



**Swarat Chaudhuri**

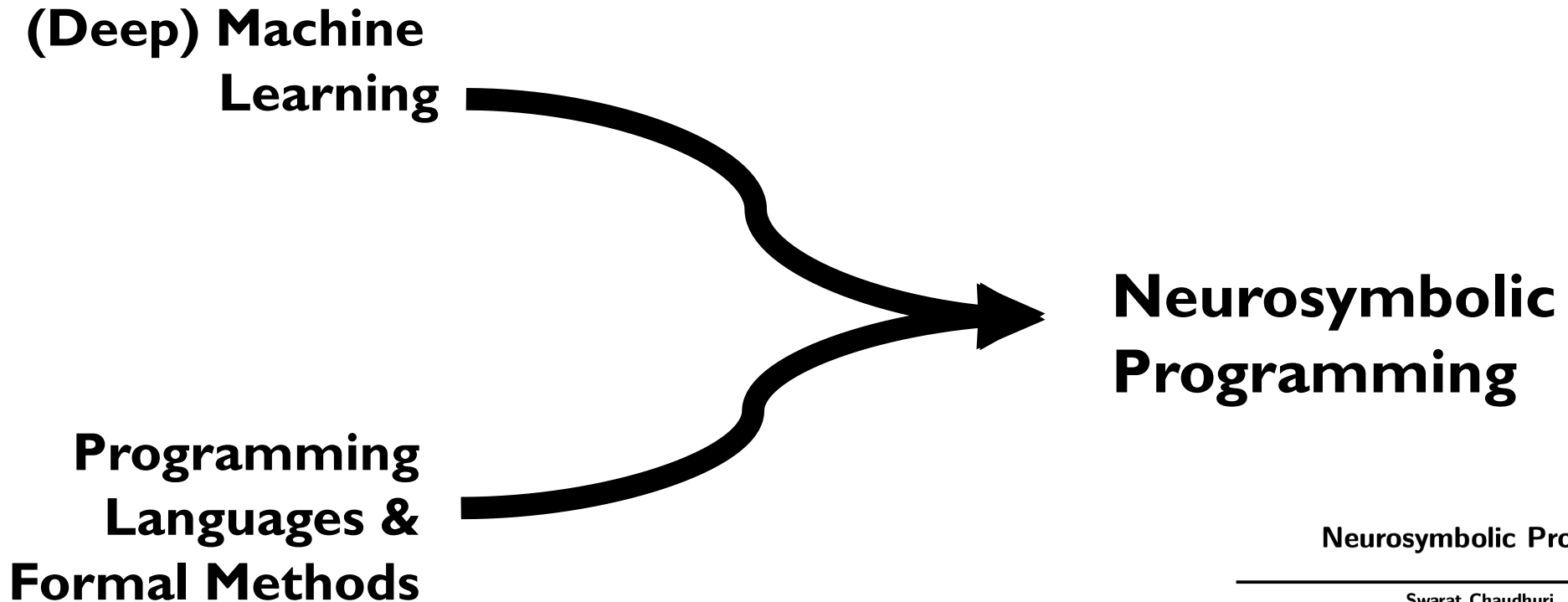
**The University of Texas at Austin**

# **Neurosymbolic Programming for Science**



\* Based on work with Jennifer Sun, Atharva Sehgal, Ameesh Shah, Eric Zhan, Ann Kennedy, Megan Tjandrasuwita, and Yisong Yue

# Neurosymbolic Programming



**Neurosymbolic Programming.** Chaudhuri, Ellis, Polozov, Singh, Solar-Lezama, Yue.  
Foundations and Trends in Programming Languages, 2021.

## Neurosymbolic Programming

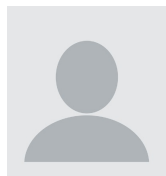
<b>Swarat Chaudhuri</b> UT Austin swarat@cs.utexas.edu	<b>Kevin Ellis</b> Cornell kellis@cornell.edu
--	---

<b>Oleksandr Polozov</b> Google polozov@google.com	<b>Rishabh Singh</b> Google rising@google.com
--	---

<b>Armando Solar-Lezama</b> MIT asolar@csail.mit.edu	<b>Yisong Yue</b> Caltech yyue@caltech.edu
--	--



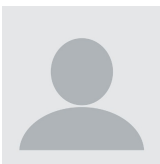
Yeming Wen



Amal Babu



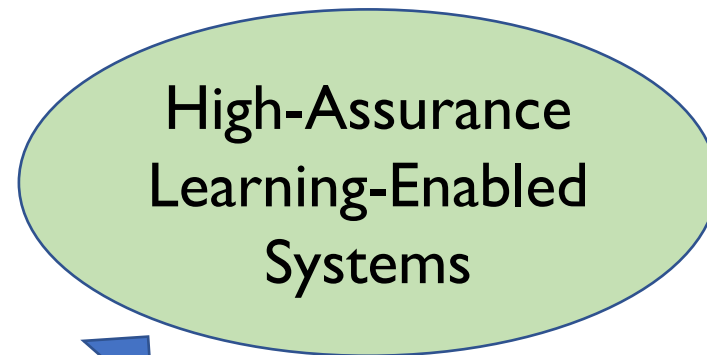
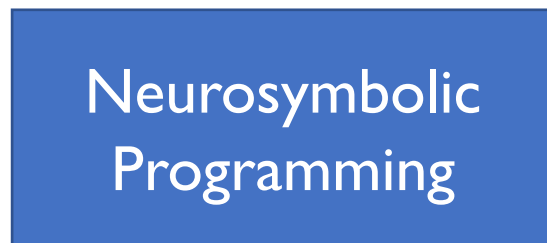
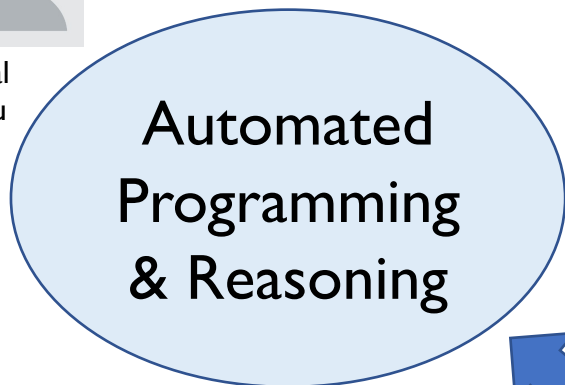
Meghana Sistla



Thomas Logan



Amitayush Thakur



Josh Hoffman



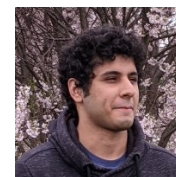
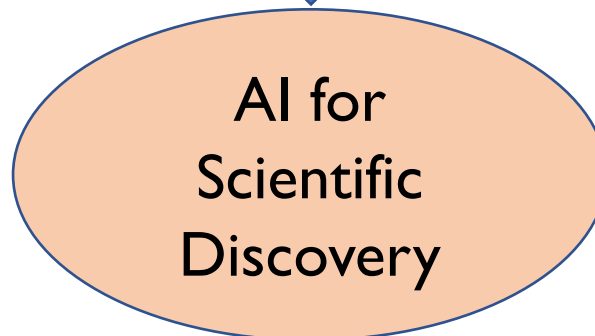
Dweep Trivedi



Greg Anderson



Chenxi Yang



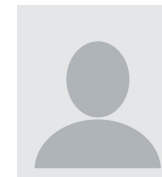
Atharva Sehgal



Sam Anklesaria

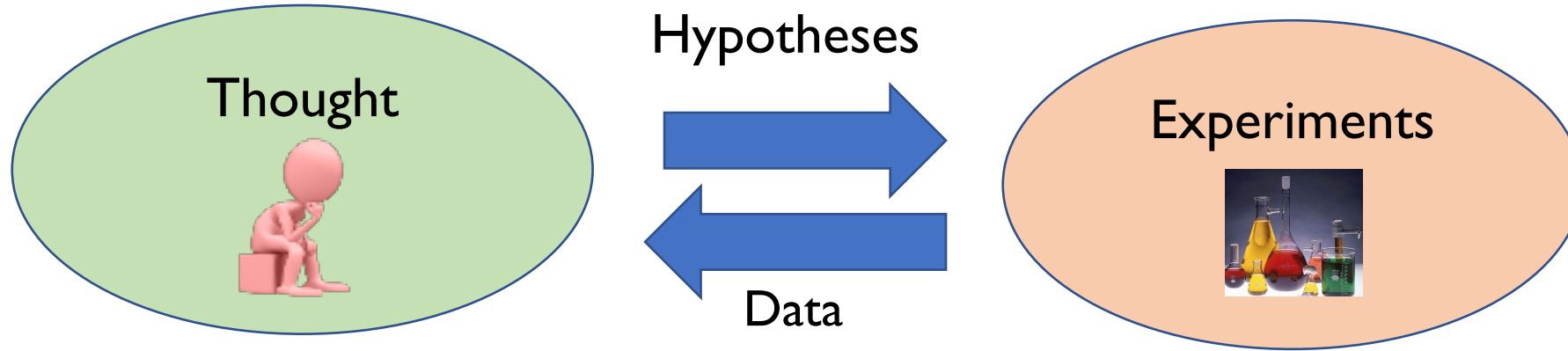


Chenxi Yang



Arya Grayeli

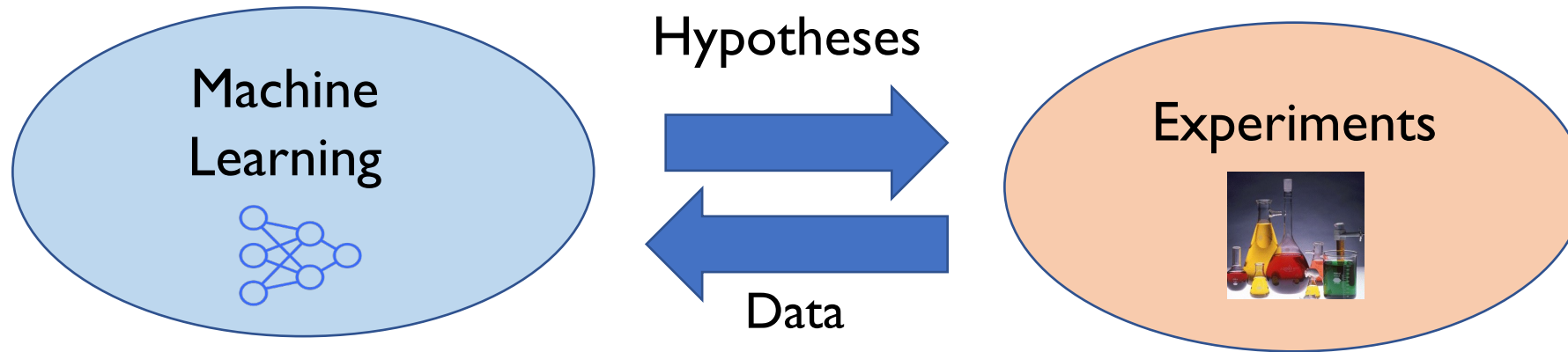




Throughout history, science has required

(i) data, and (ii) human insights to make sense of the data.

# AI for science



NEWS | 20 February 2020

## Powerful antibiotics discovered using AI

*A.I. Predicts the Shape of Nearly Every Protein Known to Science*

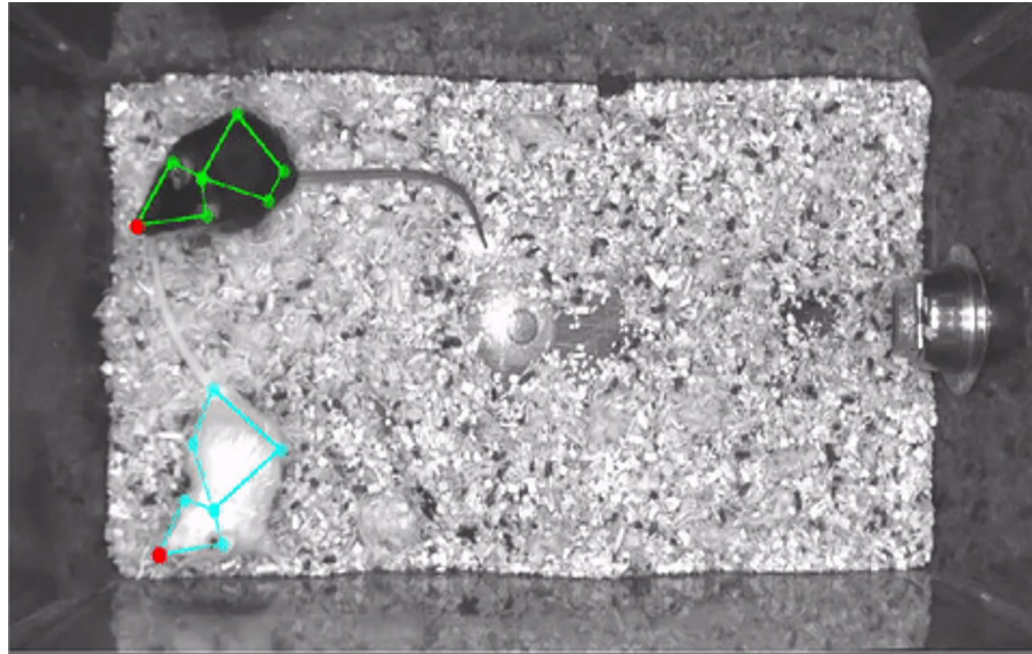
AI AND UNIVERSE

## The AI behind getting the first-ever picture of a 'black hole'

**A celebrated AI has learned a new trick: How to do chemistry**

by Marc Zimmer, The Conversation

# AI for behavioral neuroscience

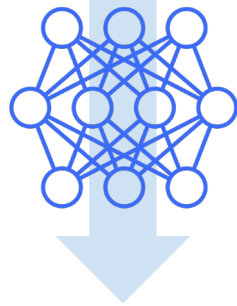


Mouse Action Recognition System [Segalin et al., 2021]

**Data:** videos depicting animal behavior

# Challenges

## I. Interpretability rather than black-box prediction



How is gait stable vs. unstable?

# Challenges

## 2. Labels can be hard to get

> 200 million image-text pairs



## DALL-E2

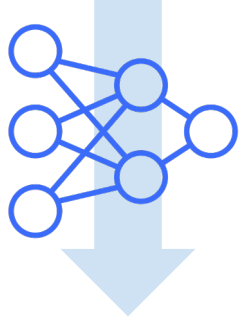
> 300 billion words



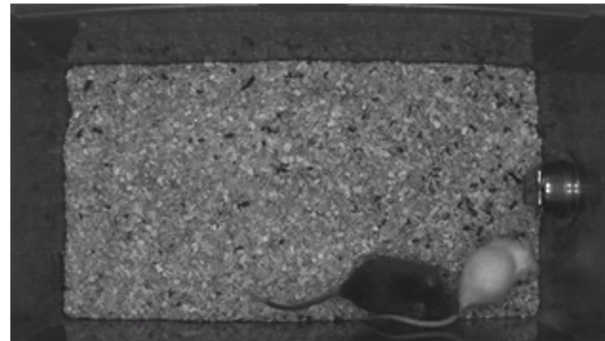
# ChatGPT

# Challenges

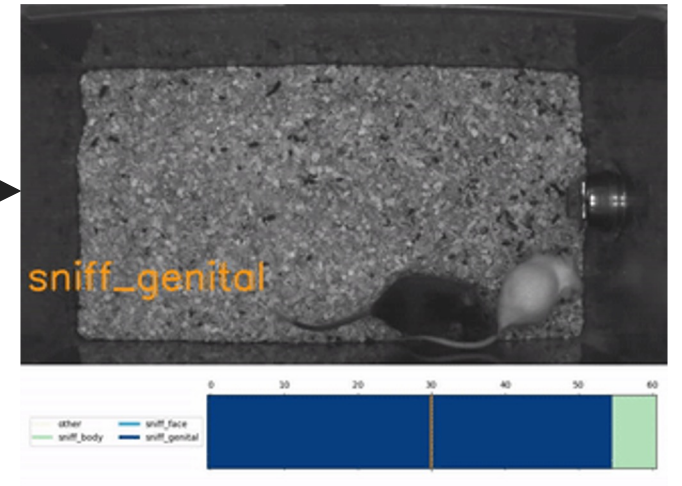
## 2. Labels can be hard to get



Experiment-specific  
behaviors



Scientists



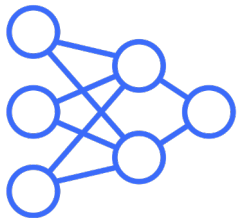
**$10^4 \sim 10^5$  of frames for training!**

**100 expert hours to annotate one day of recording**

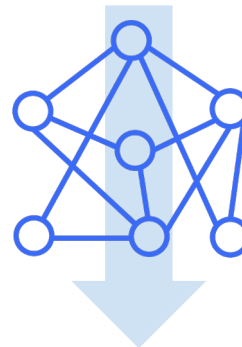
# Challenges

## 3. Labels can even be unknown

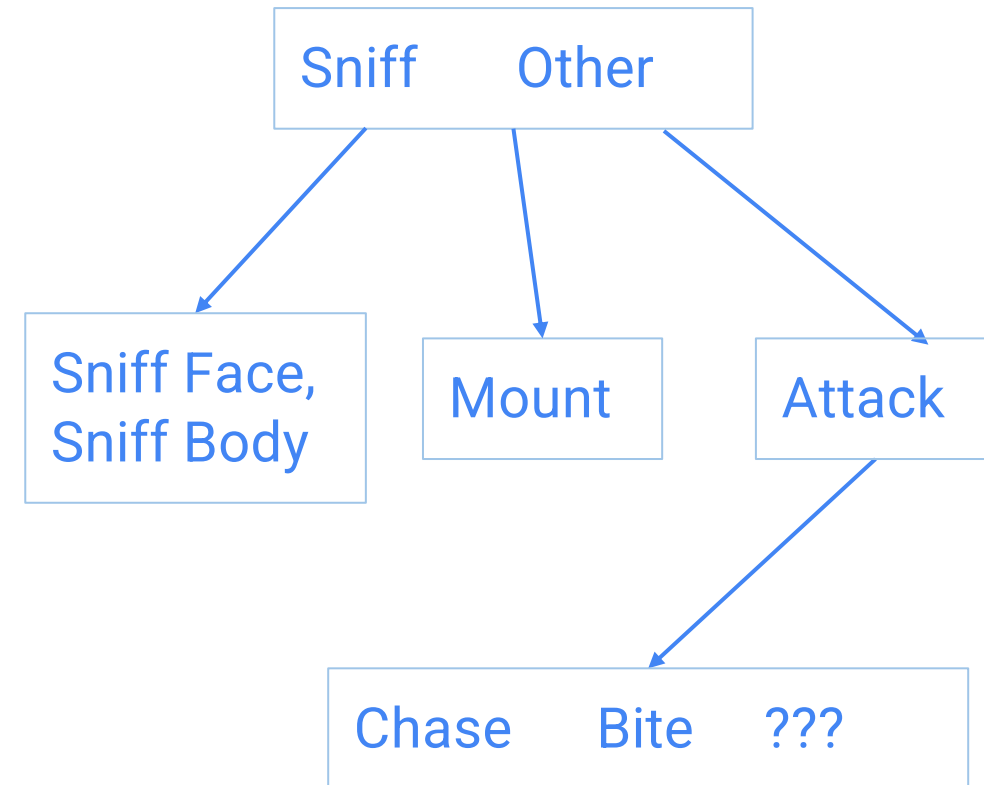
Lab A



Lab B



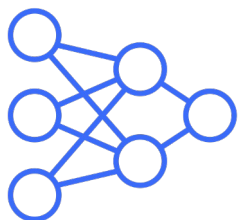
???



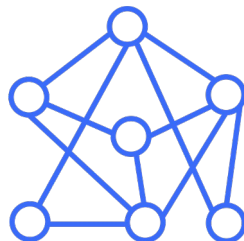
# Challenges

## 4. Distribution shifts

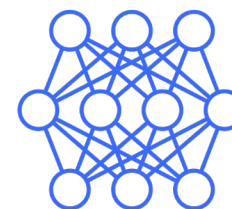
Lab A



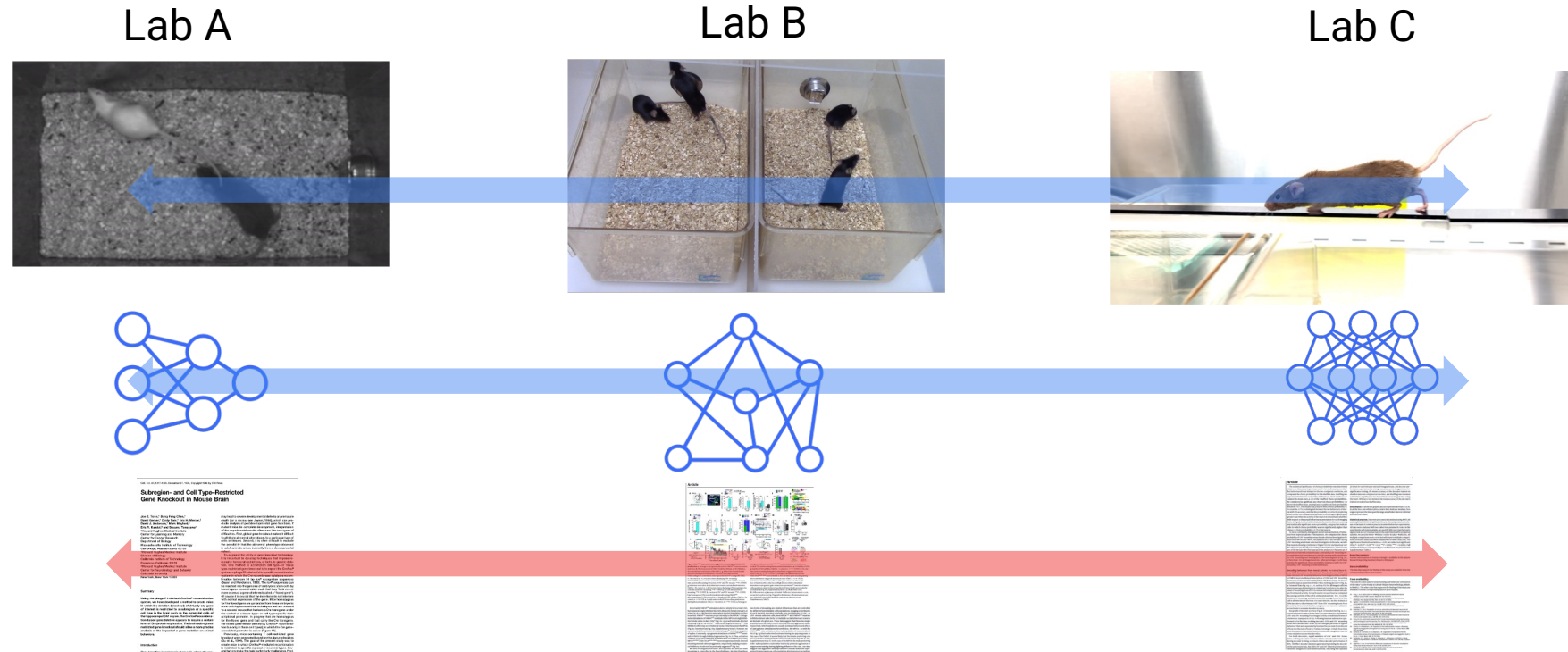
Lab B



Lab C

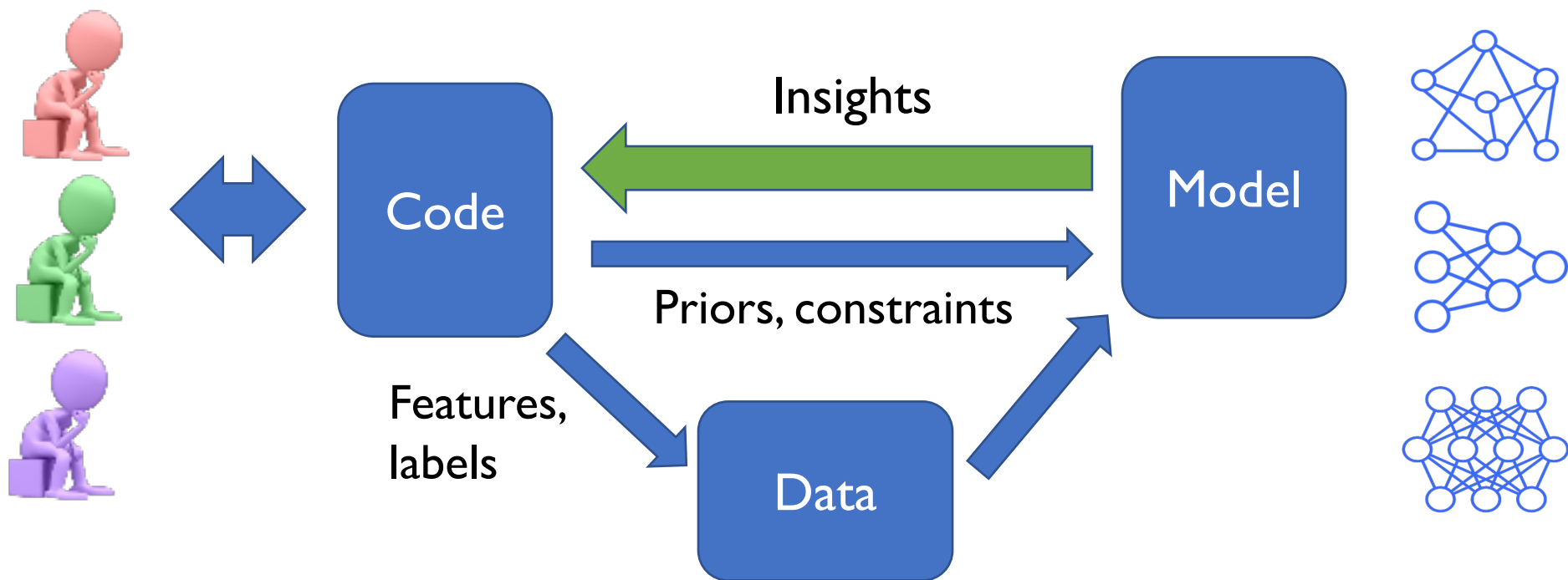


# Science needs systematic mechanisms for...



- (i) interpreting discovered insights
- (ii) incorporating domain knowledge to reduce need for data
- (iii) reusing code and data across labs

# Data-driven discovery as programming

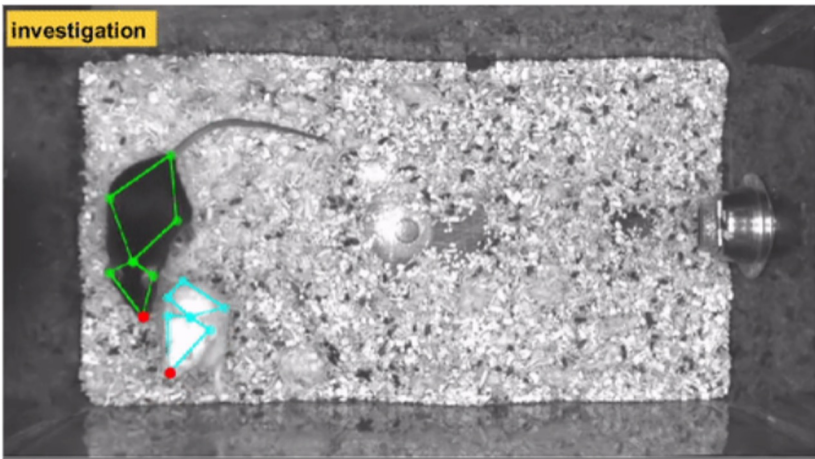


- (i) Neurosymbolic programs
- (ii) Neurosymbolic learning algorithms

**Neurosymbolic Programming for Science.** Sun\*, Tjandrasuwita\*, Sehgal\*, Solar-Lezama, Chaudhuri, Yue, Costilla-Reyes. NeurIPS AI4Science workshop 2022.

## **A. Neurosymbolic Programs**

# Neurosymbolic Program: Example



IF  (distance between noses) < A AND

 (facing angle) < B

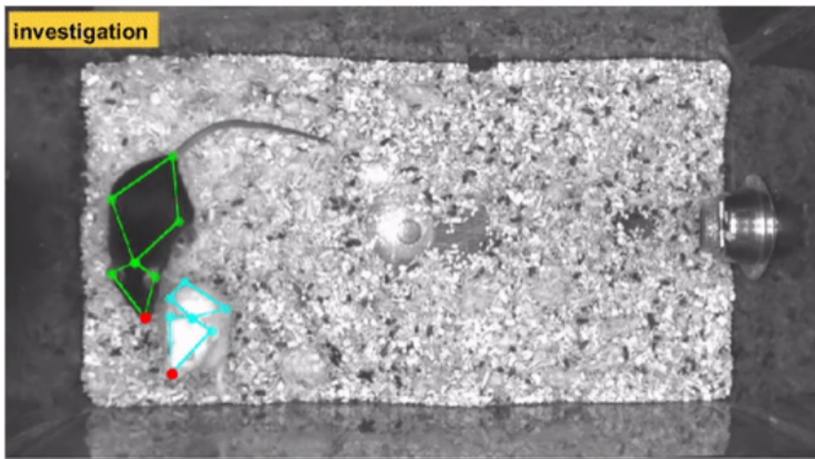
THEN **investigation** IF

 (acceleration of mouse 1) > C

ELSE **investigation** IF

 (distance from nose 1 to centroid 2) < D

# Neurosymbolic Program: Example



IF  (distance between noses)  $< A$  AND

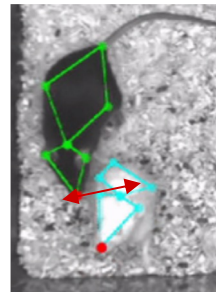
 (facing angle)  $< B$

THEN **investigation** IF

 (acceleration of mouse 1)  $> C$

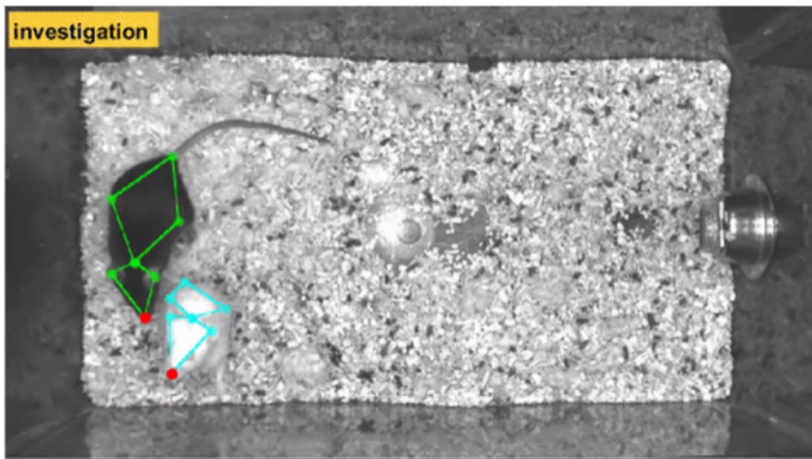
ELSE **investigation** IF

 (distance from nose 1 to centroid 2)  $< D$



Features defined by  
experts

# Neurosymbolic Program: Example



IF  (distance between noses) < A AND

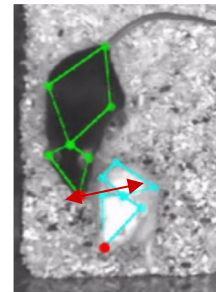
 (facing angle) < B

THEN **investigation** IF

 (acceleration of mouse 1) > C

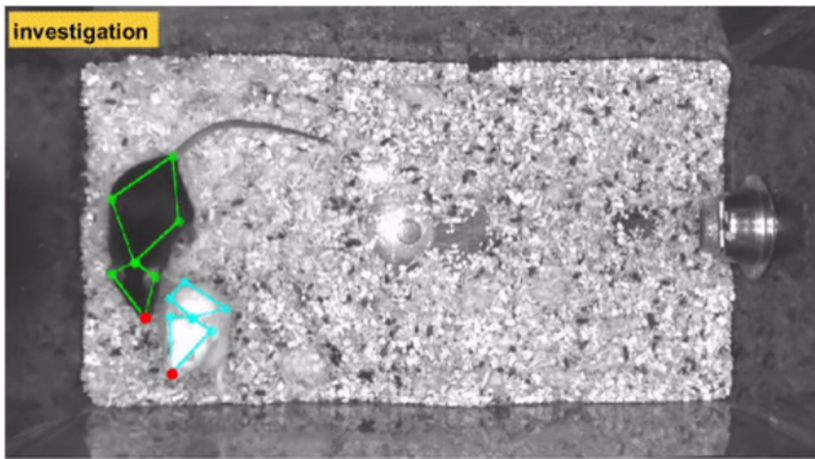
ELSE **investigation** IF

 (distance from nose 1 to centroid 2) < D



Structure & parameters  
learned from data

# Neurosymbolic Program: Example



IF  (distance between noses) < A AND

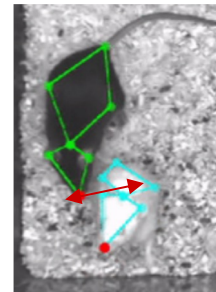
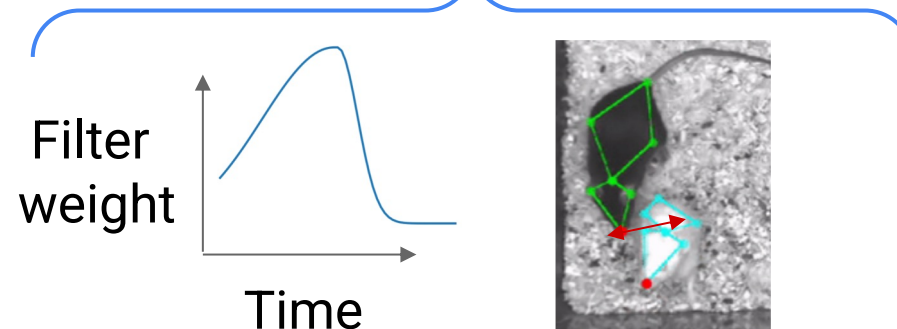
 (facing angle) < B

THEN **investigation** IF

 (acceleration of mouse 1) > C

ELSE **investigation** IF

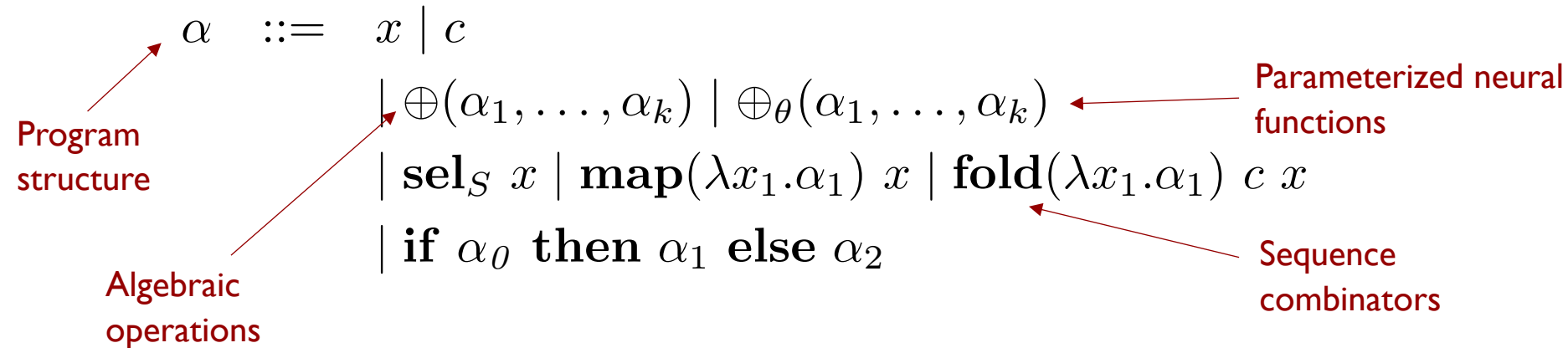
 (distance from nose 1 to centroid 2) < D



## **B. Neurosymbolic Learning Algorithms**

# Domain-Specific Language (DSL): “A Family of Programs”

Program syntax defined as a grammar:



Type system tracking, for example, vector and matrix dimensions

DSL is differentiable, so you can train an NN in the context of a larger program

- For example, differentiable interpretation of if-then-else statements

# Neurosymbolic Program Synthesis

$$\begin{aligned} \alpha \quad ::= \quad & x \mid c \\ & \mid \oplus(\alpha_1, \dots, \alpha_k) \mid \oplus_{\theta}(\alpha_1, \dots, \alpha_k) \\ & \mid \text{sel}_S x \mid \text{map}(\lambda x_1. \alpha_1) x \mid \text{fold}(\lambda x_1. \alpha_1) c x \\ & \mid \text{if } \alpha_0 \text{ then } \alpha_1 \text{ else } \alpha_2 \end{aligned}$$

**Domain Specific Language (DSL)**



**Learning Objective  
(Loss Function)**

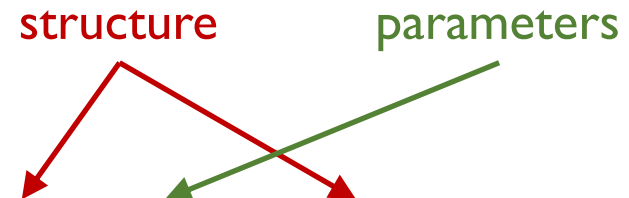


**Learning Algorithm  
(program synthesis)**



**Neurosymbolic Program  $(\alpha, \theta)$**

# Learning as Bilevel Optimization



The diagram shows the bilevel optimization problem with two arrows indicating the mapping of variables to the objective function components:

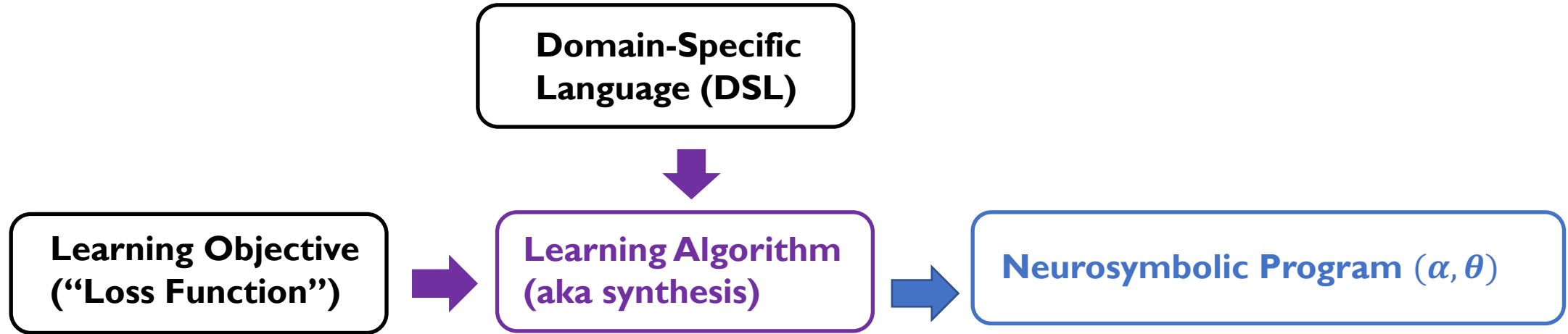
- A red arrow labeled "structure" points from the  $\alpha$  variable to the  $s(\alpha)$  term.
- A green arrow labeled "parameters" points from the  $\theta$  variable to the  $Loss(\alpha, \theta)$  term.

$$\min_{\alpha} (\min_{\theta} Loss(\alpha, \theta) + s(\alpha))$$

- $Loss(\alpha, \theta)$  quantifies fit to the dataset
- The **structural cost**  $s(\alpha)$  penalizes complex program structures.

# Learning Strategy

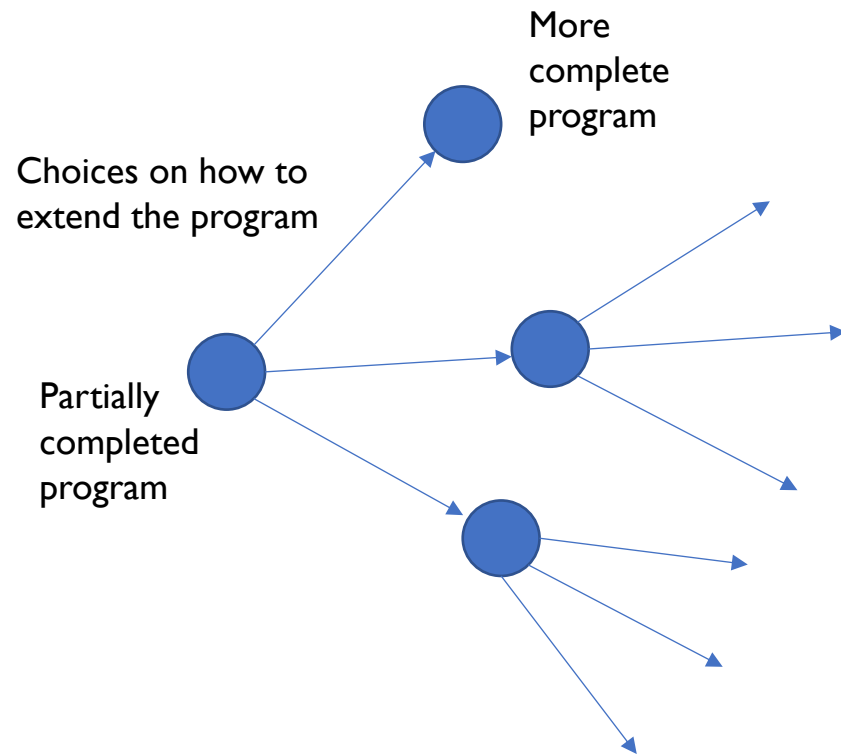
$$\min_{\alpha} (\min_{\theta} \text{Loss}(\alpha, \theta) + s(\alpha))$$



- Setting  $\alpha$  as a neural network  $\rightarrow$  standard deep learning
- Finding  $\alpha$  is analogous to neural architecture search
  - Sometimes call  $\alpha$  the “program architecture”
- Classic program synthesis focuses on  $\alpha$ , with  $\theta$  being very simple

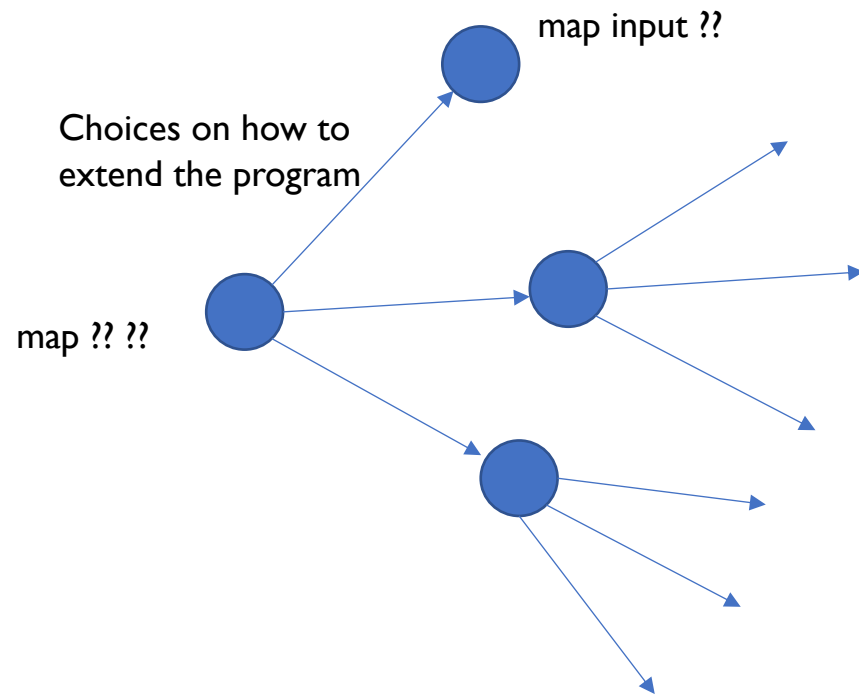
# Enumerating programs

Program enumeration is really a graph search problem



# Enumerating programs

Program enumeration is really a graph search problem



# Estimating the “Cost to Go”

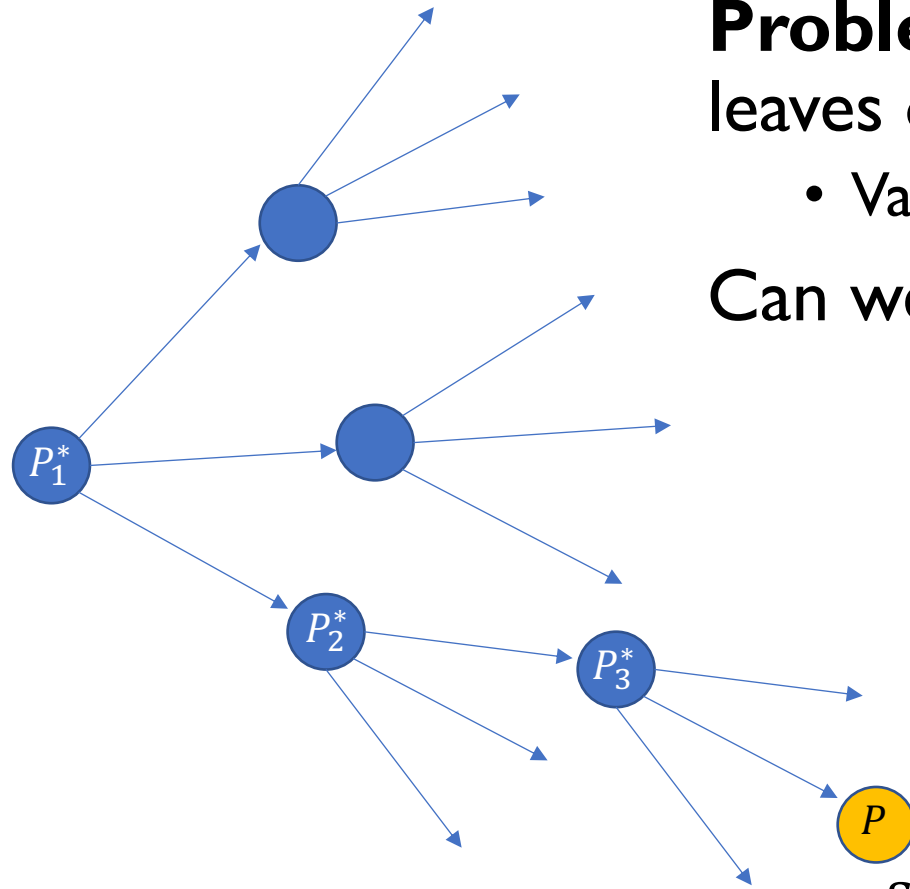
- $P^*$  = partial program (non-terminal nodes)
- $\mathbb{C}(P^*)$  = completions of  $P^*$  (reachable terminal nodes)

$$\text{Heuristic Estimate: } d(P^*) \approx \min_{P \in \mathbb{C}(P^*)} \left[ \underbrace{\Delta s(P, P^*)}_{\text{Additional Structure Cost}} + \underbrace{\min_{\theta} \text{Loss}(\alpha_P, \theta_P)}_{\text{Training Loss}} \right]$$

“Cost to Go”

- If  $d(P^*)$  is a lower bound it becomes an “admissible heuristic”

# Guiding program search



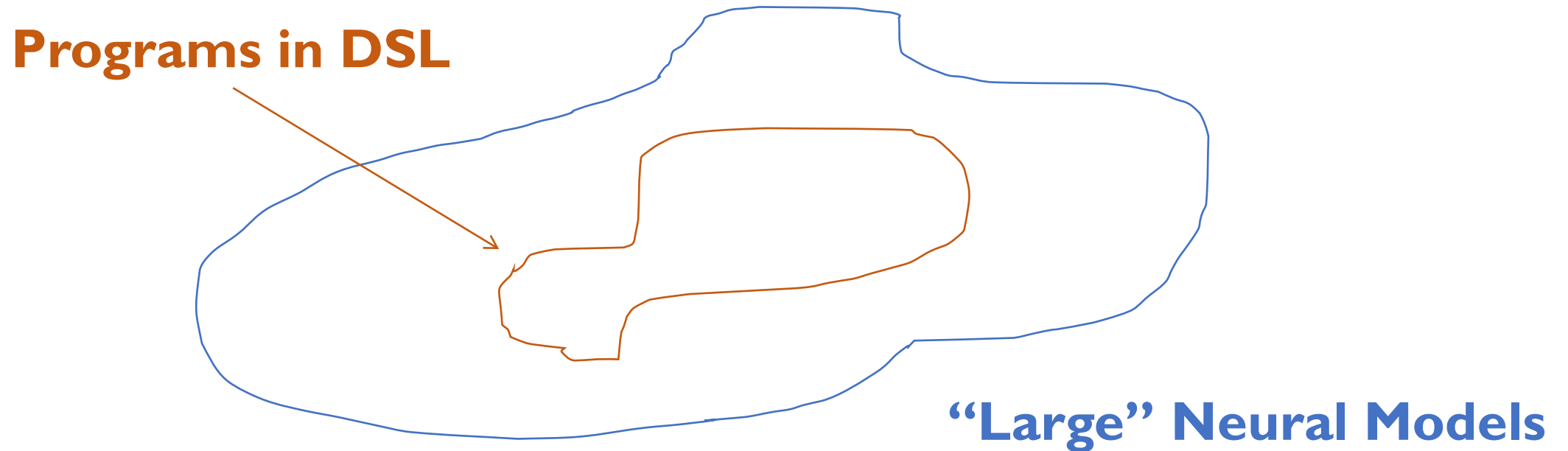
**Problem:** You only get ground truth on the leaves of the search tree

- Value for an intermediate node is only an estimate

Can we get a better estimate with deep learning?

$$s(P) + \min_{\theta} \text{Loss}(\alpha_P, \theta_P)$$

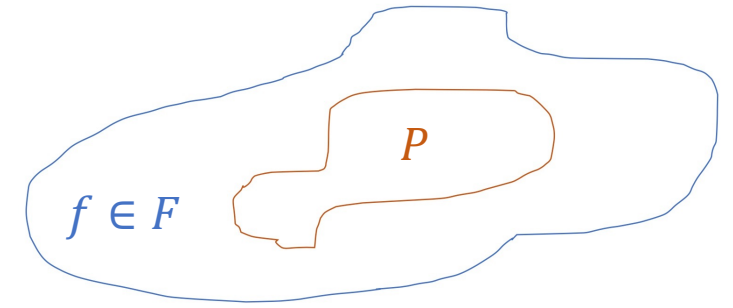
# Motivating Observation/Assumption: Functional Representational Power



## “Neural Relaxation”:

Every DSL program can be (approximately) represented by some “large” neural model.

# Implication (abstract form)



Large Neural

Slack due to approximation error or training ability

$$\forall P, \exists f \in F \text{ s.t. } d(f) \leq d(P) + \epsilon$$

From DSL

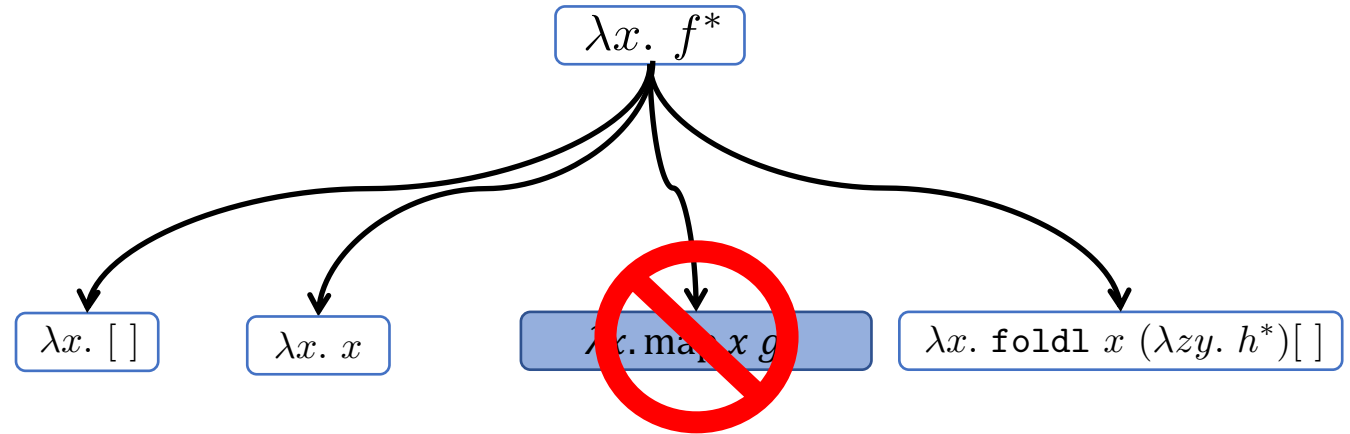
Any Cost Function

**We can train an admissible heuristic!**

**“Neural Relaxation”** Every DSL program can be (approximately) represented by some “large” neural model.

# Informed Search (e.g., A\*)

- Use  $d(P^*)$  to prune the search



**Can Prune This Branch!**

Suppose:

$$s(\lambda x. \text{map } x \, g^*) + d(\lambda x. \text{map } x \, g^*) > s(\lambda x. x) + \text{Loss}(\lambda x. x)$$

↙ Structural Cost      ↘ Training Loss

“Cost to Go” Heuristic

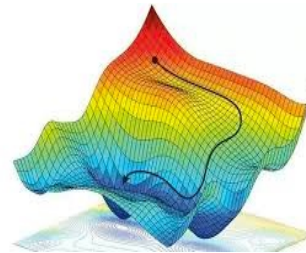
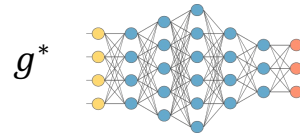
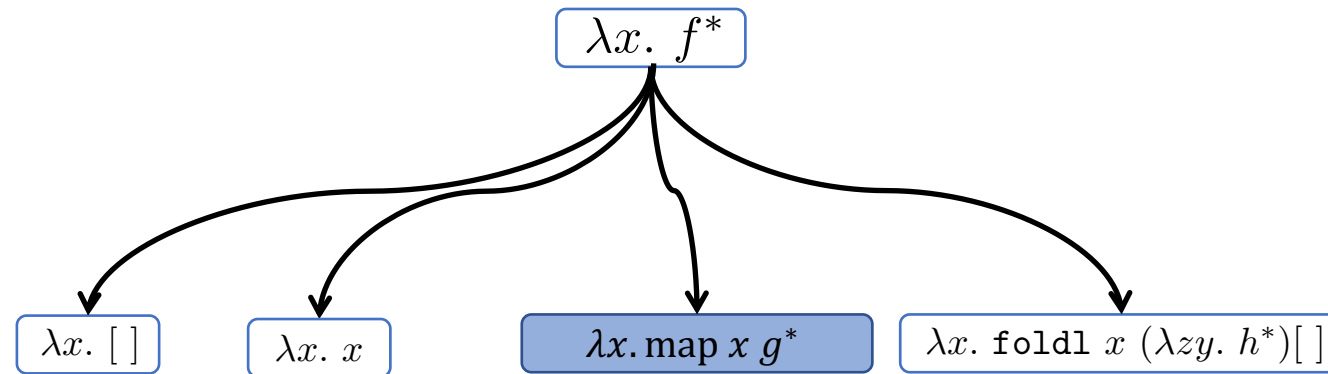
# A\* Search

- Priority queue of current leaf nodes:
  - Sorted by  $s(P^*) + d(P^*)$
- Pop off top program  $P^*$ 
  - If  $P^*$  is complete, terminate
  - Else, expand  $P^*$ , add child nodes to priority queue

Lower bounds “Cost to Go”

- **Guarantee:** if  $d(P^*)$  is **admissible**, A\* will return optimal  $P$ 
  - Tighter  $d(P^*)$  prunes more aggressively
  - Uninformed  $d(P^*)$  (e.g., always 0) → uninformed search

# NEAR: Neural Admissible Relaxations



If a large neural network cannot fit this hole, then a completion from the DSL also cannot

**Fill hole with NN**

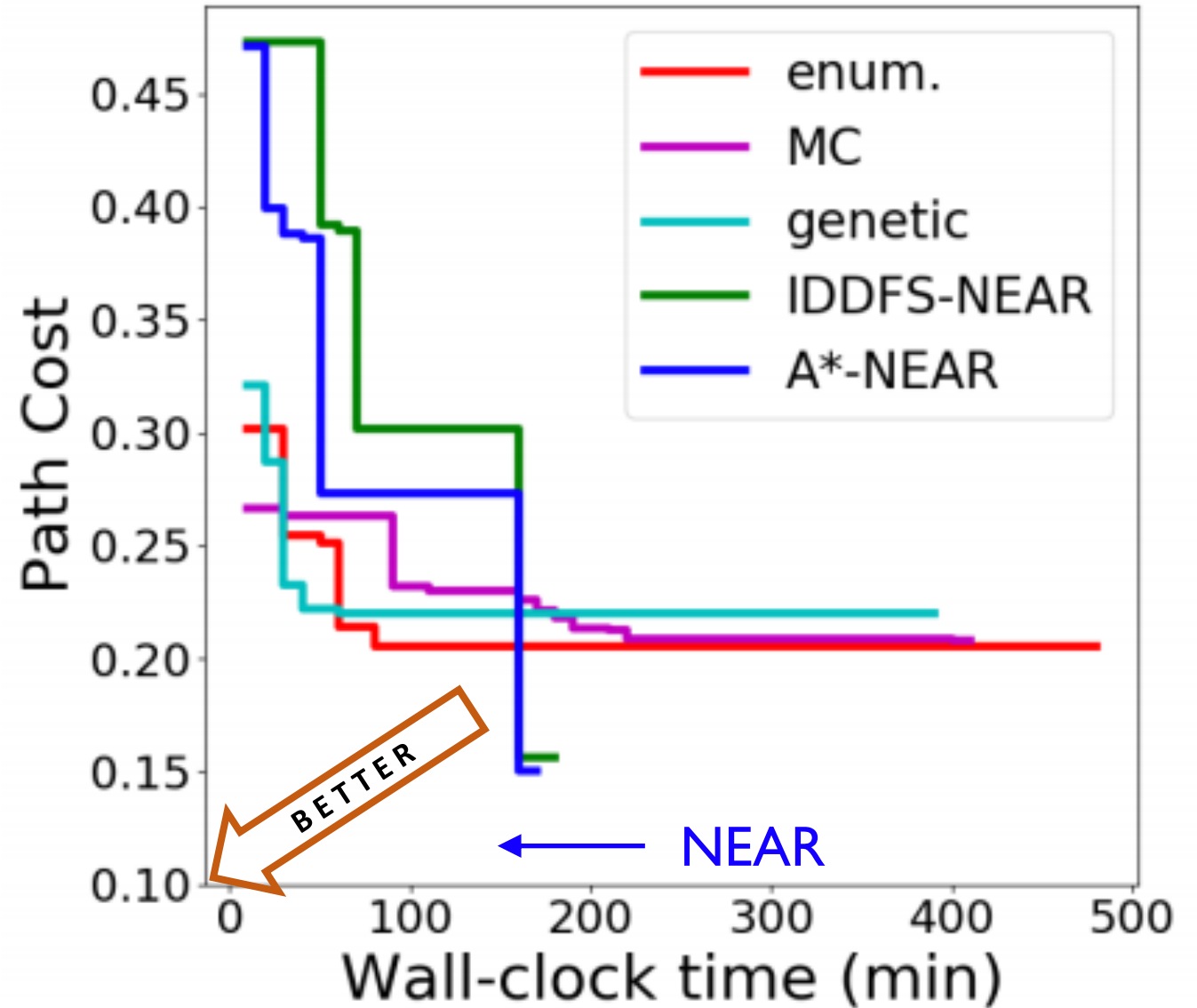
**Train parameters**

**Use training loss as admissible heuristic**

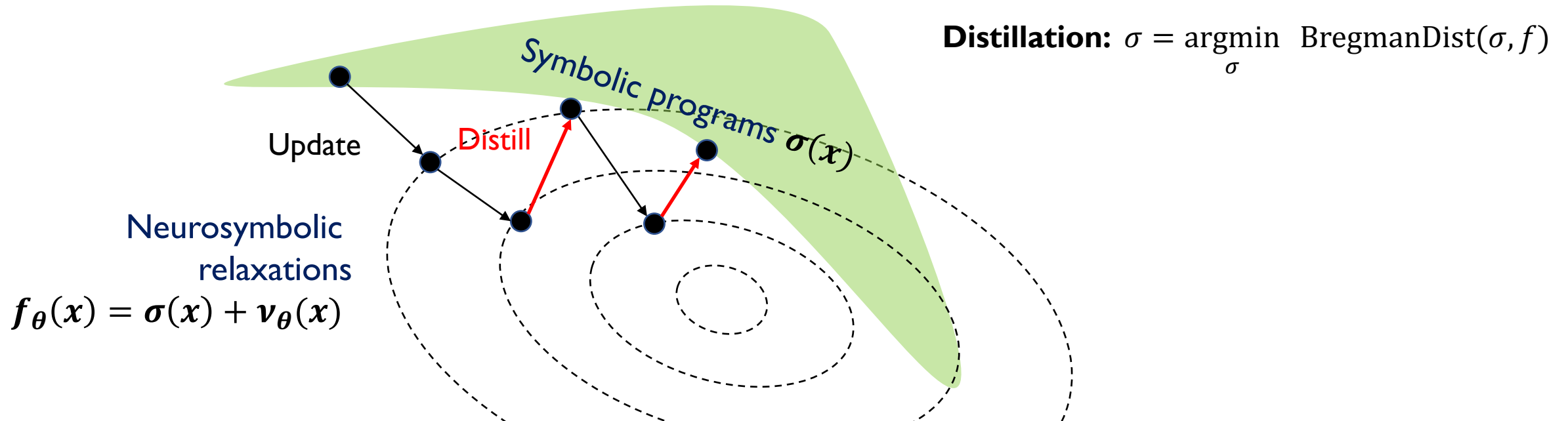
**Stop any time!**

# NEAR: Results

Order of magnitude speedup



# Other uses of relaxations



**Relax:** Add a parameterized neural component to a program

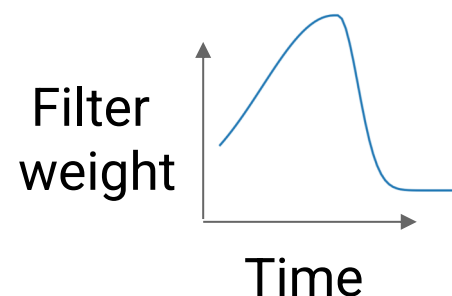
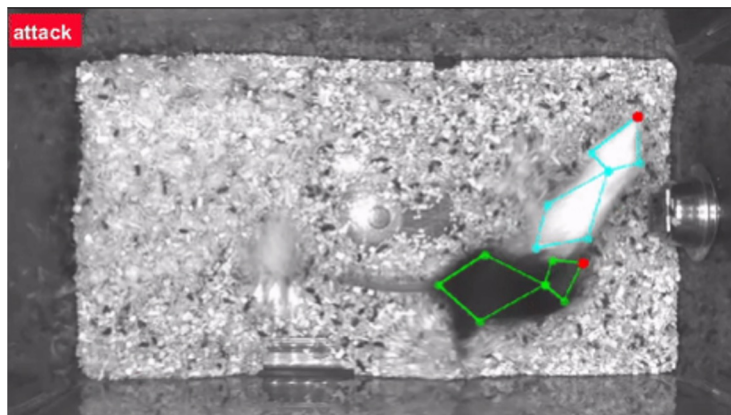
**Update:** Gradient-based update to neural component



- Approximation to gradient in program space

**Distill:** Synthesize symbolic program closest to current neurosymbolic program

# Back to behavior analysis

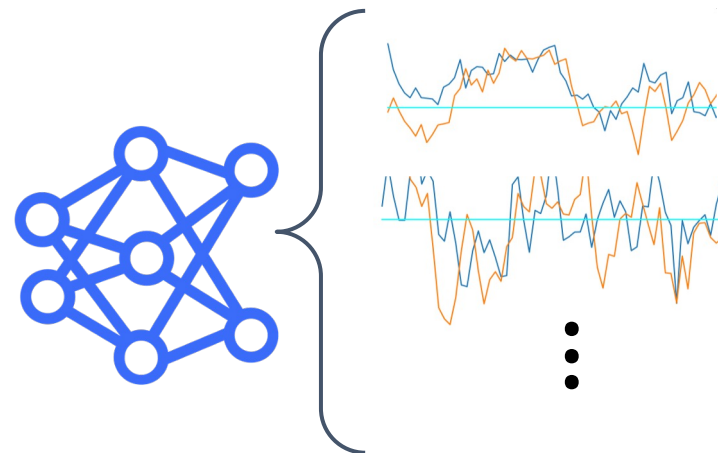
How to describe “attack” behavior?



IF  (mouse 1 & 2 acceleration) > A AND  
     (mouse 1 & 2 velocity) < B  
THEN **attack**, ELSE **not attack**

1D Conv Net

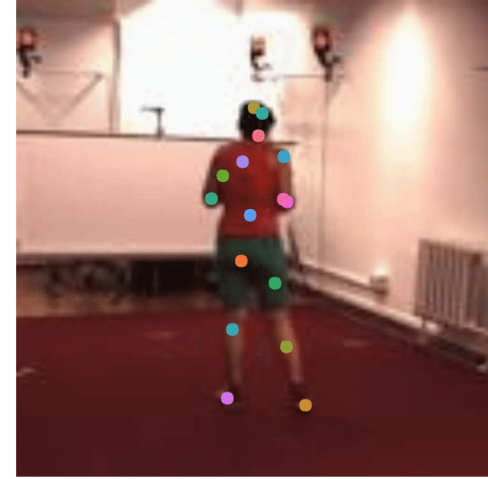
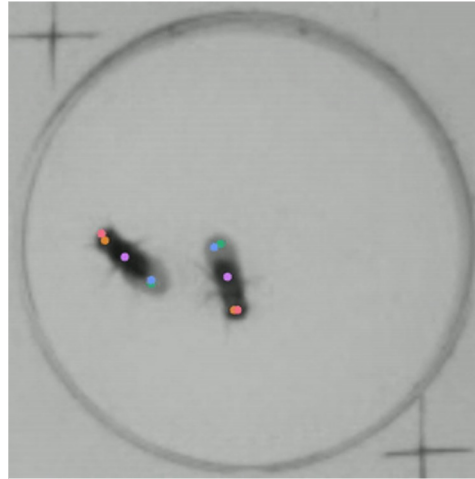
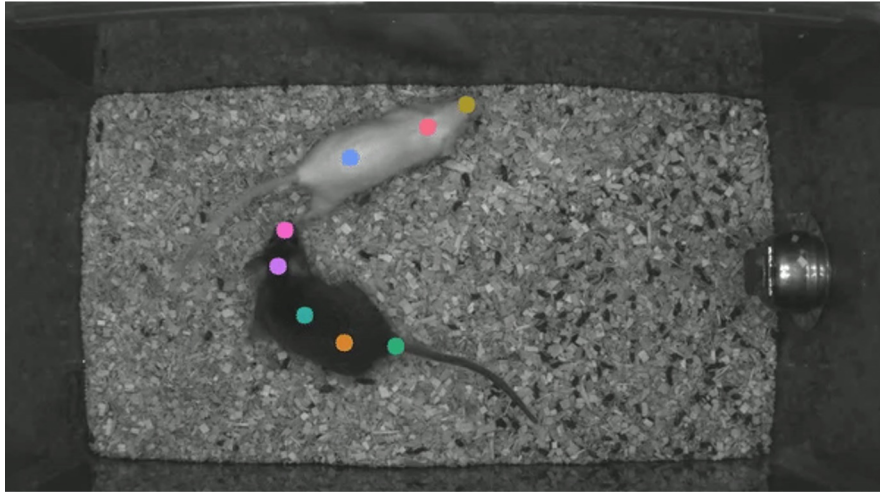
**F1: 0.86**



Learned Program

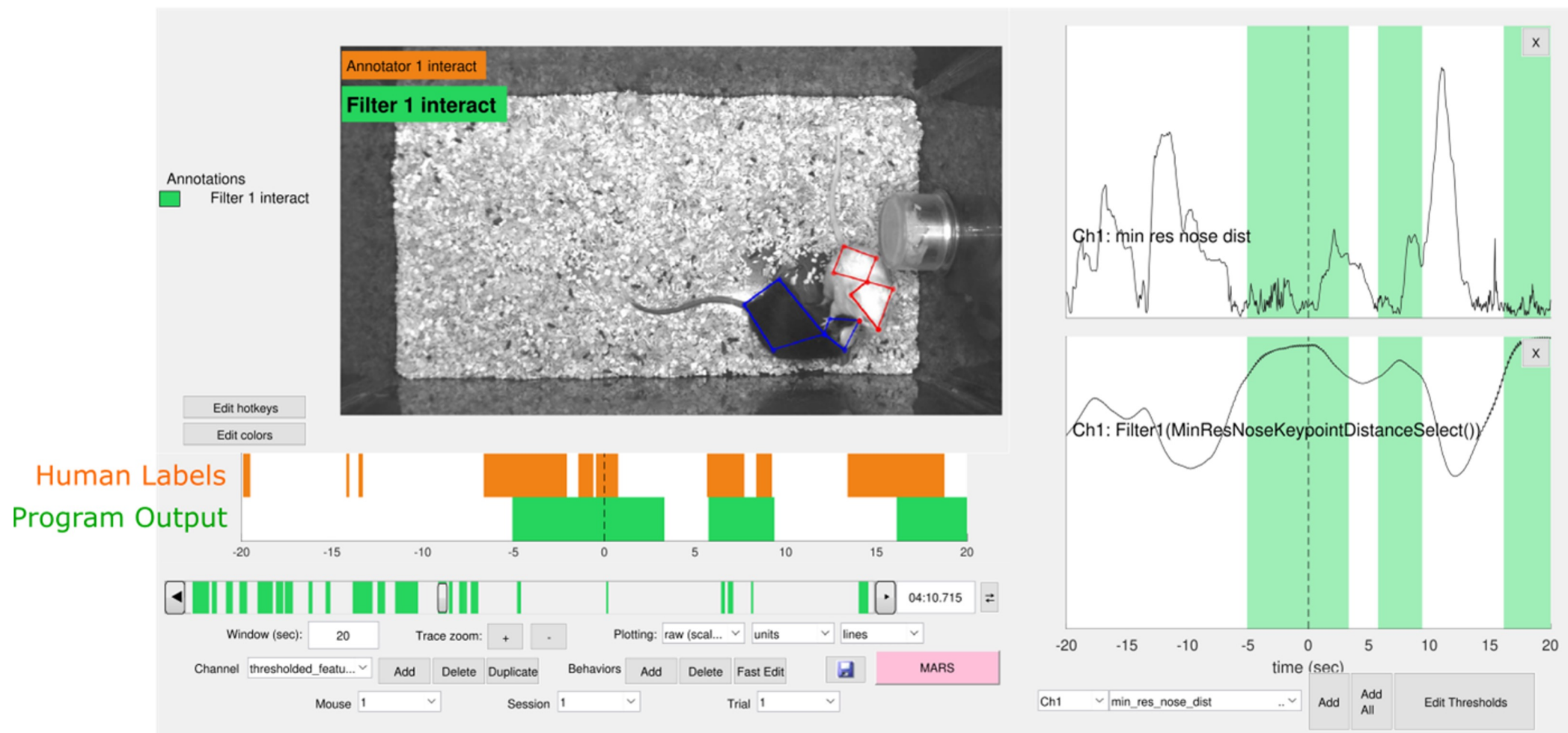
**F1: 0.84**

# Handling raw inputs

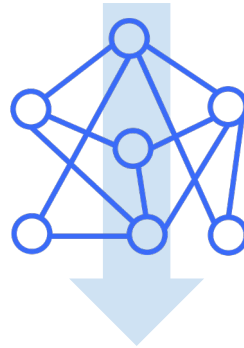


Use a complementary method (e.g., keypoints) to abstract images into symbolically interpretable features [Sun, Ryuou, et al., CVPR 2022]

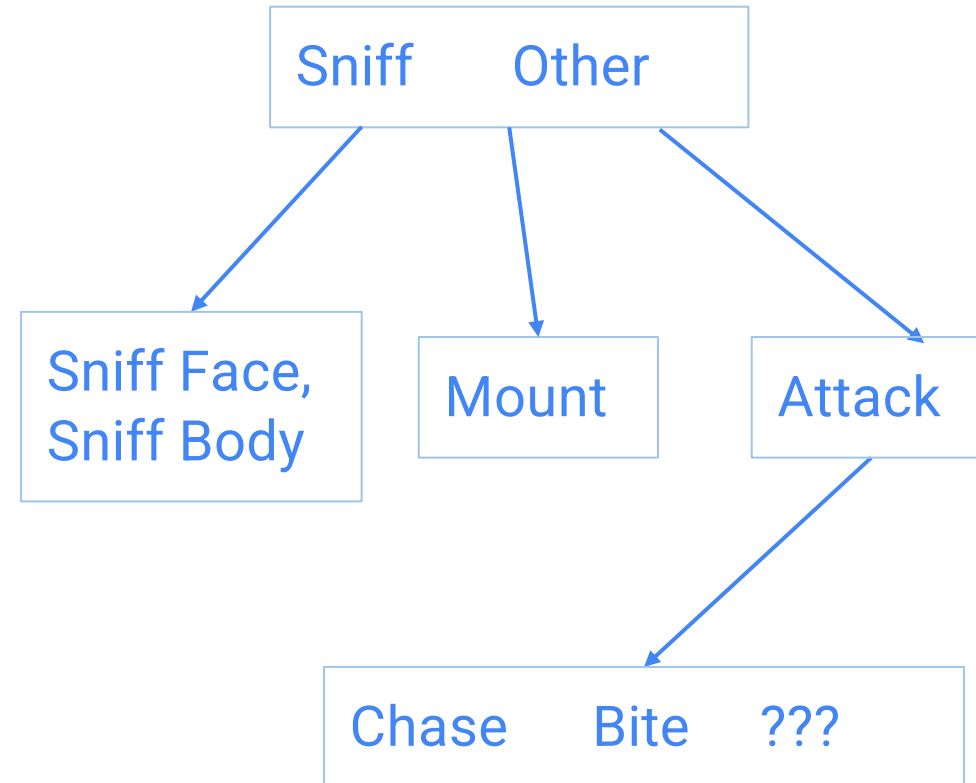
# Integration into existing tool (Bento)



# Extension to unsupervised learning



???

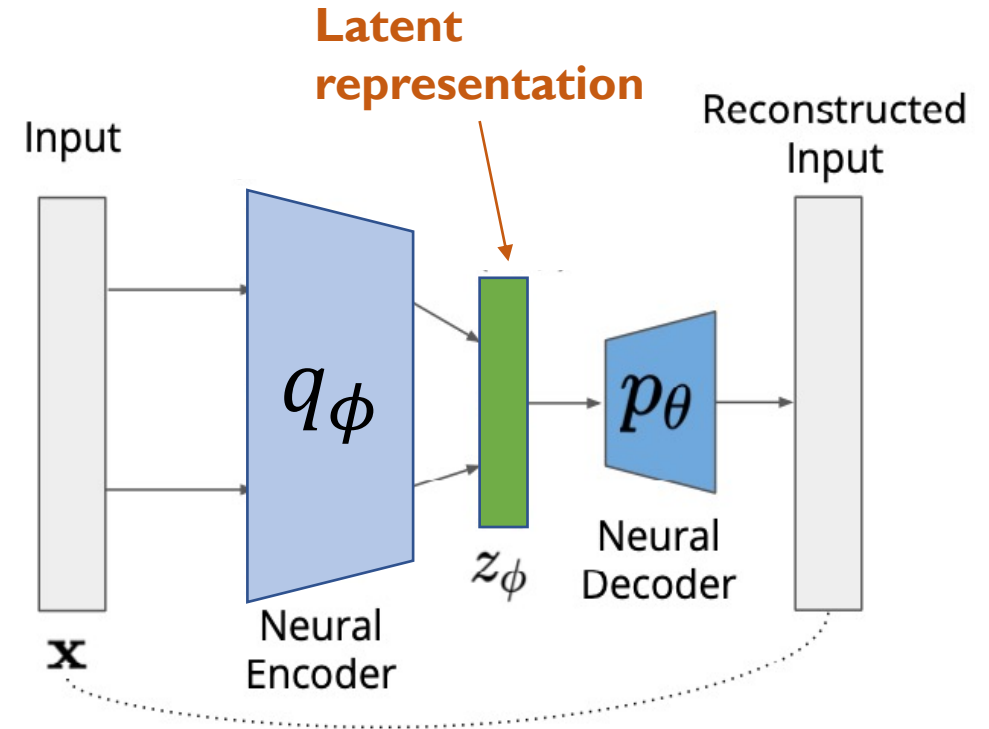


# Variational autoencoders (VAEs)

Latent representations capture semantics of inputs

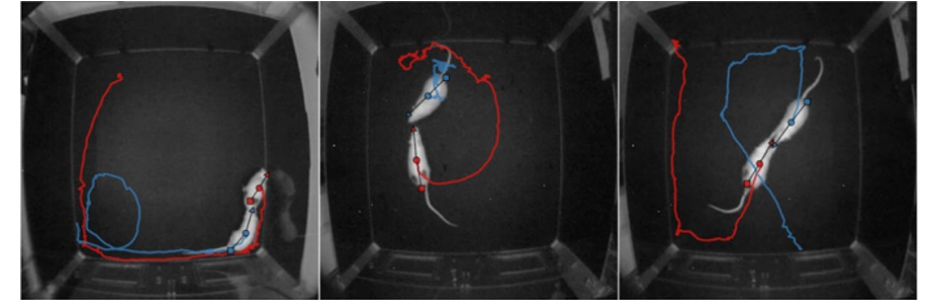
## In behavior analysis:

- Cluster the representations
- Create new labels that capture the clusters



During training, maximize  $\text{ELBO} := \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \leq \log p(\mathbf{x})$

# Neurosymbolic encoders



$$\mathbb{1}_{[>-7.02]} \left[ \begin{array}{l} \text{mapaverage (fun } x_t. \\ \quad \text{multiply (ResidentSpeedAffine}_{[-6.28];-8.28}(x_t), \\ \quad \text{NoseTailDistAffine}_{[.042];-9.06}(x_t)) \ x \end{array} \right] \vdots \vdots$$

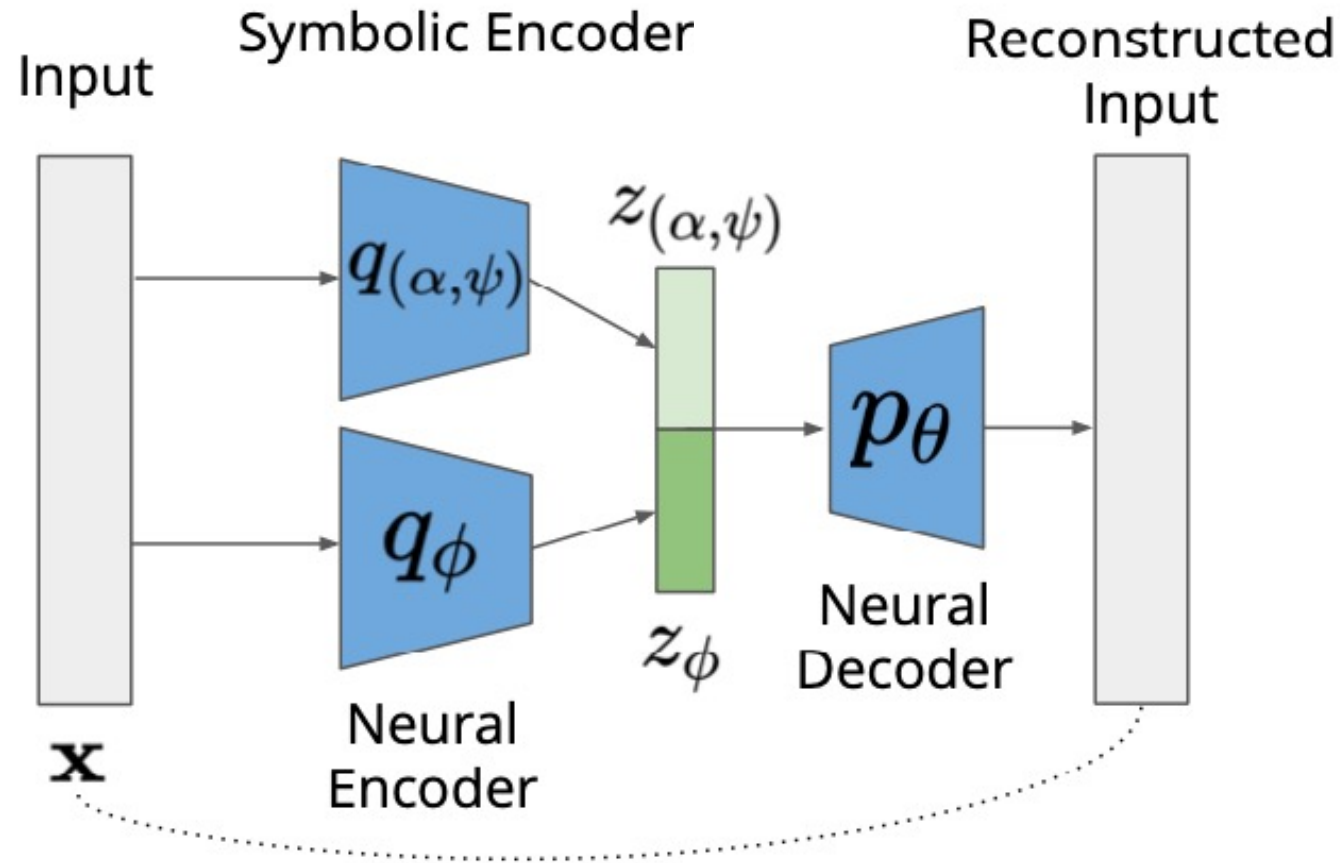
**Cluster 0:** The mice are further apart

- Second term is positive, negative product is less than the threshold.

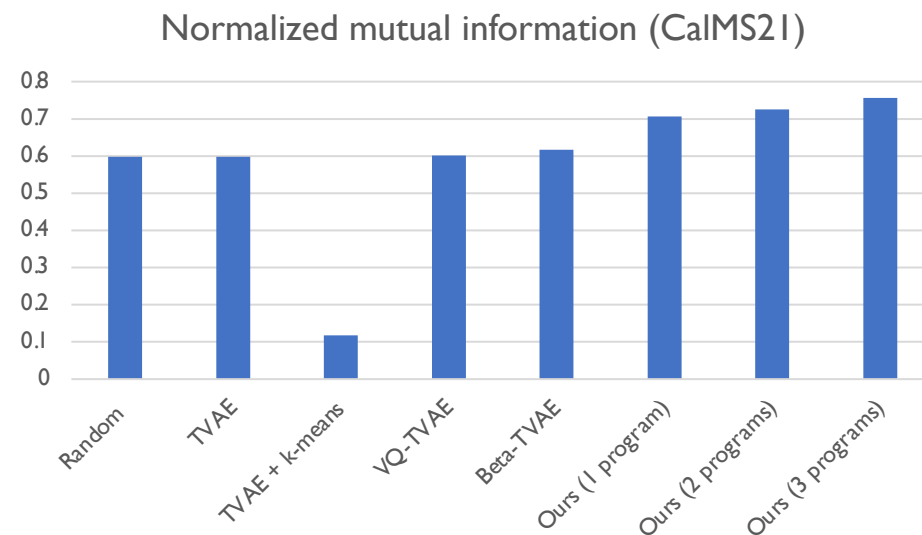
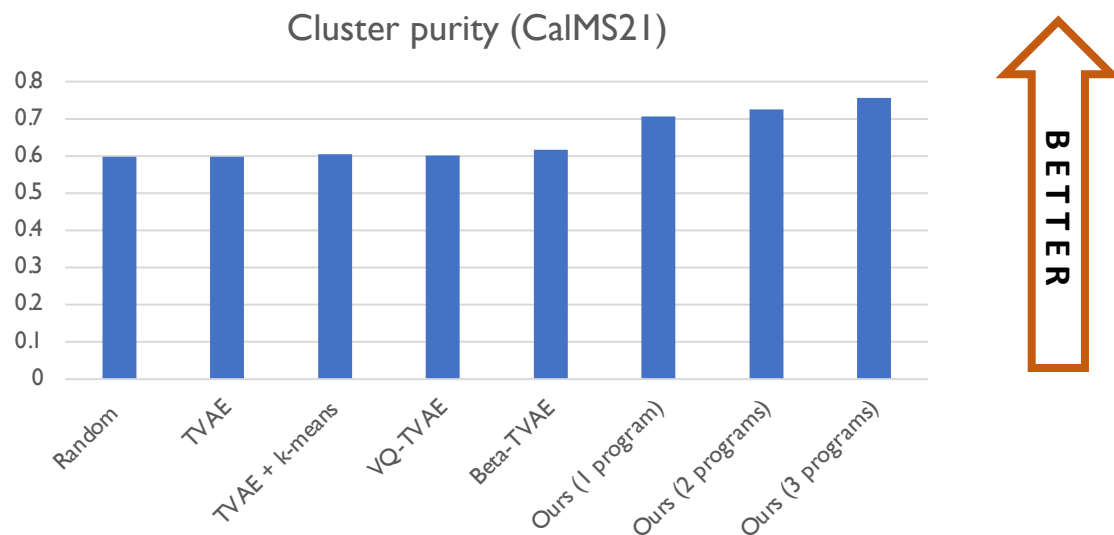
**Cluster 1:** The mice are close together

- Second term is negative, product is positive.

# VAEs with neurosymbolic encoders



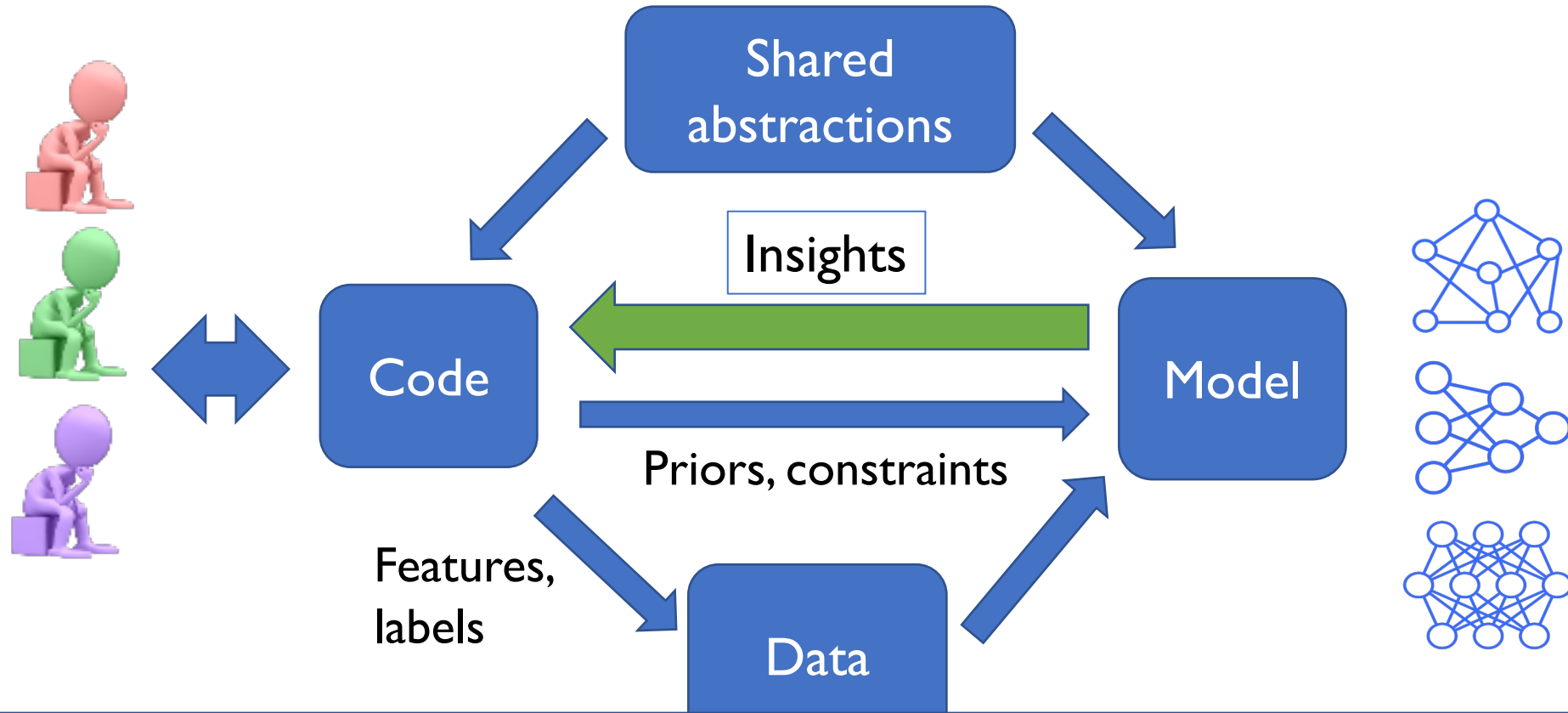
# Results (on human-annotated behavior data)



More well-structured latent spaces

Comparable performance to expert-written programs in downstream tasks

# What's ahead?



**Full-stack AI-aided science through neurosymbolic programming**

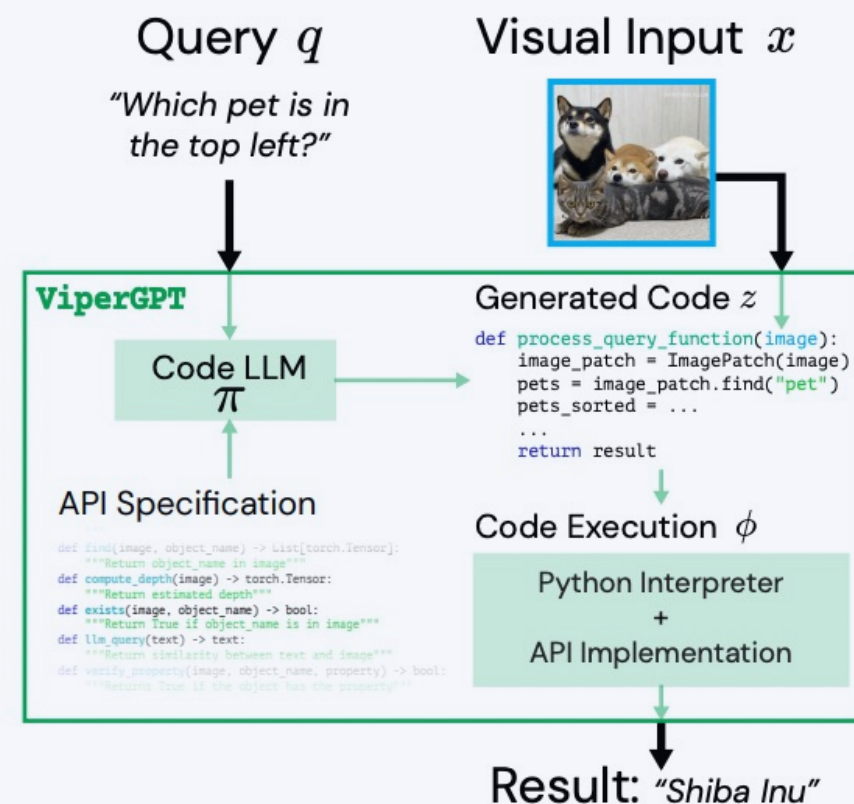
# Challenge: Scalability

Searching for program structures is fundamentally expensive.

## Possible recipes:

- Large Language Models
- Parallelism
- ...

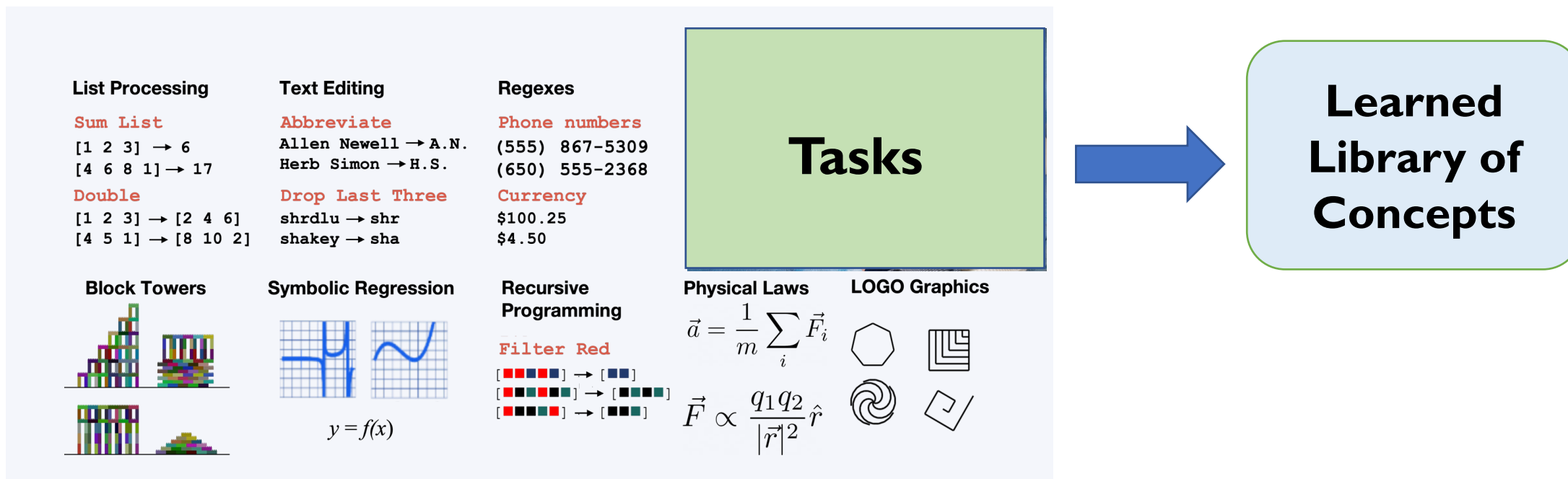
**ViperGPT: Visual Inference via Python Execution for Reasoning.** Suris, Menon, Vondrick, 2023.



# Challenge: Vocabulary discovery

Where does the DSL come from?

**Possible recipe:** Library learning




**Dreamcoder: Growing generalizable, interpretable knowledge with wake-sleep Bayesian learning.** Ellis et al., 2021.

# Challenge: Vocabulary Discovery

**Possible recipe:** Symbol discovery through vision-language models

**Food101**  
**guacamole** (90.1%) Ranked 1 out of 101 labels



- ✓ a photo of **guacamole**, a type of food.
- ✗ a photo of **ceviche**, a type of food.
- ✗ a photo of **edamame**, a type of food.
- ✗ a photo of **tuna tartare**, a type of food.
- ✗ a photo of **hummus**, a type of food.

**Learning Transferable Visual Models From Natural Language Supervision.** Radford et al., 2021.

# Neurosymbolic Programming Everywhere!

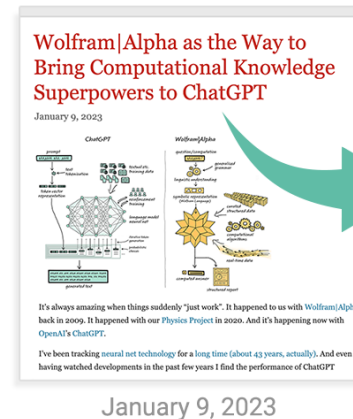
**Understanding the World Through Code**  
Funded through the NSF Expeditions in Computing Program



[Vechev et al., 2023]



[Naik et al., 2022]



Used Wolfram

Here is the current configuration of Jupiter's Galilean moons:

Callisto | Europa | Jupiter | Io | Ganymede (as viewed from Earth)

Wolfram

Install

Access computation, math, curated knowledge & real-time data through Wolfram|Alpha and Wolfram Language

Moon	Mass (kg)	Average Radius (mi)	Rotation Period	Age (billion yr)
Io	$8.9298 \times 10^{22}$	1131.82	42.46 h	4.5
Europa	$4.7987 \times 10^{22}$	969.84	85.22 h	4.5
Ganymede	$1.4815 \times 10^{23}$	1635	7.155 days	4.5
Callisto	$1.0757 \times 10^{23}$	1497.7	16.69 days	4.5

March 23, 2023

[OpenAI Plugins, 2023]

December 5, 2022, 10 am Chicago time

# Neurosymbolic Programming

Tutorial at Neural Information Processing Systems, 2022

## Speakers



Swarat Chaudhuri  
UT Austin



Armando Solar-Lezama  
MIT

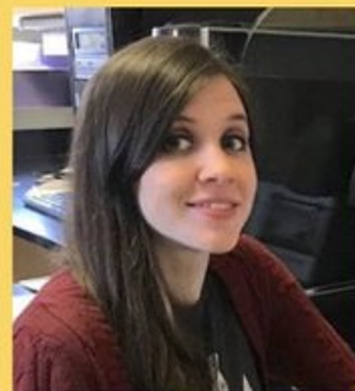


Jennifer J. Sun  
Caltech

## Panelists



Jeevana Inala  
Microsoft Research,  
Redmond



Ann Kennedy  
Northwestern University



Pushmeet Kohli  
Deepmind



Sriram Rajamani  
Microsoft Research, India



Yisong Yue  
Caltech  
(Moderator)

# Acknowledgements



Yisong  
Yue



Armando  
Solar-Lezama



Jennifer  
Sun



Ann  
Kennedy



Omar  
Costilla-Reyes



Ameesh  
Shah



Eric  
Zhan



Megan  
Tjandrasuwita



Atharva  
Sehgal



# Notebooks on neurosymbolic programming for science



Open in Colab

> tutorial\_notebook1.ipynb



[bit.ly/neurosym\\_tutorial\\_pop123](https://bit.ly/neurosym_tutorial_pop123)

