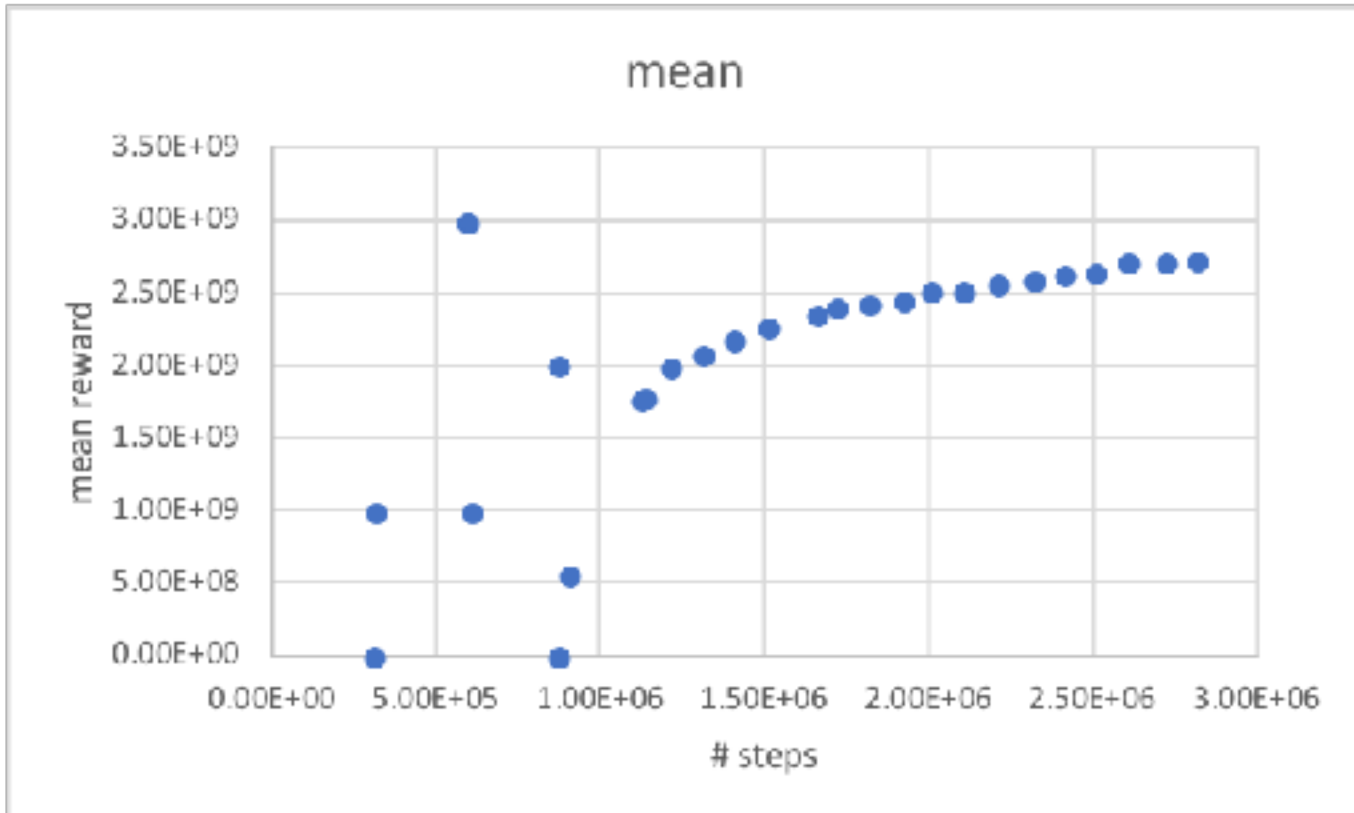
- Reward R : Need a notion of "how good a state is"
  1. By how much does a set of stacks violate the tadpole?
  2. Is a set of stacks fully consistent (Tadpole, K-Theory, SUSY) (Note: the latter two are binary, hard to define distance)
  3. How far is the state from the Standard Model
     - Missing a group factor of $SU(3) \times SU(2) \times U(1)$?
     - Too few Standard model particles $(Q, u, d, L, H_u, H_d, e)$?
     - Extra exotics (particles charged under the Standard Model but not observed so far)

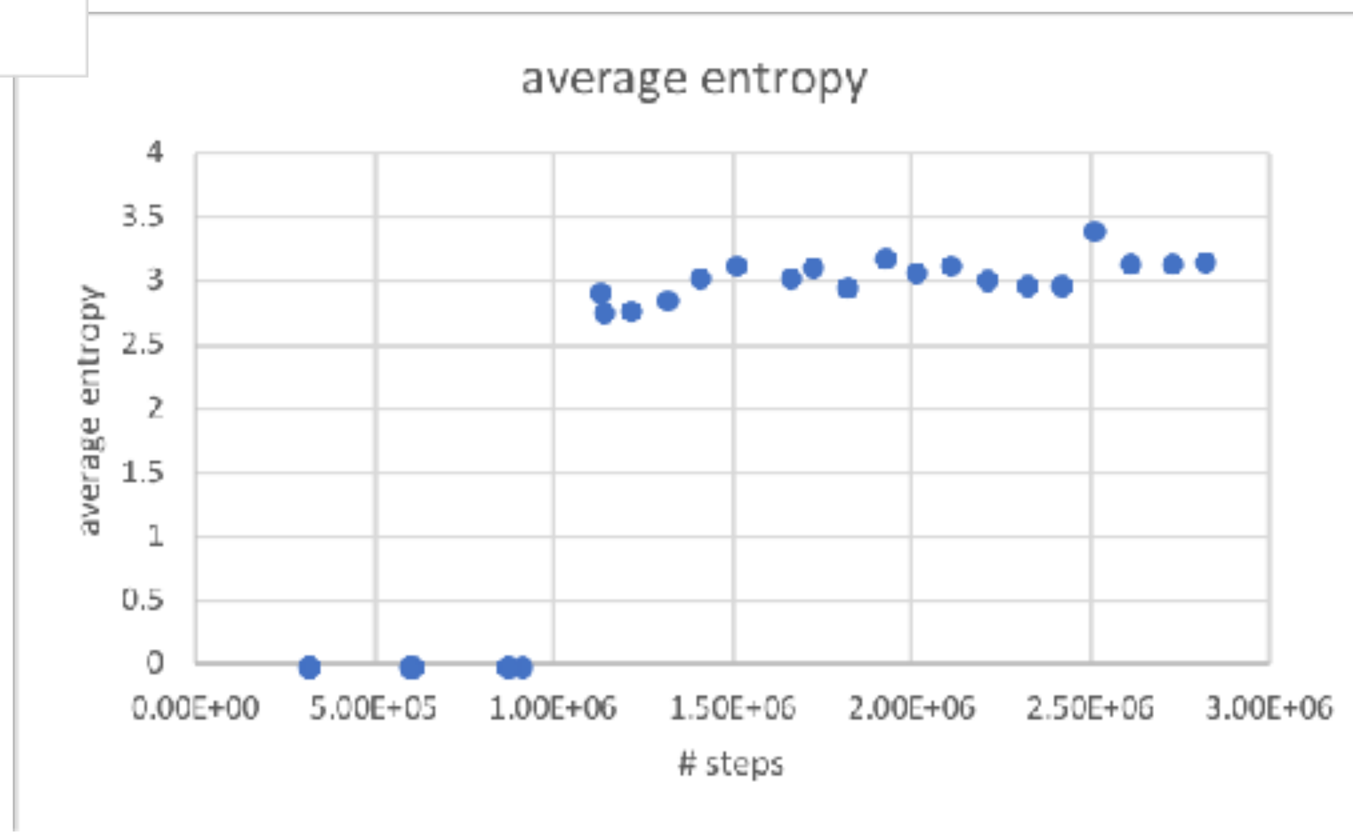
  Note: Only works if good states are "close by" in this sense…
- Need multi-task RL:
  - Check properties consecutively/simultaneously and use different reward hierarchies for different tasks
  - Split up async workers and let them prioritise different goals

Preliminary results

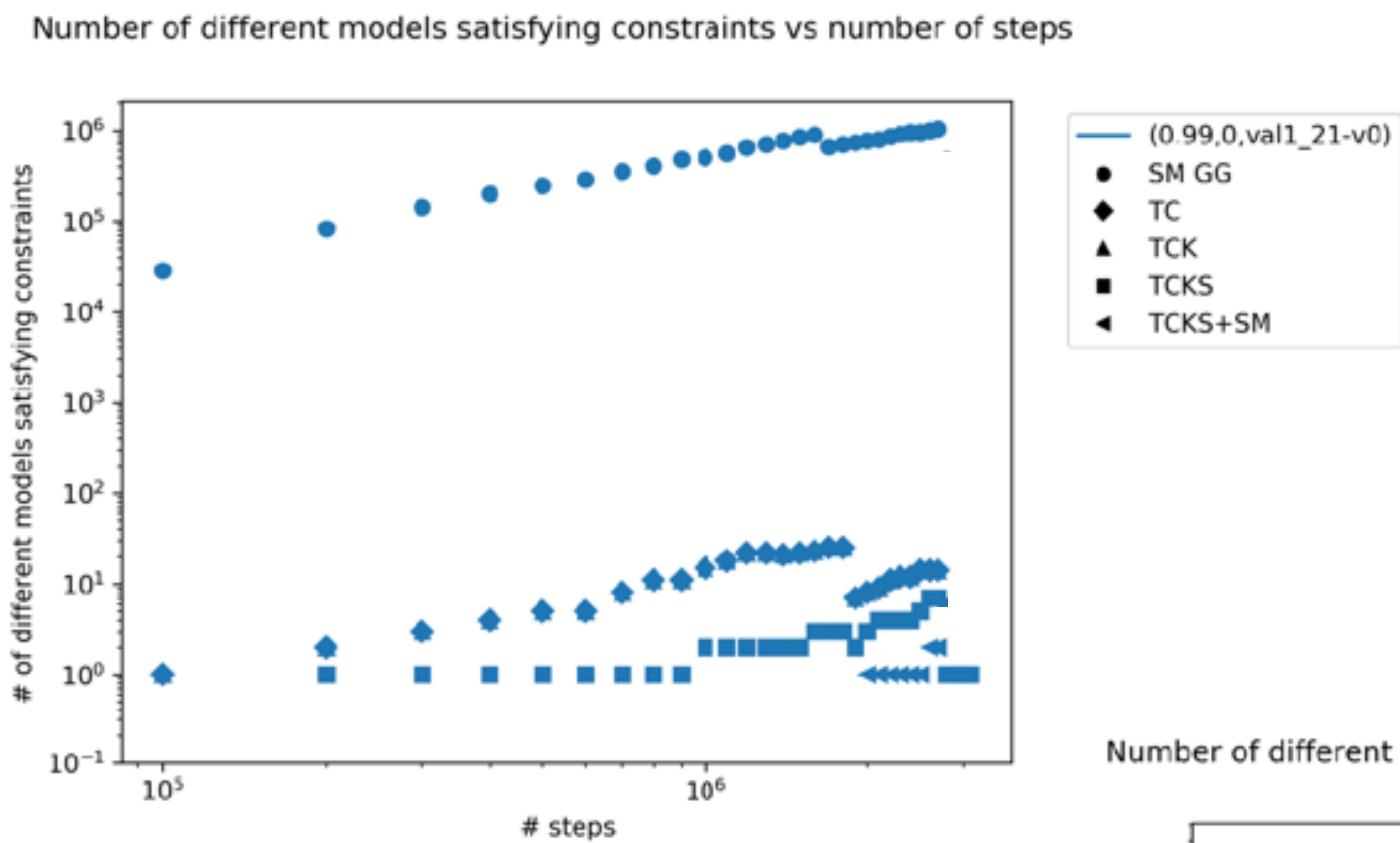
- Parameters:
  - 16 or 32 workers (1 CPU, 16-32 threads, 2.6GHz)
  - Training time of the order few hours to a day
  - Neural network for value and policy evaluation:
    Feed-forward NN with 2 hidden Softmax layers with 200 nodes
  - Initial state: Empty stack
  - Maximal steps per episode: 10,000 - 250,000
  - 10 evaluation runs every 100,000 steps

Preliminary results - Finding models Approach 1

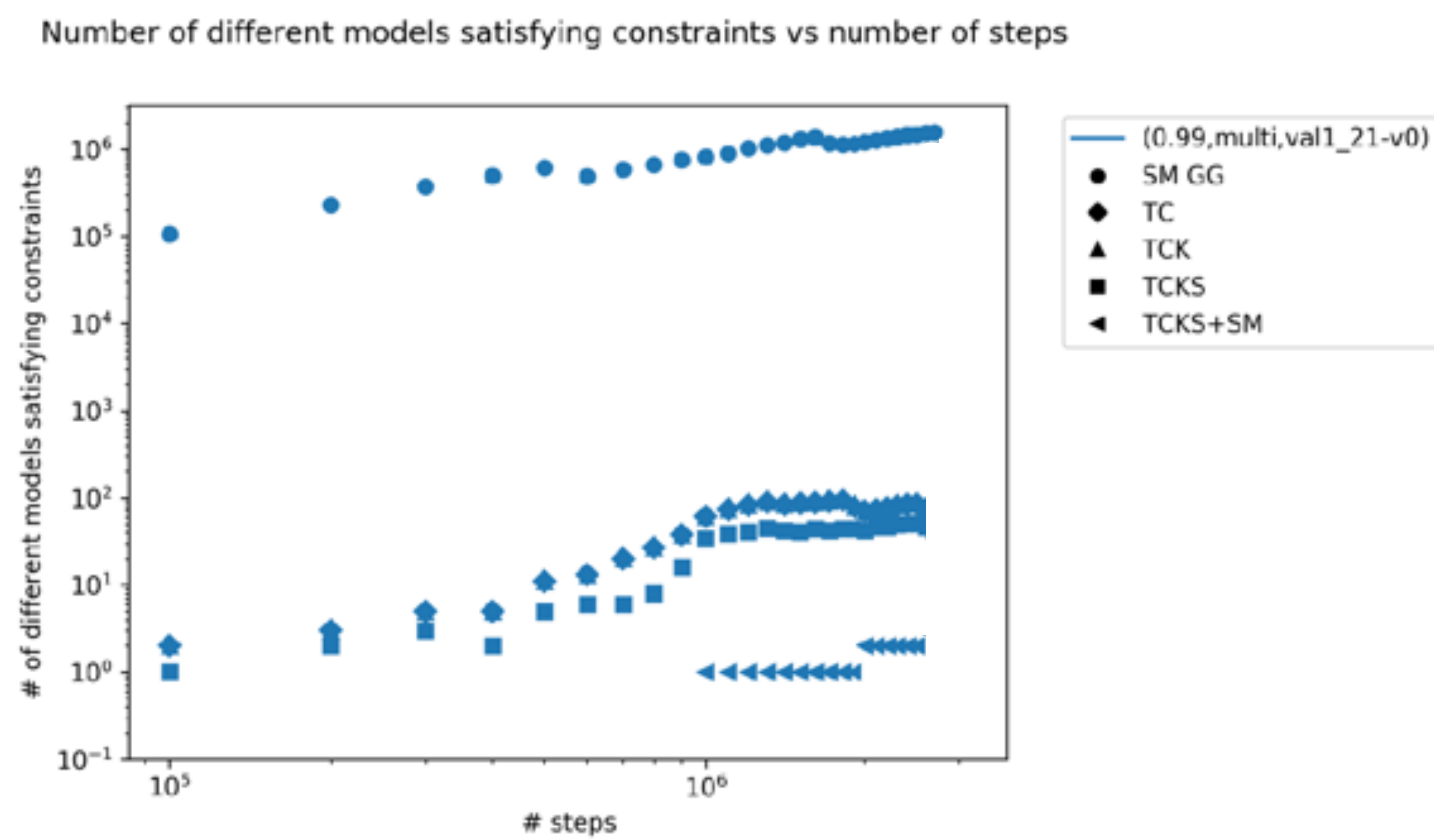




Preliminary results - Finding models Approach 1

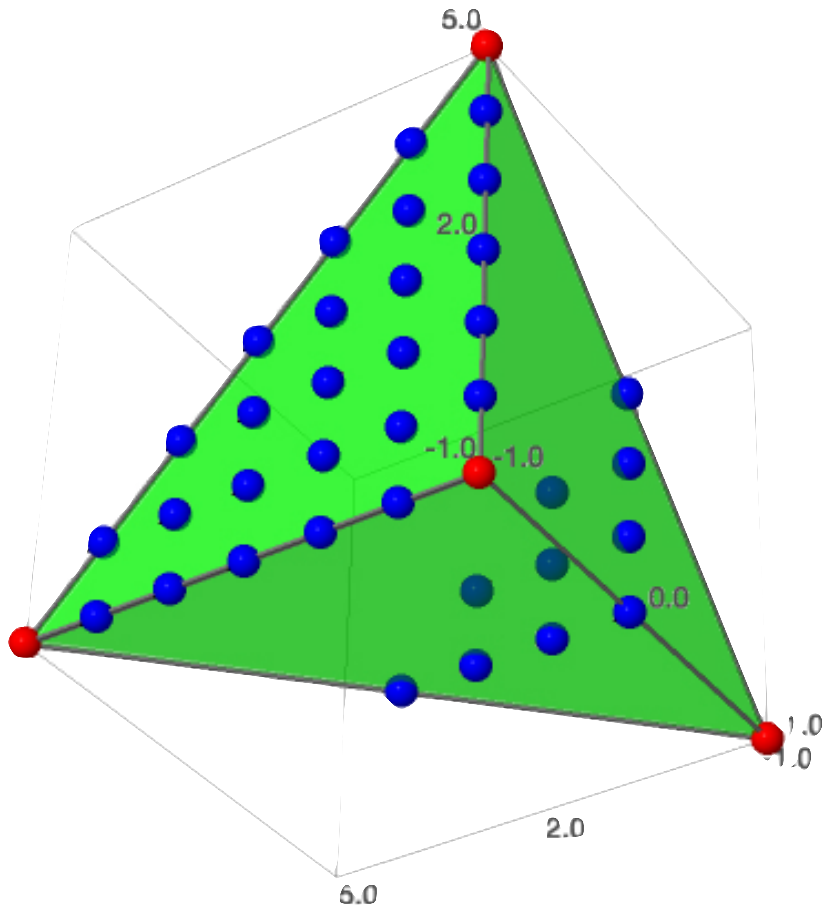


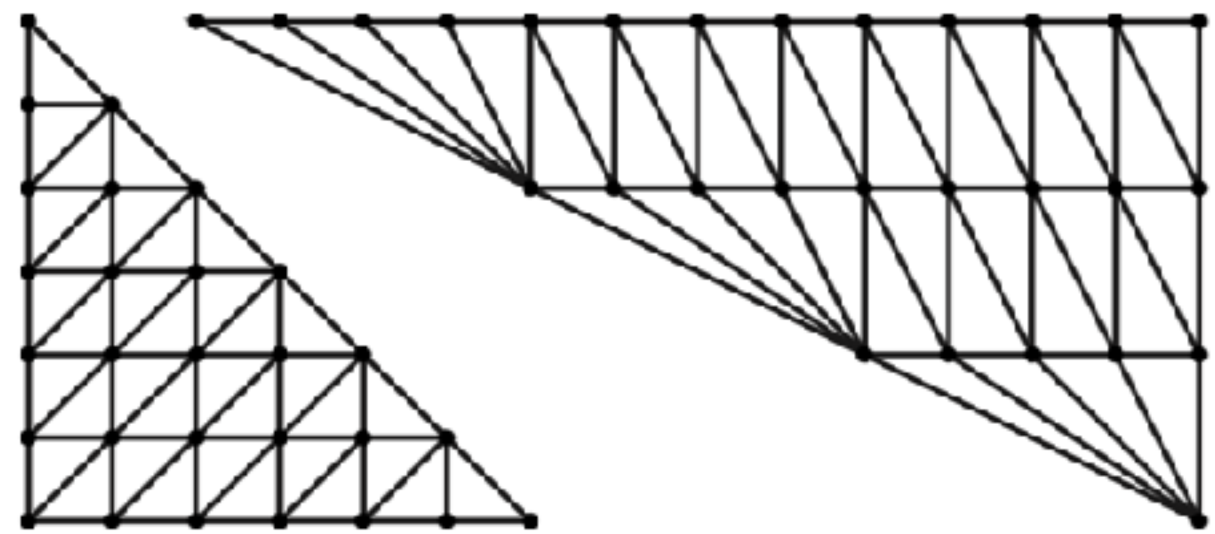
1.) Check consistency
2.) Check particle physics



multitask:

- 16 workers consistency
- 16 workers particle physics





Sen Limits in F-Theory

F-theory Sen limits

- F-Theory is strongly coupled IIB [Vafa '96]
- Coupling encoded in complex structure of auxiliary torus ($\Rightarrow$ 12-dimensional theory)
- Starting point: CY 1-fold in $\mathbb{P}_{2,3,1}[6]$
$$y^2 = x^3 + fxz^4 + gz^6\,, \quad \Delta = 4f^3 + 27g^2$$
- CY N-folds: turn (f, g) into sections of the base
- Singularities in codimension 1 correspond to gauge theories, these might or might not have a weakly coupled IIB limit

F-theory Sen limits

- Blowups will force (resolved) singularities in elliptic fibration
  $$y^2 = x^3 + fxz^4 + gz^6\,, \quad \Delta = 4f^3 + 27g^2$$
- Singularities classified by Kodaira

F	a	b	c	Sing.	G	
I_0	≥ 0	≥ 0	0	none	none	Sen
I_n	0	0	$n \geq 2$	A_{n-1}	$SU(n)$ or $Sp(\lfloor n/2 \rfloor)$	
II	≥ 1	1	2	none	none	no Sen
III	1	≥ 2	3	A_1	$SU(2)$	
IV	≥ 2	2	4	A_2	$SU(3)$ or $SU(2)$	
I_0^*	≥ 2	≥ 3	6	D_4	$SO(8)$ or $SO(7)$ or G_2	Sen
I_n^*	2	3	$n \geq 7$	D_{n-2}	$SO(2n-4)$ or $SO(2n-5)$	
IV^*	≥ 3	4	8	E_6	E_6 or F_4	no Sen
III^*	3	≥ 5	9	E_7	E_7	
II^*	≥ 4	5	10	E_8	E_8	

- If blowup enforces these (resolved) singularities: no Sen (i.e. weak coupling) limit

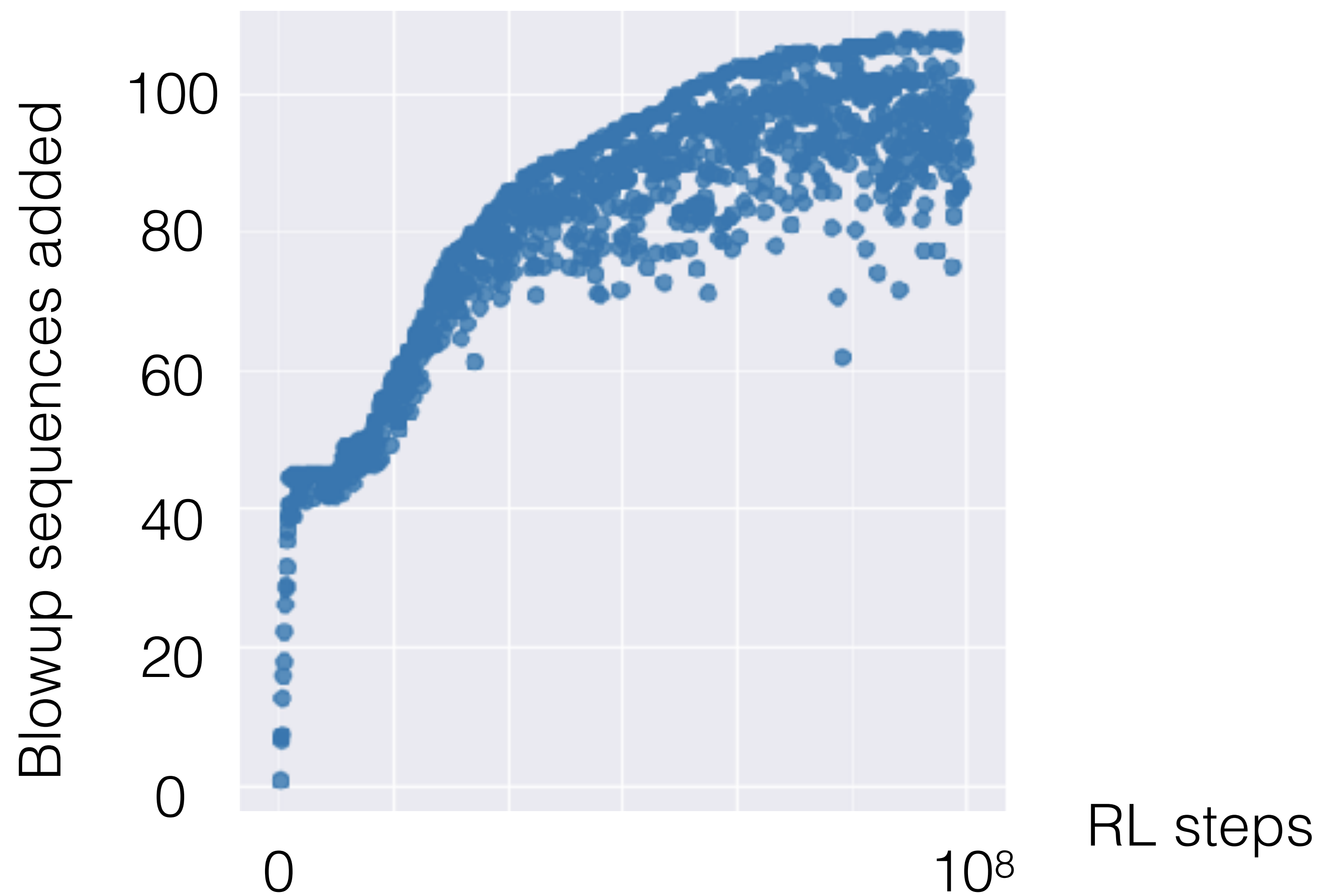
F-theory Sen limits

- Why this setup?
  - Huge ensemble of geometries (10^{755}) [Halverson, Long, Sung ’17]
  - Probability for Sen limit < $1{:}10^{-400}$
  - Still leaves a huge “anomaly” of Sen limit models
  - Geometries simple to construct (Toric blowups/blowdowns)
  - Interesting to know how “non-perturbative” string theory generically is
  - Result can have implications for phenomenology, e.g. dark matter [Halverson, Nelson, FR ’16; Halverson, Nelson, FR, Salinas ’18]

F-theory Sen limits

- Start from blown-down geometries that are known to have a Sen limit
  - Add sequence of blowups over faces or edges
  - If you can add a sequence without spoiling Sen limit, you can also add all sub-sequences
  - Two polytopes dominate the 10^{755} models $\Rightarrow$ focus on those
  - To get lower bound on number of Sen models find:
    - … as many simultaneous edge/face blowups as possible
    - … as large blowup sequences as possible

Preliminary results - Sen Blowups



- 108 blowup sequences can be added simultaneously
- Each corresponds to a single blowup, each can be done independently
- $\#\text{Sen states} < 2^{108} \simeq 10^{32}$

Conclusion

- RL well suited for search & explore in the string landscape
- Very versatile applications to string theory:
  - String models in Type II intersecting brane models on toroidal orientifolds
  - Sen limit in F-Theory geometries

Thank you for your attention!