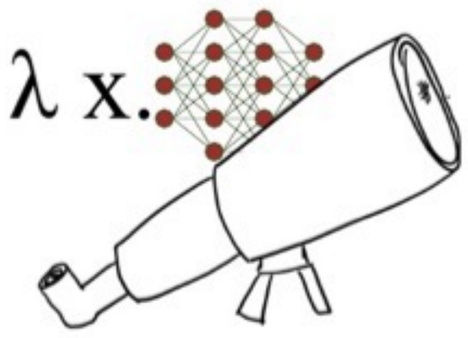
Neurosymbolic Programming



Yisong Yue

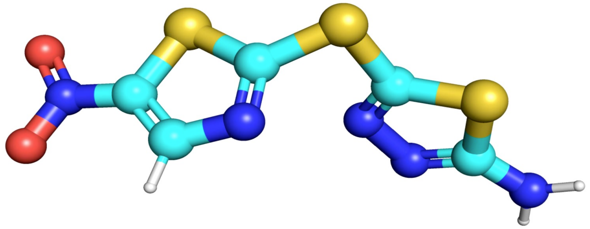
Caltech

*** Includes materials from: Armando Solar-Lezama, Osbert Bastani, Swarat Chaudhuri, Ann Kennedy, David Anderson

Machine learning is transforming science

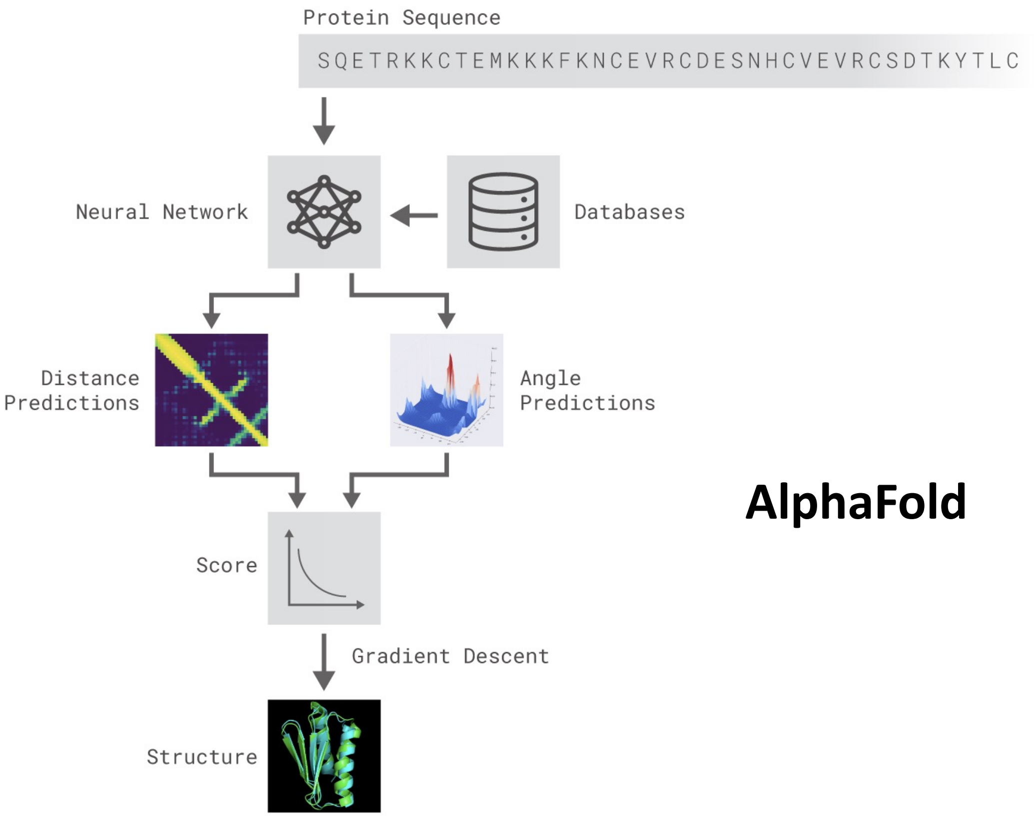
CRISPR ML

https://www.microsoft.com/en-us/research/project/crispr/



Halicin: structurally new antibiotic

https://news.mit.edu/2020/artificial-intelligence-identifies-new-antibiotic-0220

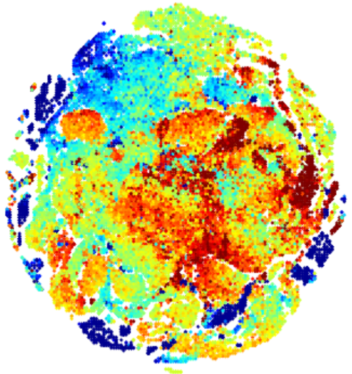


AlphaFold

Personalized Exoskeletons

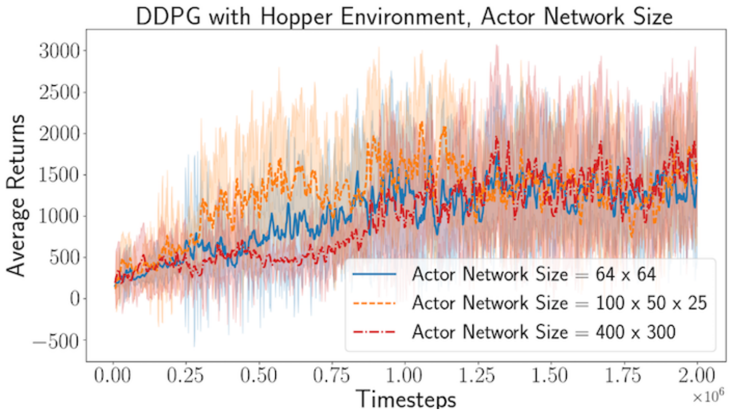
http://roams.caltech.edu/

But something is missing…

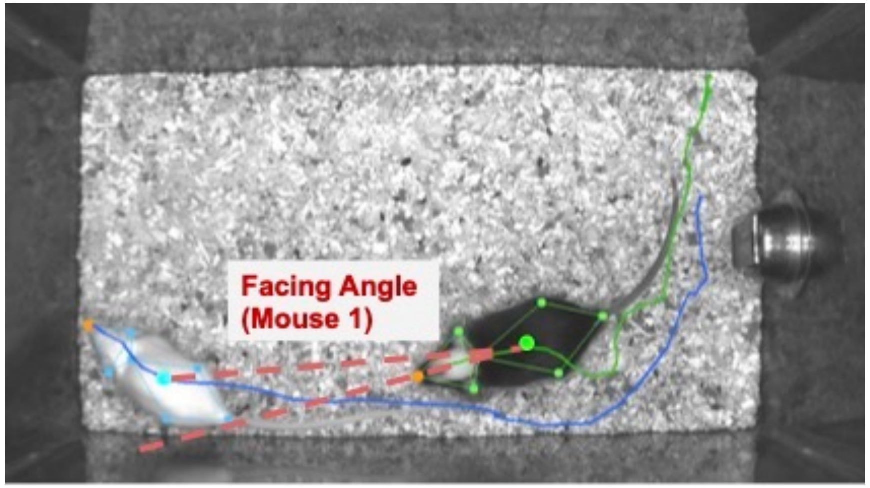


https://arxiv.org/abs/1611.00094

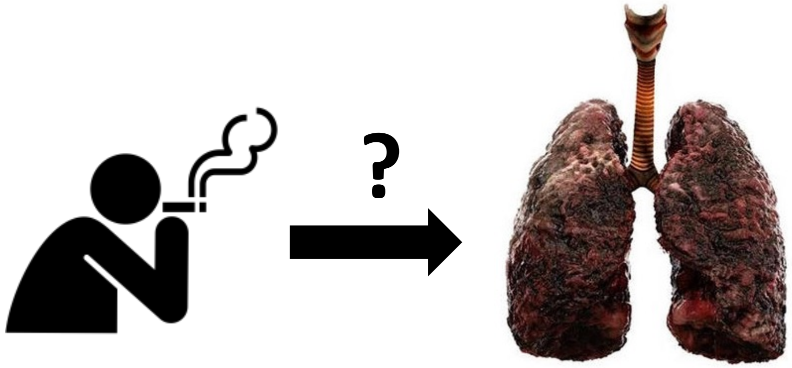
Interpretability



Data Efficiency
https://arxiv.org/abs/1709.06560

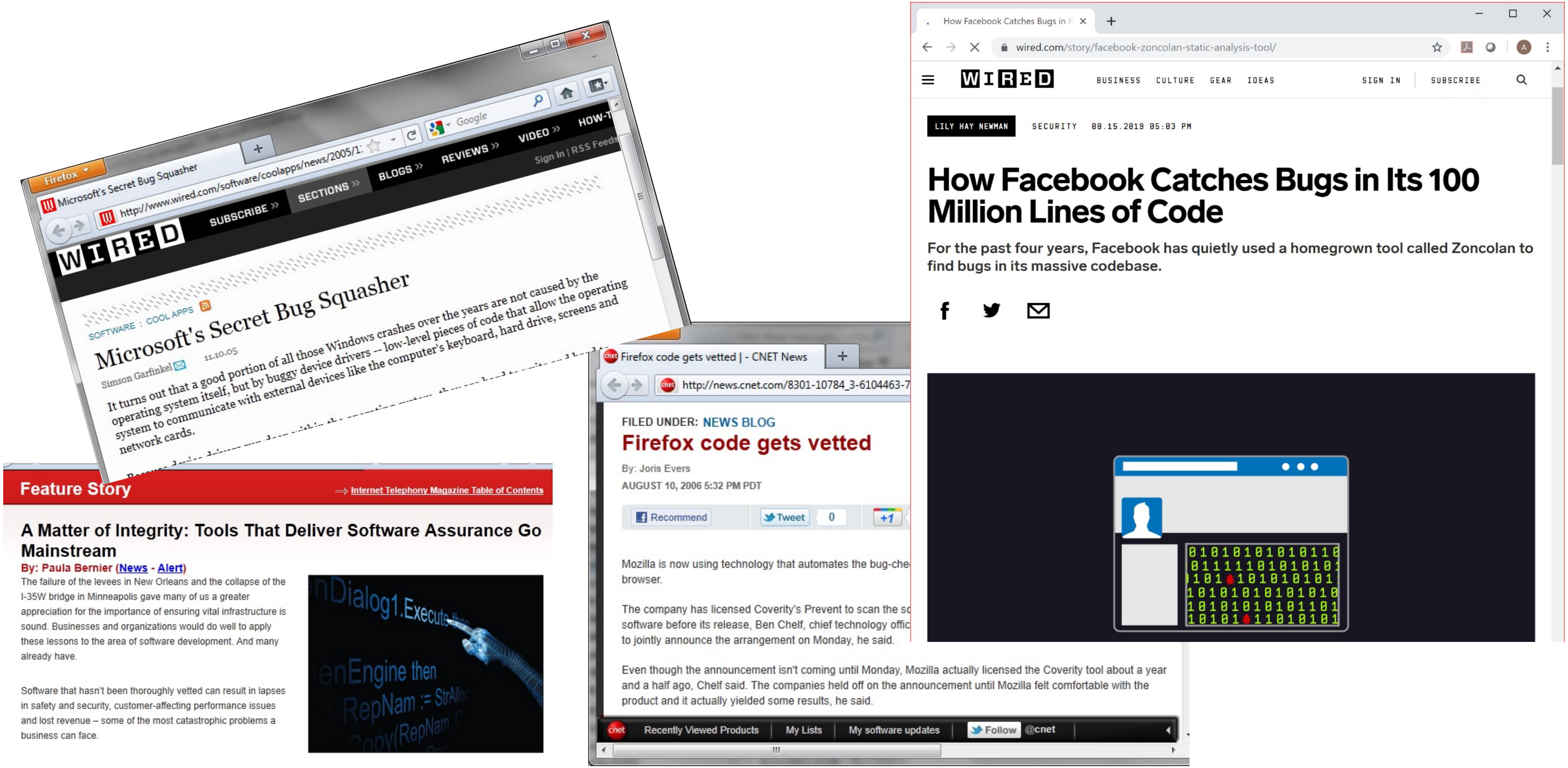


Domain Knowledge



Correlation vs Causation

A revolution in formal methods



Microsoft's Secret Bug Squasher

http://www.wired.com/software/coolapps/news/2005/1

WIRED

SUBSCRIBE » SECTIONS » BLOGS » REVIEWS » VIDEO » HOW-T

Sign In | RSS Feeds

SOFTWARE : COOL APPS

Microsoft's Secret Bug Squasher

Simson Garfinkel 11.10.05

It turns out that a good portion of all those Windows crashes over the years are not caused by the operating system itself, but by buggy device drivers -- low-level pieces of code that allow the operating system to communicate with external devices like the computer's keyboard, hard drive, screens and network cards.

Feature Story

Internet Telephony Magazine Table of Contents

A Matter of Integrity: Tools That Deliver Software Assurance Go Mainstream

By: Paula Bernier (News - Alert)

The failure of the levees in New Orleans and the collapse of the I-35W bridge in Minneapolis gave many of us a greater appreciation for the importance of ensuring vital infrastructure is sound. Businesses and organizations would do well to apply these lessons to the area of software development. And many already have.

Software that hasn't been thoroughly vetted can result in lapses in safety and security, customer-affecting performance issues and lost revenue – some of the most catastrophic problems a business can face.

Firefox code gets vetted | - CNET News

http://news.cnet.com/8301-10784_3-6104463-7

FILED UNDER: NEWS BLOG

Firefox code gets vetted

By: Joris Evers

AUGUST 10, 2006 5:32 PM PDT

Recommend Tweet 0 +1

Mozilla is now using technology that automates the bug-che browser.

The company has licensed Coverity's Prevent to scan the so software before its release, Ben Chelf, chief technology offic to jointly announce the arrangement on Monday, he said.

Even though the announcement isn't coming until Monday, Mozilla actually licensed the Coverity tool about a year and a half ago, Chelf said. The companies held off on the announcement until Mozilla felt comfortable with the product and it actually yielded some results, he said.

Recently Viewed Products My Lists My software updates Follow @cnet

How Facebook Catches Bugs in It

wired.com/story/facebook-zoncolan-static-analysis-tool/

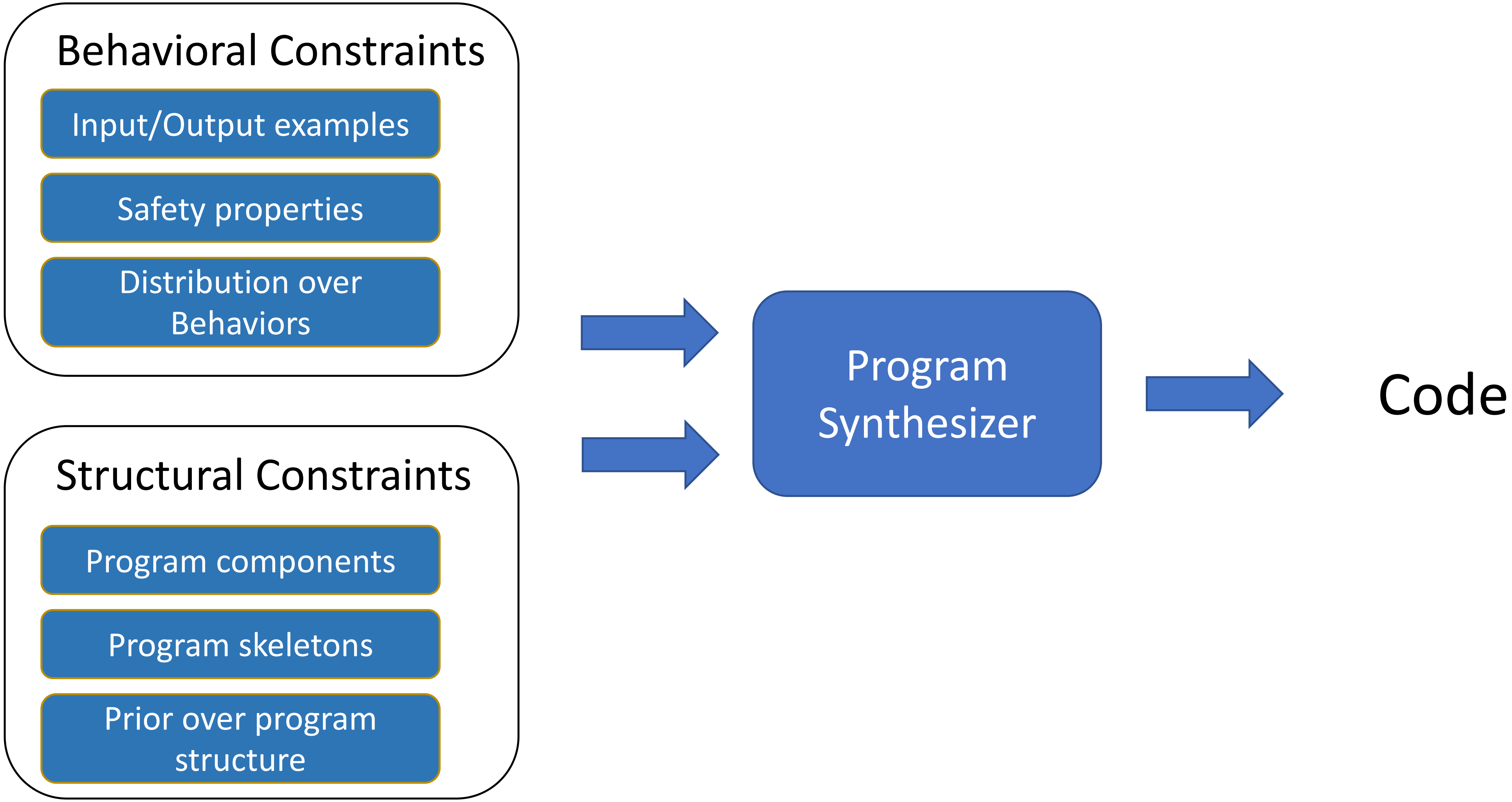
WIRED BUSINESS CULTURE GEAR IDEAS SIGN IN SUBSCRIBE

LILY HAY NEWMAN SECURITY 08.15.2019 05:03 PM

How Facebook Catches Bugs in Its 100 Million Lines of Code

For the past four years, Facebook has quietly used a homegrown tool called Zoncolan to find bugs in its massive codebase.

Program Synthesis



Scientific knowledge is code

$$E=mc^2$$

Scientific knowledge is code

Understanding Morpho-phonology

⟨/opɛn/, [**root**:OPEN]⟩ ⟨/d/, [**tense**:PAST]⟩

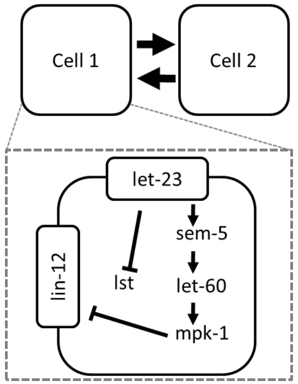
⟨/opɛnd/, [**root**:OPEN;**tense**:PAST]⟩

/wɔkd/ → [wɔkt]

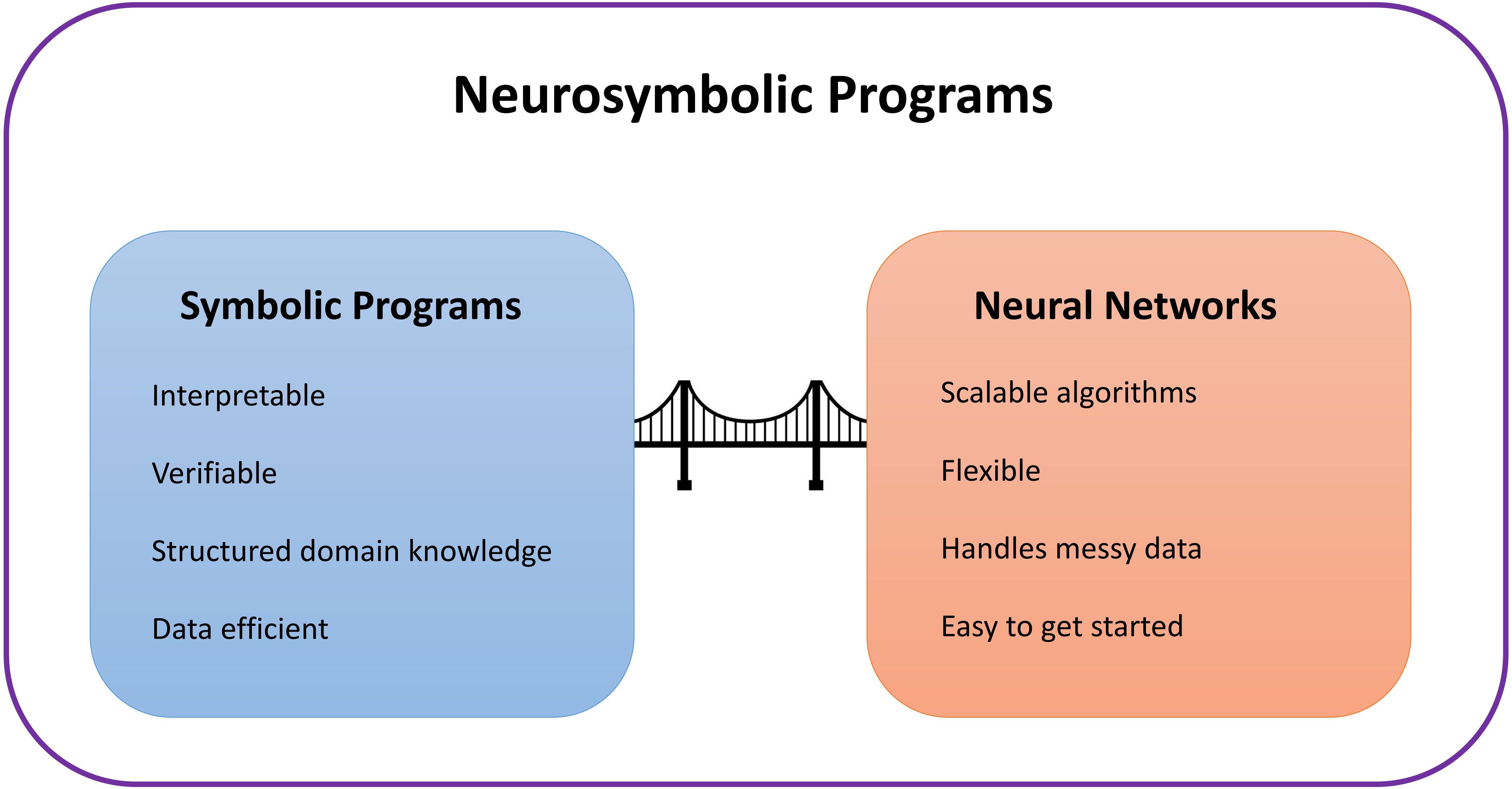
Underlying form Surface form

https://dspace.mit.edu/handle/1721.1/113870

Synthesis of biological models

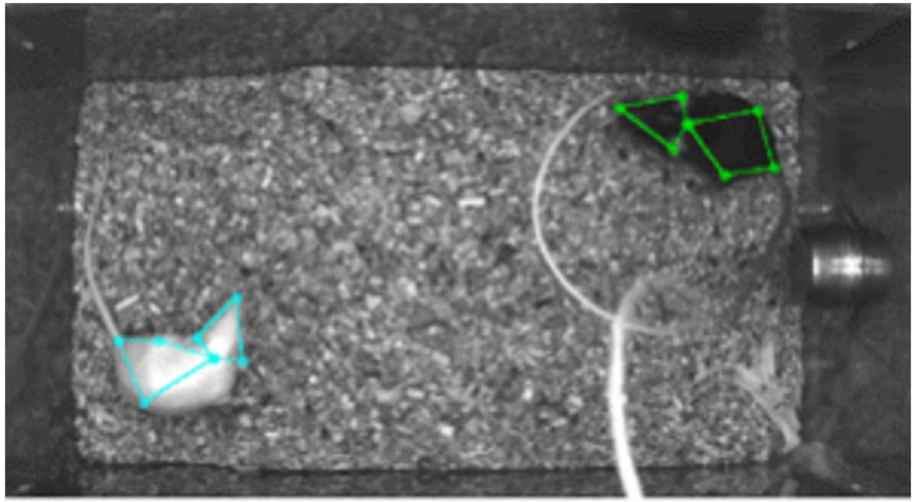


https://dl.acm.org/doi/10.1145/2480359.2429125

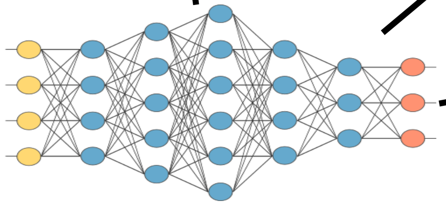

Neurosymbolic Programs
Symbolic Programs
Interpretable
Verifiable
Structured domain knowledge
Data efficient
Neural Networks
Scalable algorithms
Flexible
Handles messy data
Easy to get started

Example in Behavior Analysis

Goal: Classify “sniff” action between two mice



$$
\begin{aligned}
&\mathbf{map}\ (\mathbf{fun}\ x_t.\\
&\quad \mathbf{if}\ \mathit{DistAffine}_{[.0217];-.2785}(x_t)\\
&\quad\ \mathbf{then}\ \mathit{AccAffine}_{[-.0007,.0055,.0051,-.0025];3.7426}(x_t)\ \mathbf{else}\ \mathit{DistAffine}_{[-.2143];1.822)}(x_t))\ x
\end{aligned}
$$



learned in conjunction with program

Neurosymbolic learning isn't new…

…but it's a good time to push on it!

- **Respective revolutions in both fields**
  - Rapidly maturing tools
- **New algorithms that can scale**
  - Computation (e.g., neural-guided search)
  - Data (e.g., programmatic weak supervision)
- **Demands by the domain experts & science applications**

Understanding the World Through Code

Funded through the NSF Expeditions in Computing Program

http://www.neurosymbolic.org/

PIs

Armando Solar-Lezama
Associate Professor, MIT

Swarat Chaudhuri
Associate Professor, UT Austin

Yisong Yue
Professor, Caltech

Regina Barzilay
Professor, MIT

Isil Dillig
Associate Professor, UT Austin

Osbert Bastani
Research Assistant Professor, University of Pennsylvania

Michael Carbin
Assistant Professor, MIT

Martin Rinard
Professor, MIT

Phillip Sharp
Institute Professor and Professor of Biology, MIT

Tommi Jaakkola
Professor, MIT

Noah Goodman
Associate Professor, Stanford

Chris Jermaine
Professor, Rice

Advisory Board

Josh Tenenbaum
Professor, MIT

Pushmeet Kohli
Principal scientist and research team leader, Deep Mind

Rishabh Singh
Research scientist, Google Brain

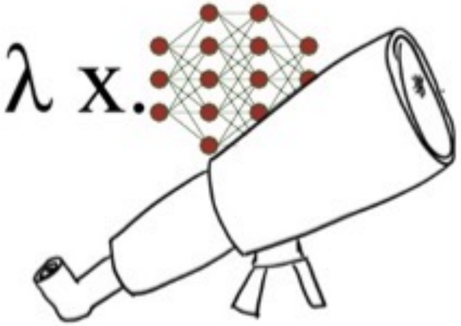
Justin Gottschlich
Head of Machine Programming Research, Intel

Satish Chandra
Engineering Manager, Facebook

Susmit Jha
Principal Computer Scientist, SRI International

Closing the loop between data and insight

Neurosymbolic Models

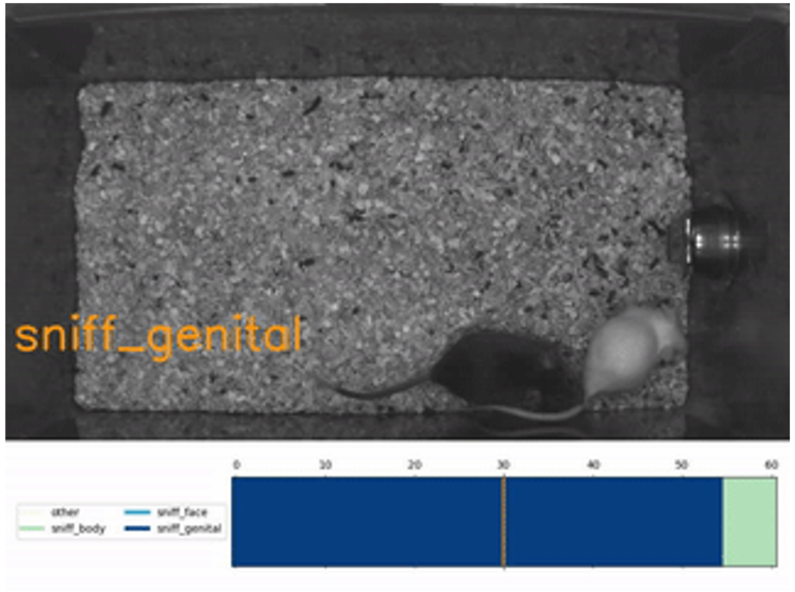


Query model
Extract insights
Inject domain knowledge

Fit model to data
Suggest experiments

Domain Expert

Check model insights against data



The Basic Recipe

Inputs

Algebraic Operators

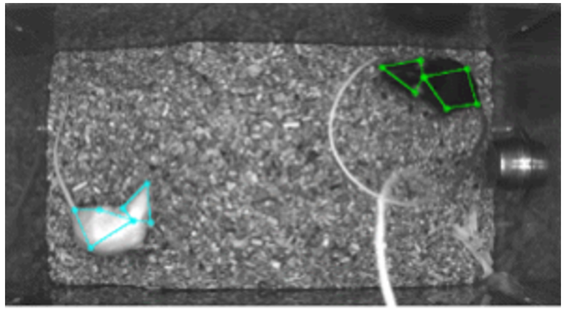
Parameterized Operators
(θ are the parameters)

$$\alpha \quad ::= \quad x \mid \oplus(\alpha_1, \ldots, \alpha_k) \mid \oplus_\theta(\alpha_1, \ldots, \alpha_k)$$
$$\textbf{if } \alpha_1 \textbf{ then } \alpha_2 \textbf{ else } \alpha_3 \mid \textbf{sel}_S\, x \mid \textbf{mapaverage}\ (\textbf{fun } x_1 . \alpha_1)\ x$$

Program Structure

Domain Specific Language (DSL) -- "Family of programs"

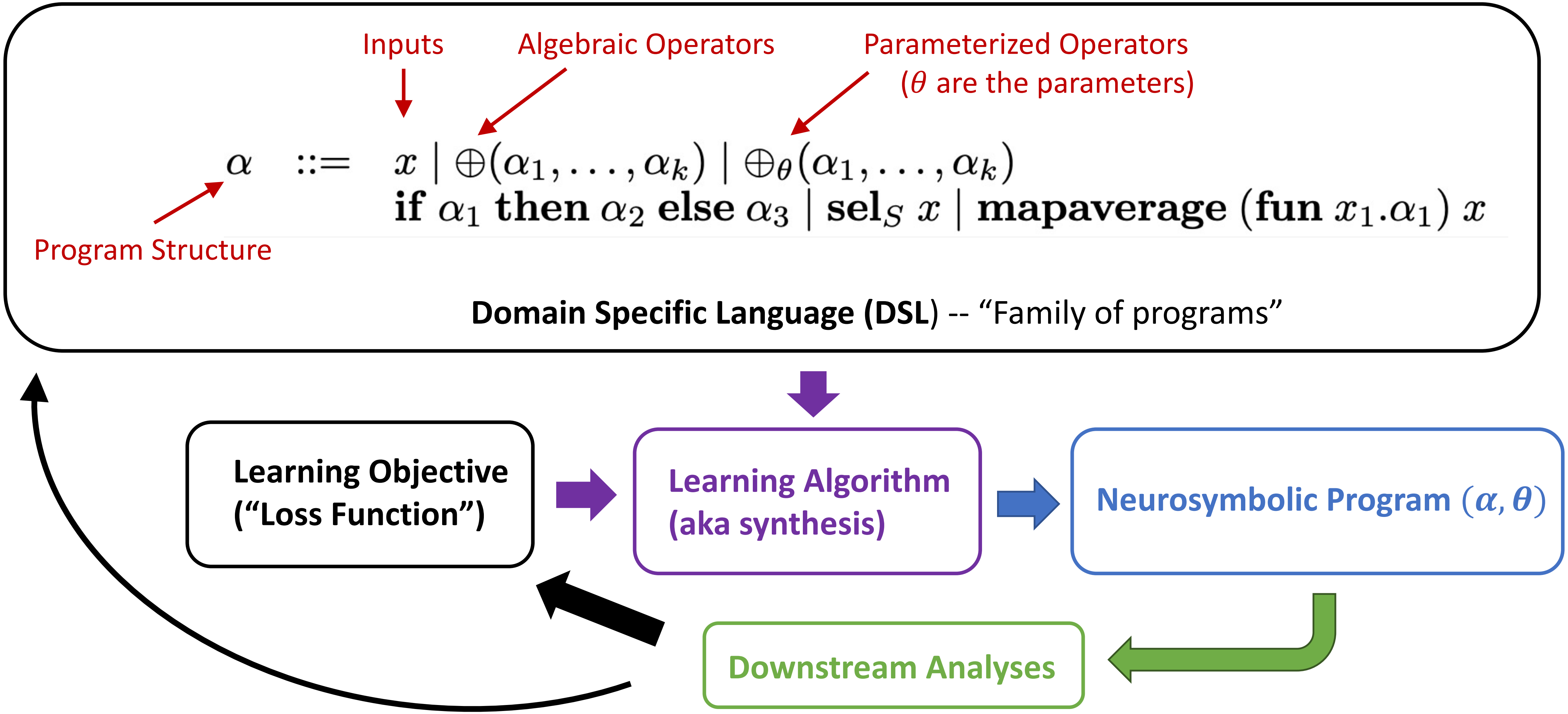
Recall Earlier Example:



Examples of $\oplus_\theta$

$$\textbf{map}\ (\textbf{fun } x_t .$$
$$\textbf{if } \mathit{DistAffine}_{[.0217];-.2785}(x_t)$$
$$\textbf{then } \mathit{AccAffine}_{[-.0007,.0055,.0051,-.0025];3.7426}(x_t) \textbf{ else } \mathit{DistAffine}_{[-.2143];1.822)}(x_t))\ x$$

The Basic Recipe



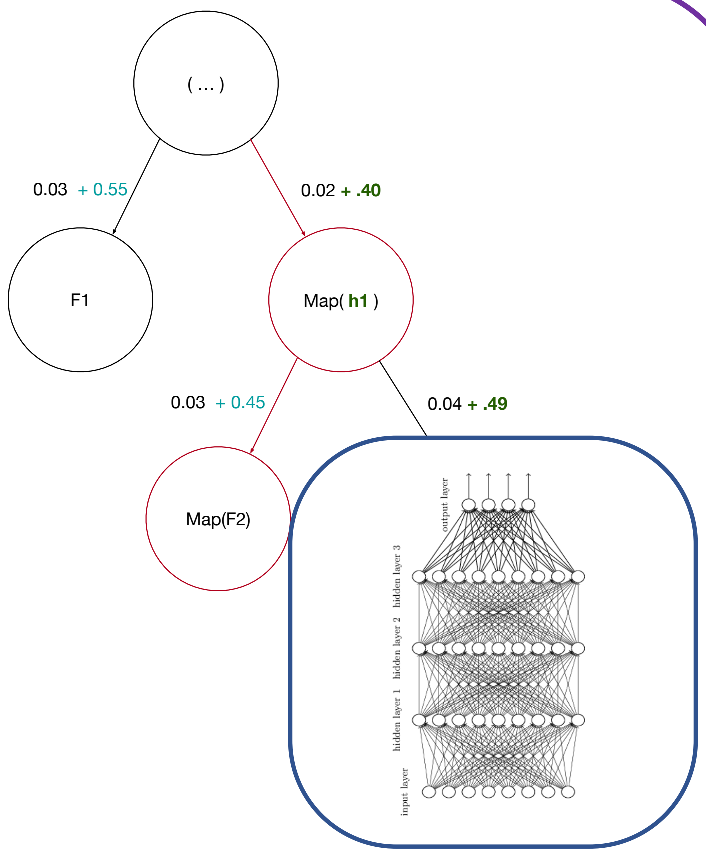
Observations:

- Fixed program structure $\boldsymbol{\alpha}$ → train $\boldsymbol{\theta}$ via gradient descent
- Setting $\boldsymbol{\alpha}$ as a neural network → standard deep learning
- Finding $\boldsymbol{\alpha}$ is analogous to neural architecture search
  - Sometimes call $\boldsymbol{\alpha}$ the "program architecture"
- Classic program synthesis focuses on $\boldsymbol{\alpha}$, with $\boldsymbol{\theta}$ being very simple

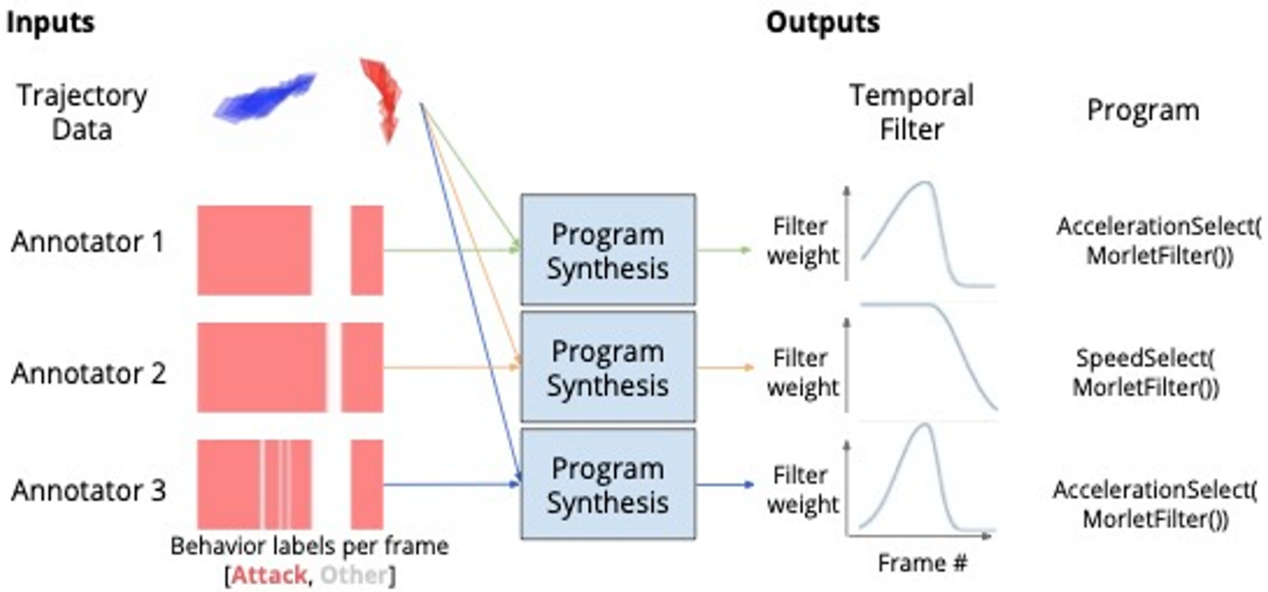
Example Program:

$$\begin{aligned}
&\textbf{map}\ (\textbf{fun}\ x_t. \\
&\quad \textbf{if}\ \mathit{DistAffine}_{[.0217];-.2785}(x_t) \\
&\quad\ \textbf{then}\ \mathit{AccAffine}_{[-.0007,.0055,.0051,-.0025];3.7426}(x_t)\ \textbf{else}\ \mathit{DistAffine}_{[-.2143];1.822)}(x_t))\ x
\end{aligned}$$

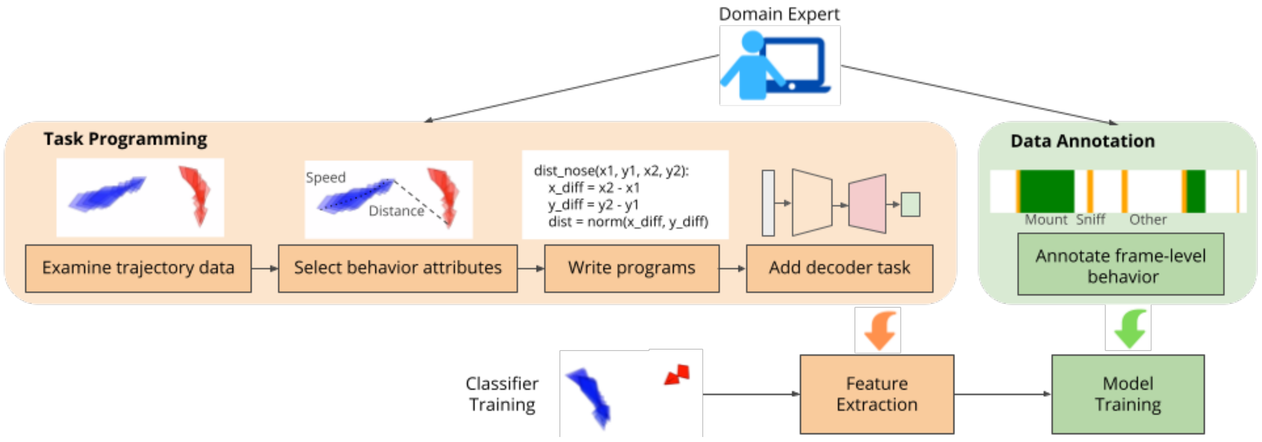
Remainder of Talk



Algorithm Vignette
(Computational Scalability)

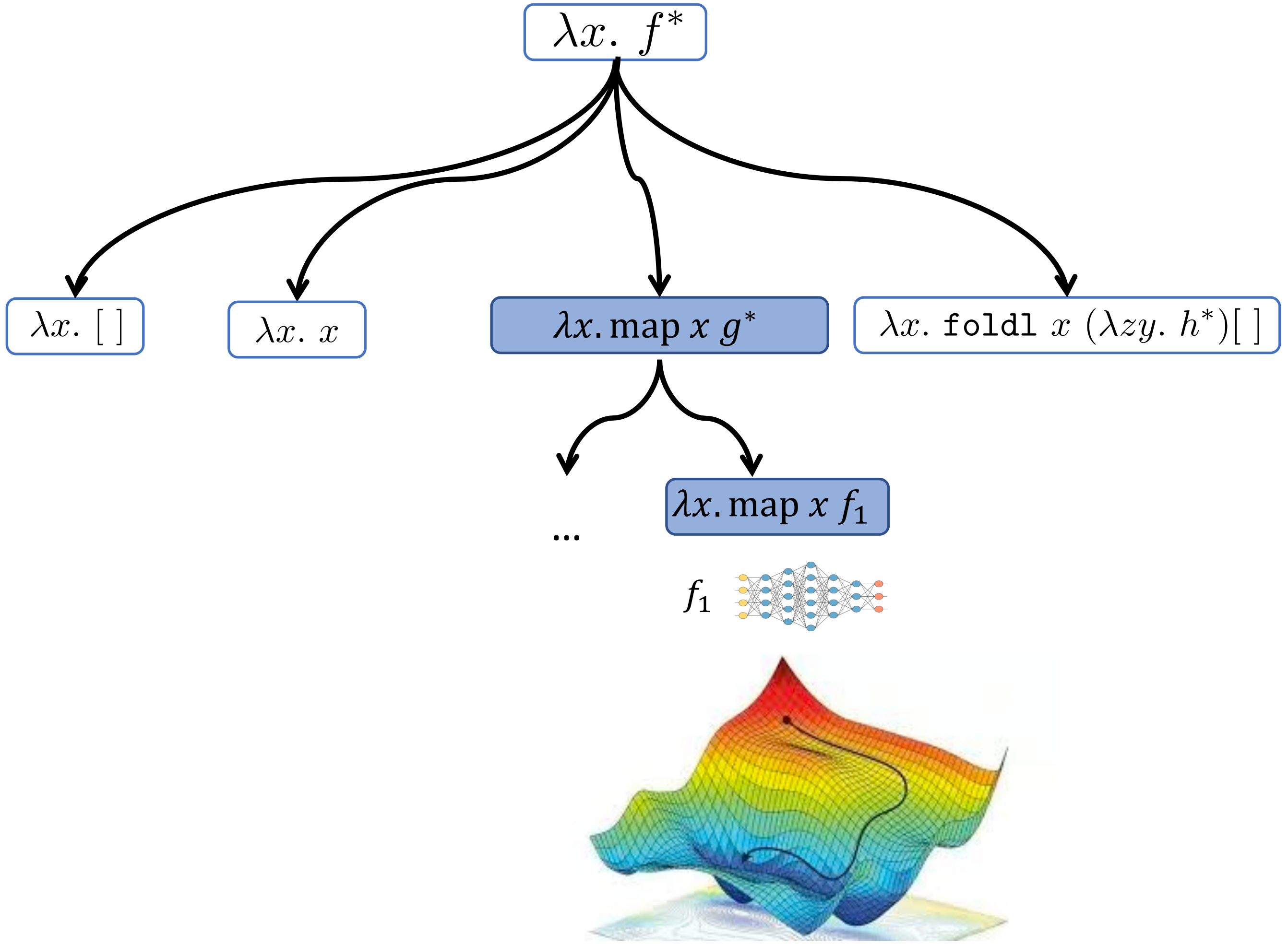


User Study Vignette (Interpretability)



Data Augmentation Vignette (Data Efficiency)

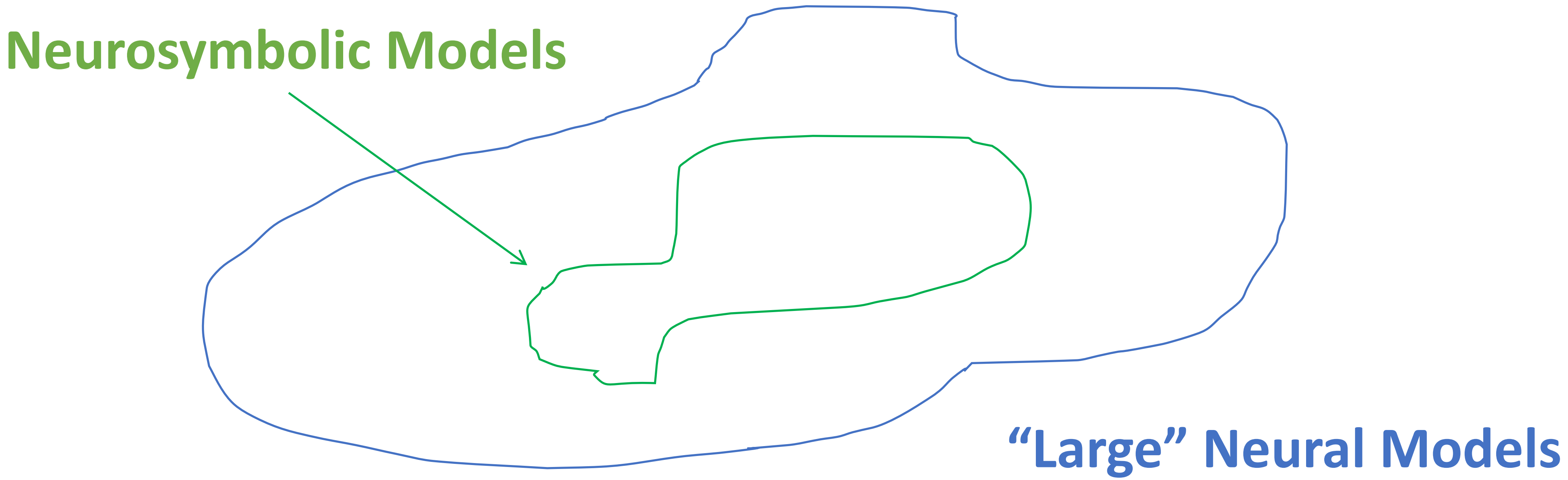
Top-Down Induction



Exponentially large search space!

Popular approaches (e.g., A*) require admissible heuristic

Motivating Observation/Assumption:
Functional Representational Power



"Neural Relaxation" Every neurosymbolic model can be (approximately) represented by some "large" neural model.

NEAR: Neural Admissible Relaxations


Ameesh Shah

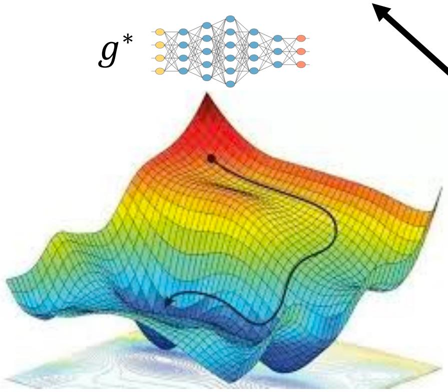

Eric Zhan


Jennifer Sun

$\lambda x.\ f^*$

$\lambda x.\ [\,]$ | $\lambda x.\ x$ | $\lambda x.\,\mathrm{map}\ x\ g^*$ | $\lambda x.\ \texttt{foldl}\ x\ (\lambda zy.\ h^*)[\,]$

g^*

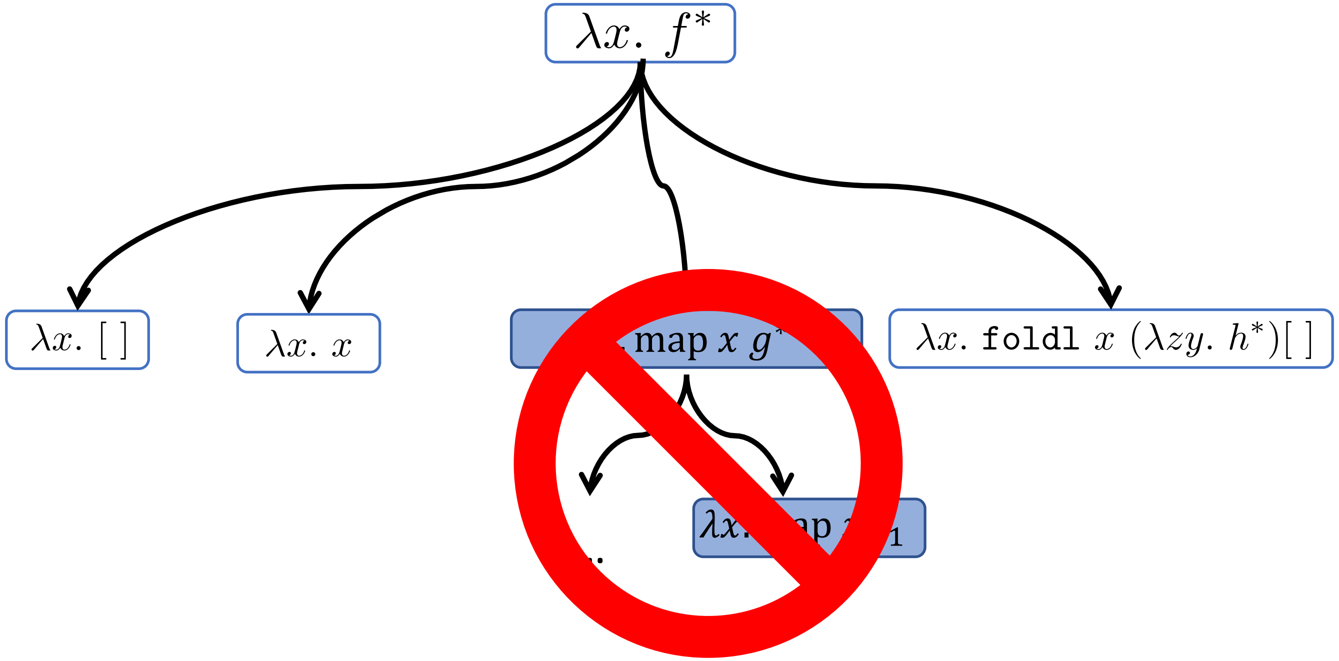


If a large neural network cannot fit this hole, then a program also cannot

Learning Differentiable Programs with Admissible Neural Heuristics, Ameesh Shah*, Eric Zhan*, et al., NeurIPS 2020

NEAR: Neural Admissible Relaxations

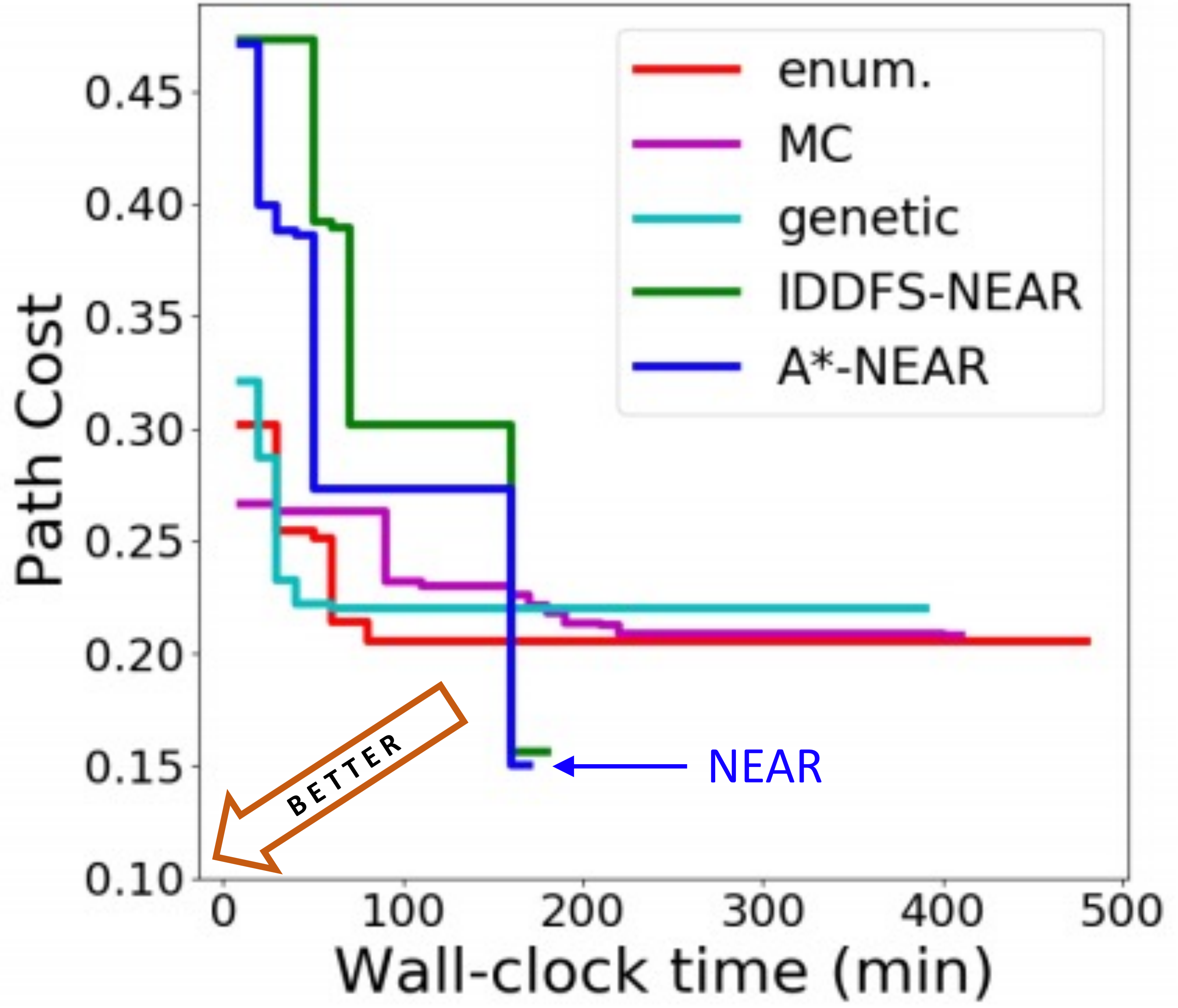
Ameesh Shah
Eric Zhan
Jennifer Sun



Neural Relaxation as Admissible Heuristic!
Usable in any informed search (e.g., A*)

Learning Differentiable Programs with Admissible Neural Heuristics, Ameesh Shah*, Eric Zhan*, et al., NeurIPS 2020

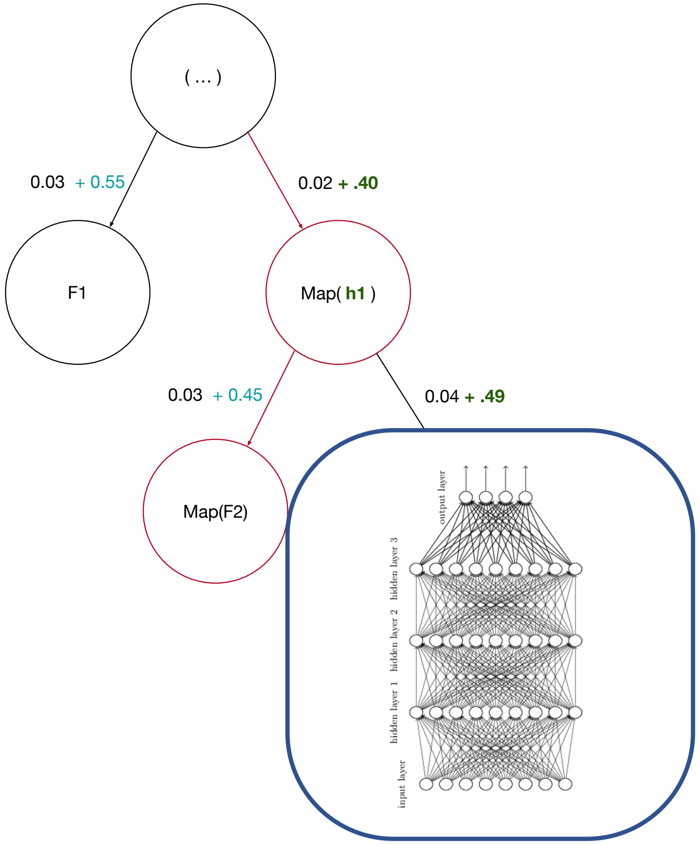
NEAR: Results



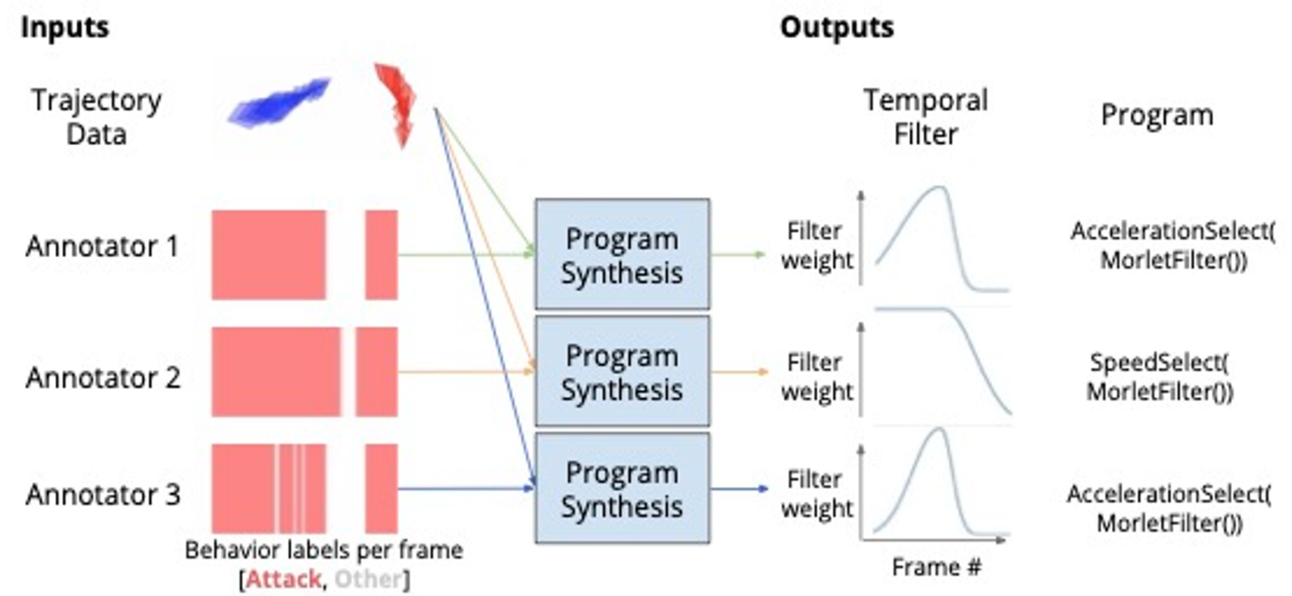
Order of magnitude speedup!

Learning Differentiable Programs with Admissible Neural Heuristics, Ameesh Shah*, Eric Zhan*, et al., NeurIPS 2020

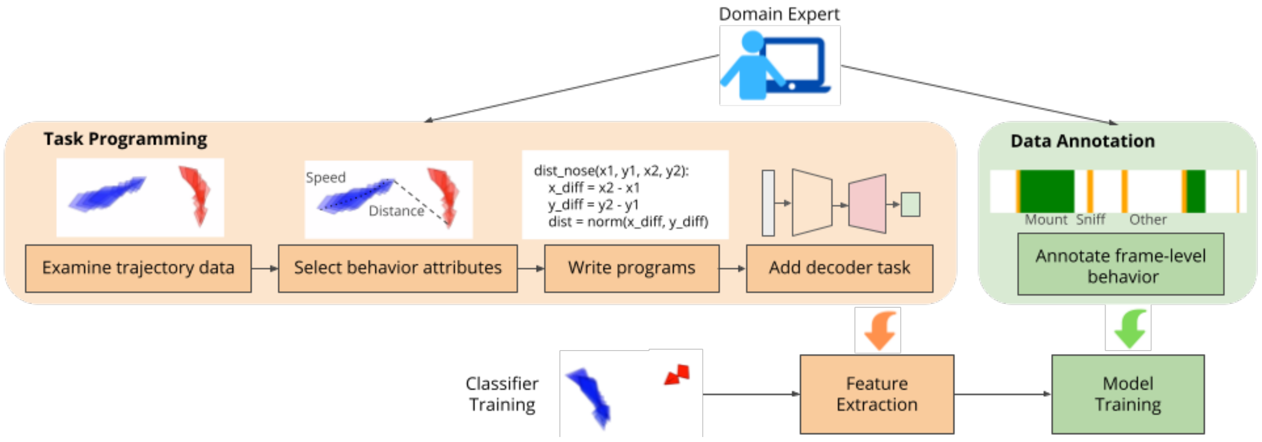
Remainder of Talk



Algorithm Vignette
(Computational Scalability)

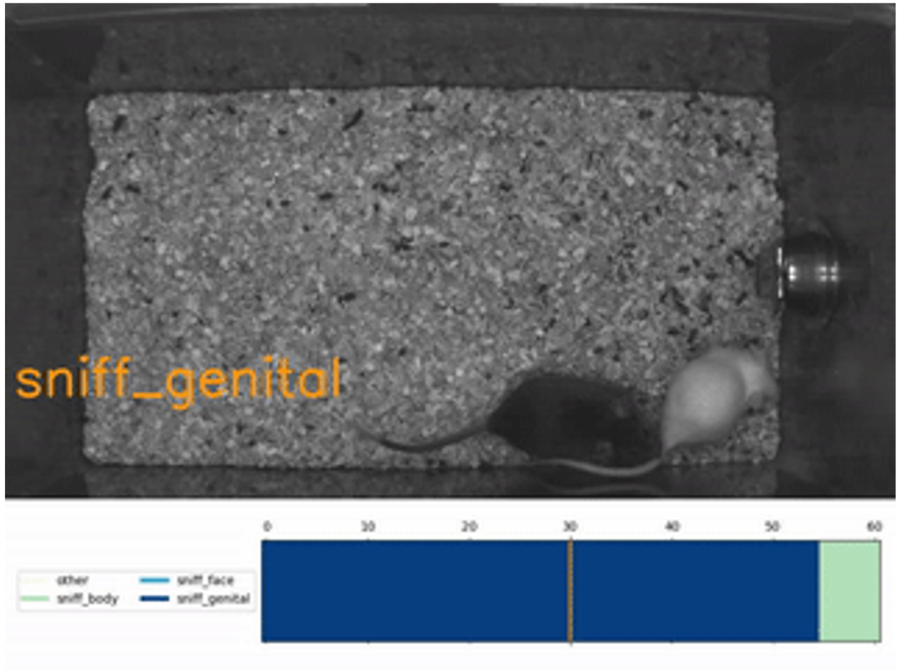


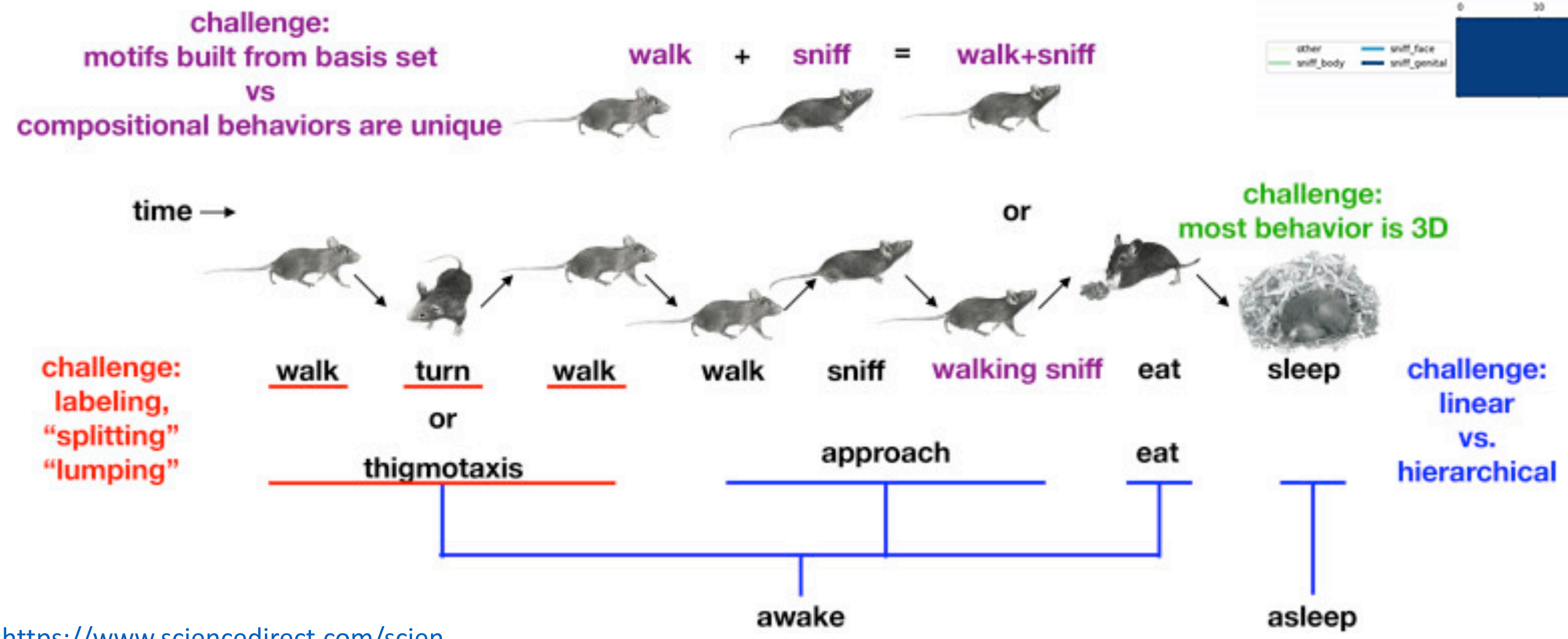
User Study Vignette (Interpretability)



Data Augmentation Vignette (Data Efficiency)

Behavior categorization & definitions are ambiguous!



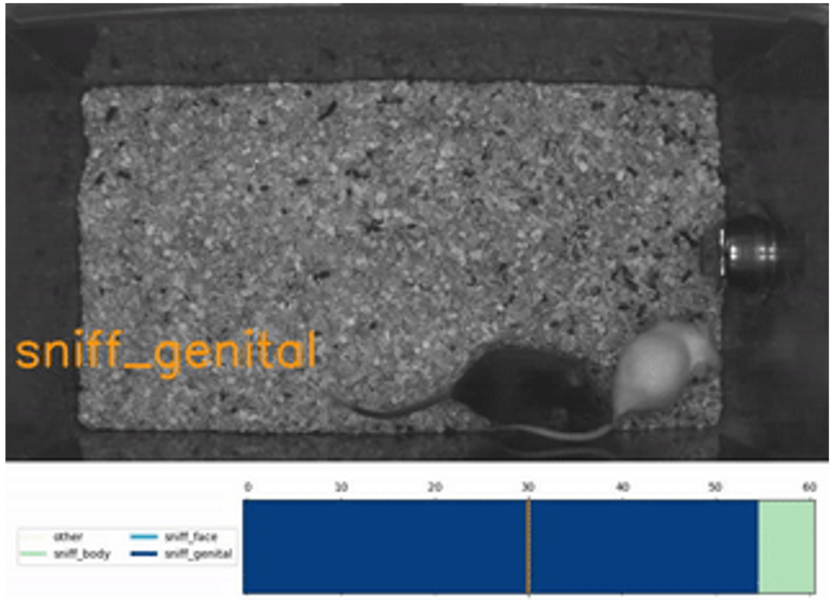


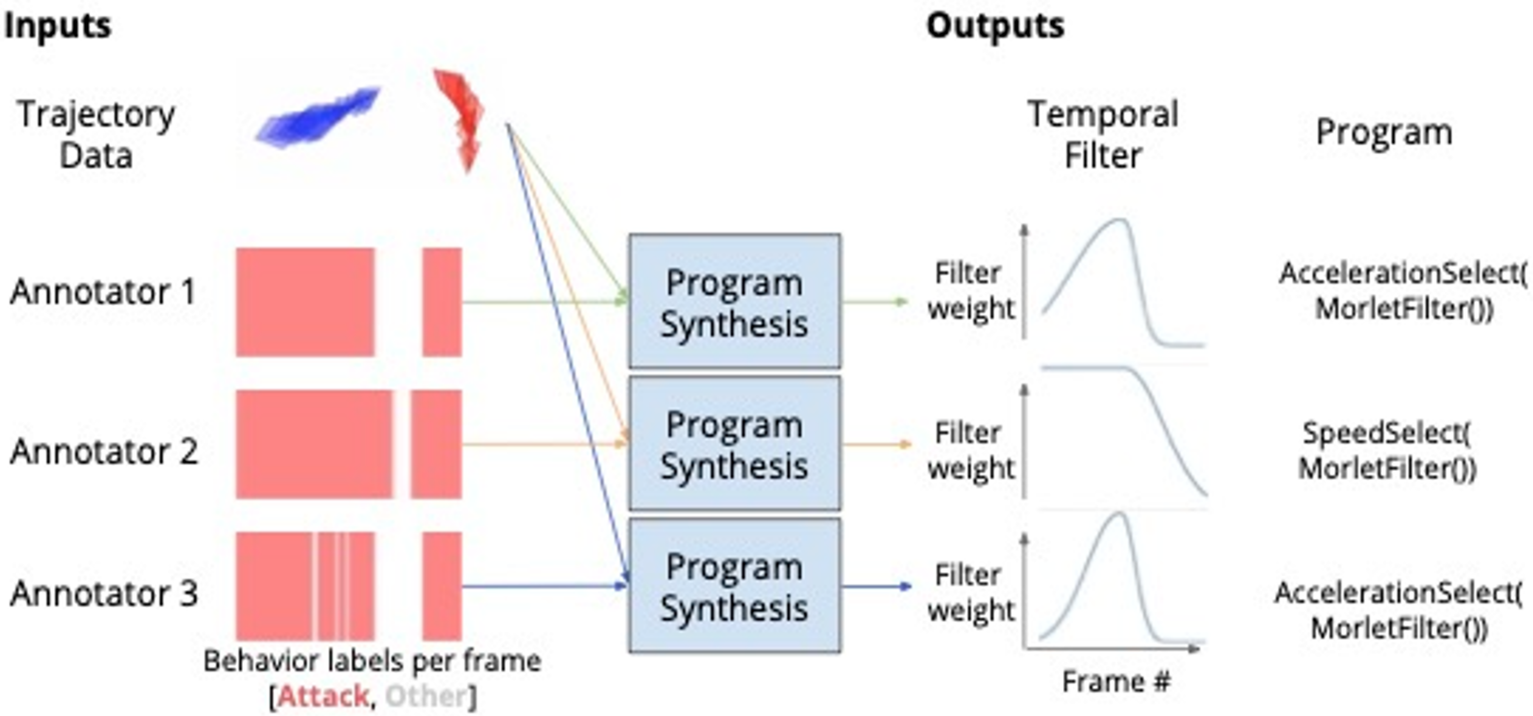
https://www.sciencedirect.com/science/article/pii/S0896627319308414

Understanding annotator differences

Megan Tjandrasuwita

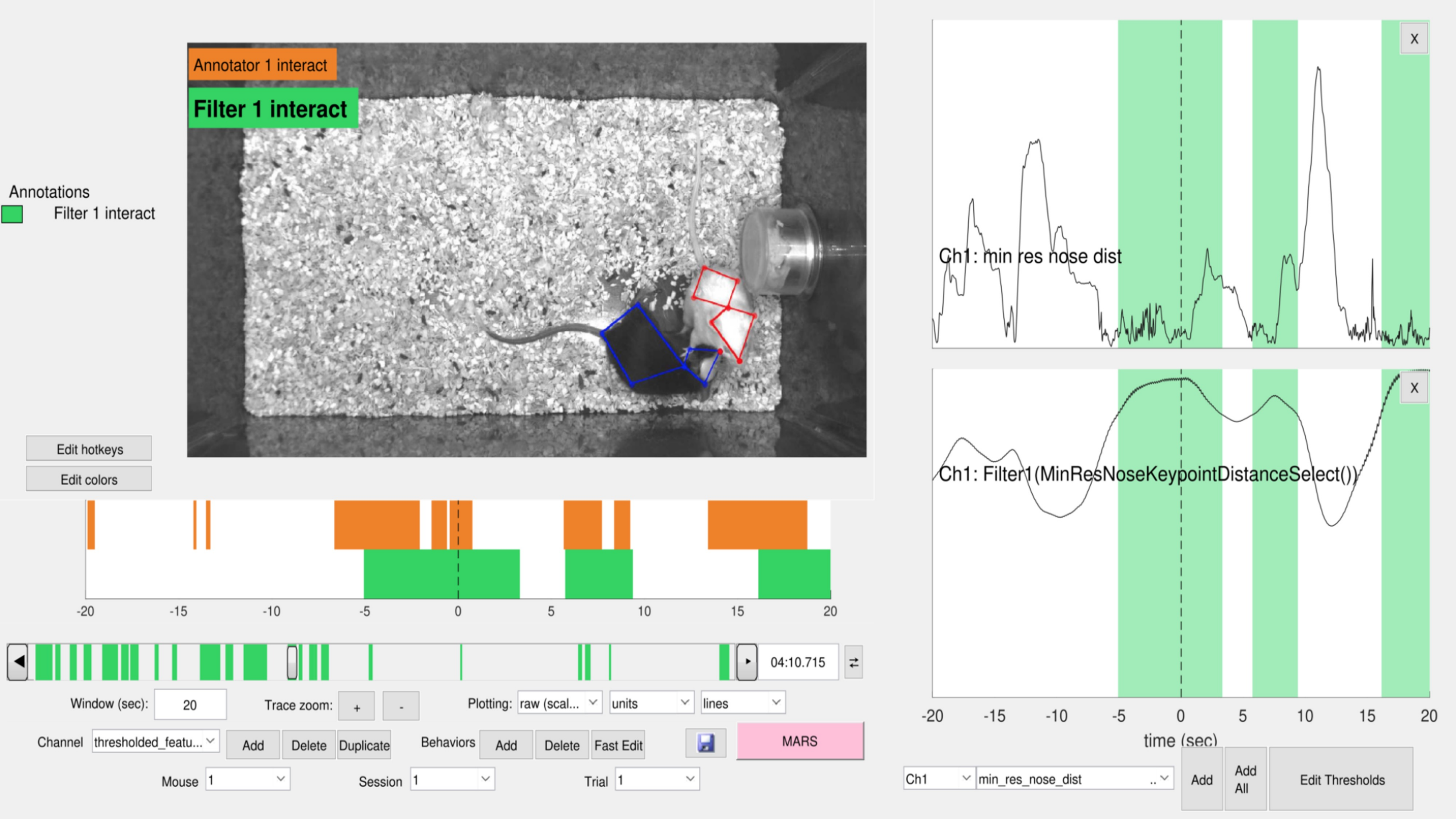
Jennifer Sun





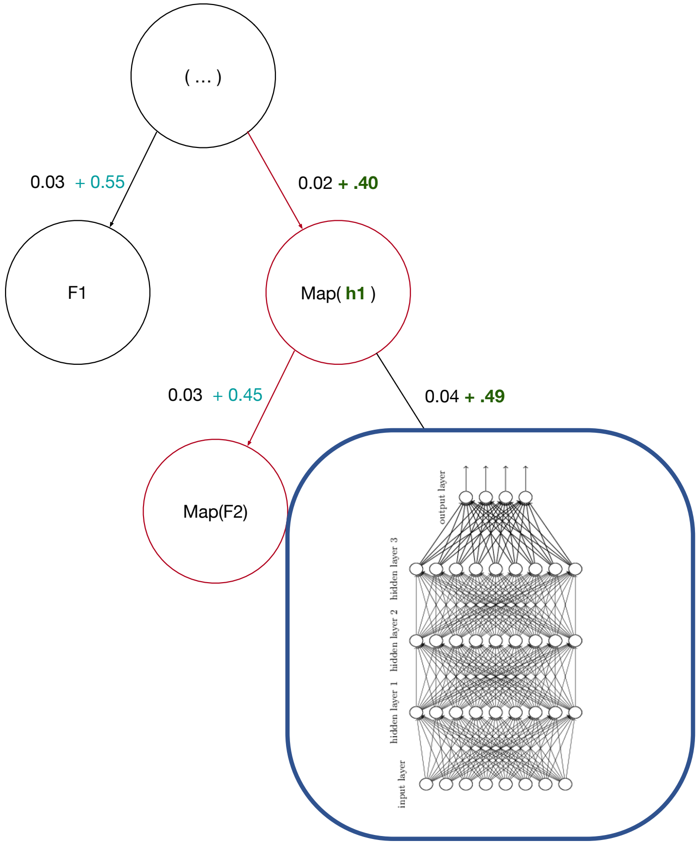
https://arxiv.org/abs/2106.06114

Megan Tjandrasuwita, Jennifer J. Sun, Ann Kennedy, Swarat Chaudhuri, Yisong Yue

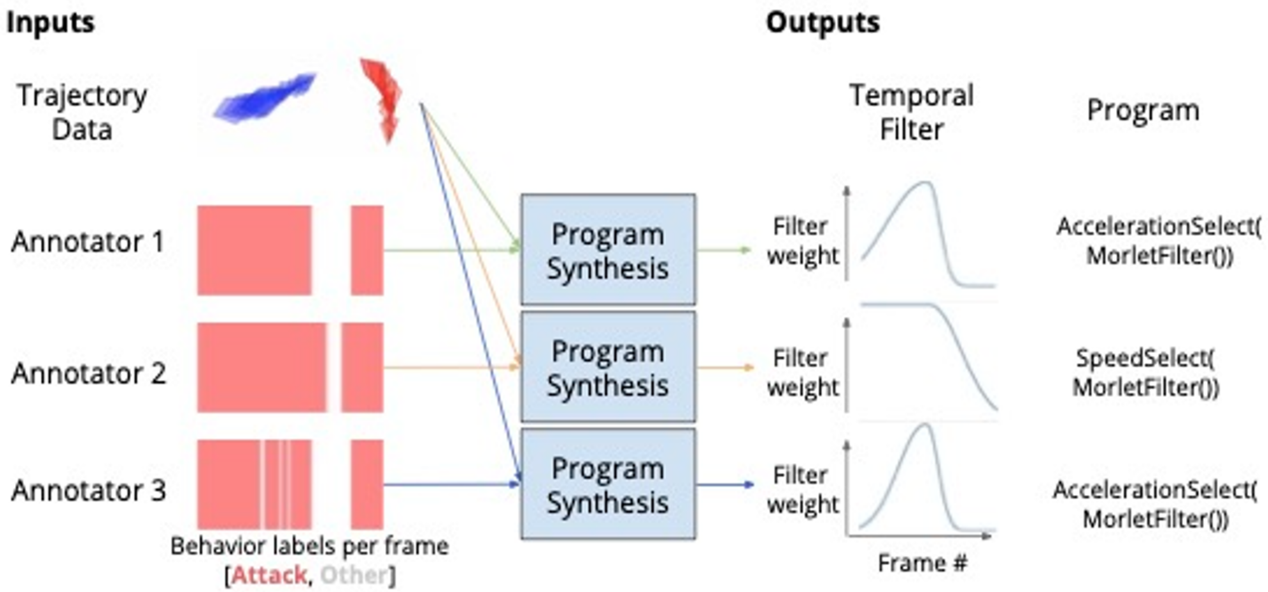


Annotator 1 interact
Filter 1 interact
Annotations
Filter 1 interact
Edit hotkeys
Edit colors
-20 -15 -10 -5 0 5 10 15 20
04:10.715
Window (sec): 20
Trace zoom: + -
Plotting: raw (scal... units lines
Channel thresholded_featu... Add Delete Duplicate
Behaviors Add Delete Fast Edit
MARS
Mouse 1
Session 1
Trial 1
Ch1: min res nose dist
Ch1: Filter1(MinResNoseKeypointDistanceSelect())
X
X
-20 -15 -10 -5 0 5 10 15 20
time (sec)
Ch1 min_res_nose_dist
Add
Add All
Edit Thresholds

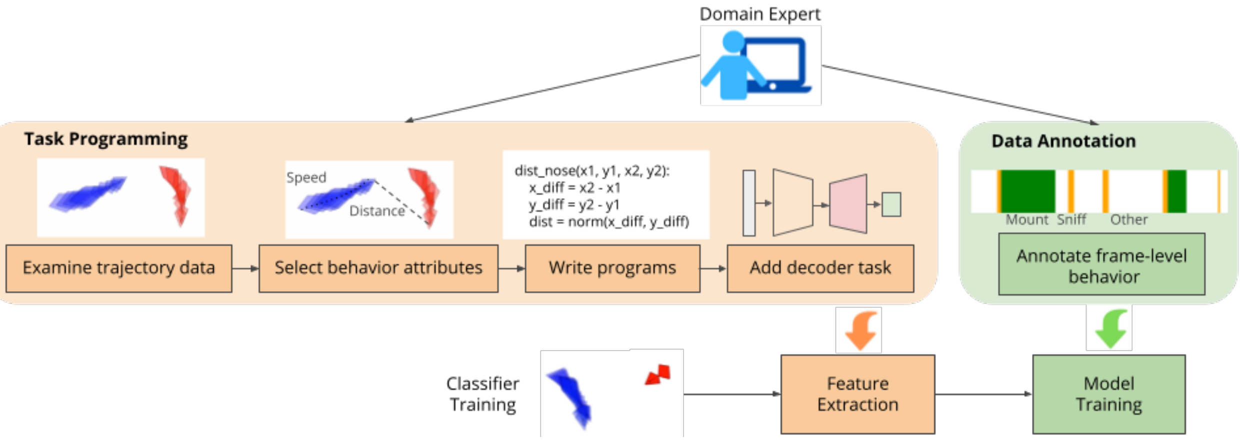
Remainder of Talk



Algorithm Vignette
(Computational Scalability)



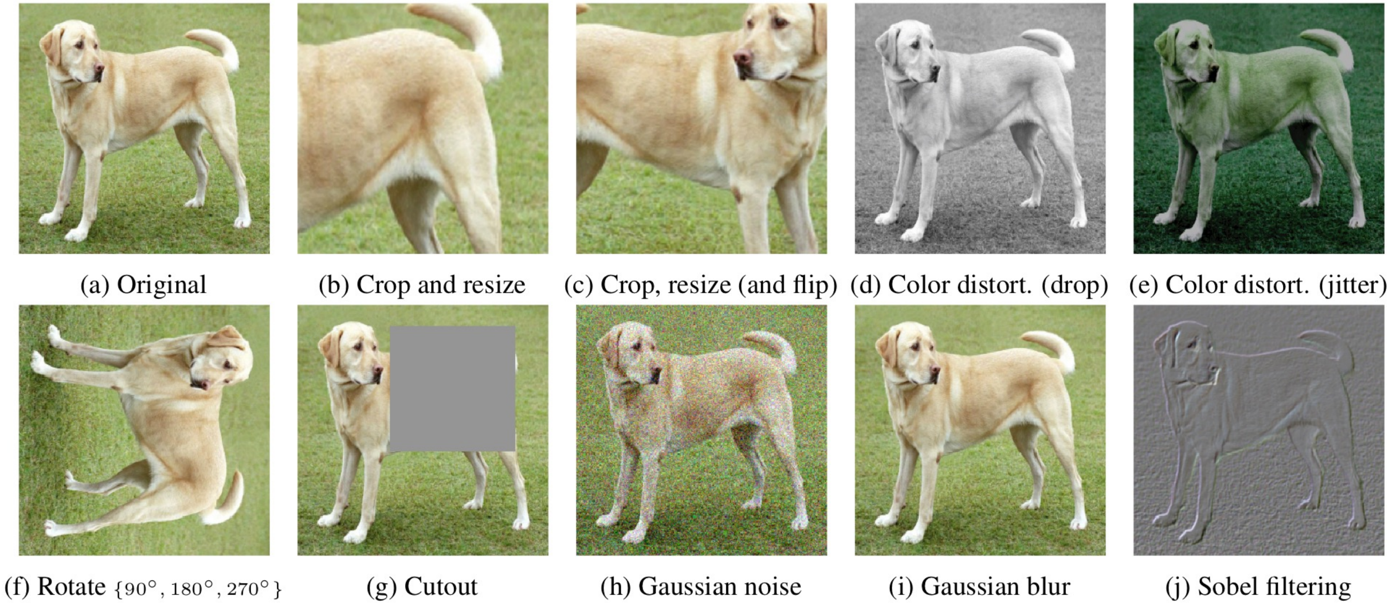
User Study Vignette (Interpretability)



Data Augmentation Vignette (Data Efficiency)

Data Augmentation, Self Supervision, Weak Supervision, etc…

Example: image transformations that preserve "meaning"



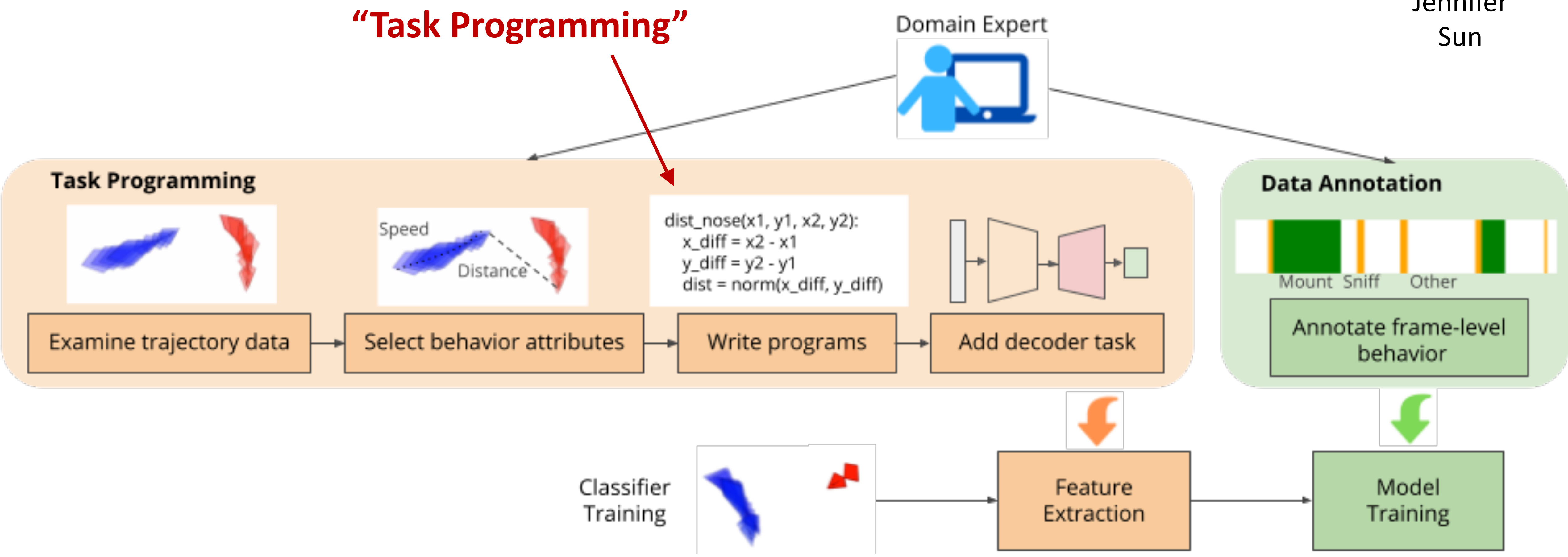
https://arxiv.org/abs/2002.05709

Labeled data is expensive

Use augmentations to reduce label burden

Auxiliary Supervision via Programmed Decoding Tasks

Jennifer Sun

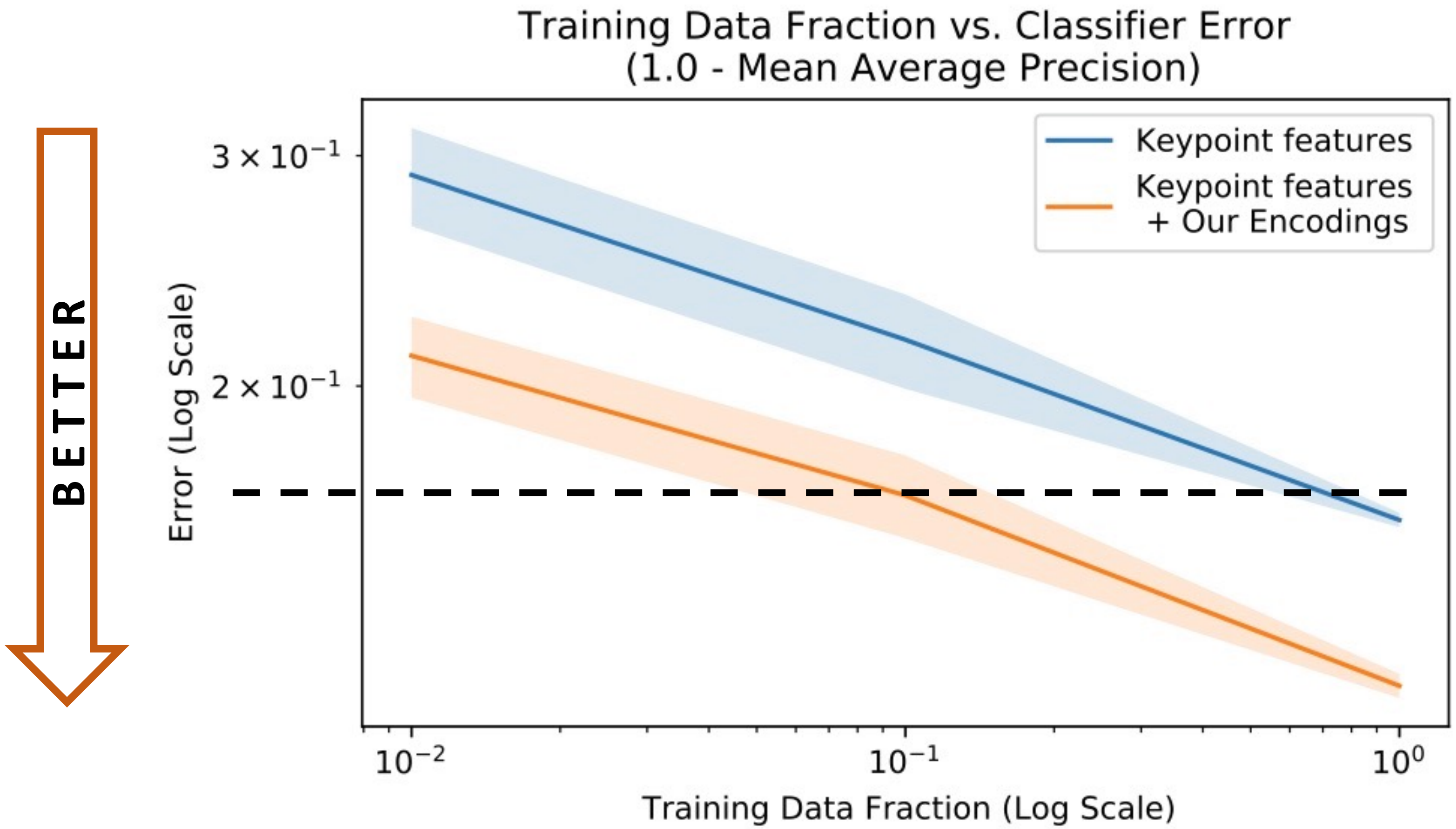


Task Programming: Learning Data Efficient Behavior Representations,
Jennifer J. Sun, Ann Kennedy, Eric Zhan, David J. Anderson, Yisong Yue, Pietro Perona, CVPR 2021 ****Best Student Paper Award*

Task Programming

Follow-up Work:
Automatically Synthesizing Decoding Tasks via Unsupervised Program Learning

Jennifer Sun



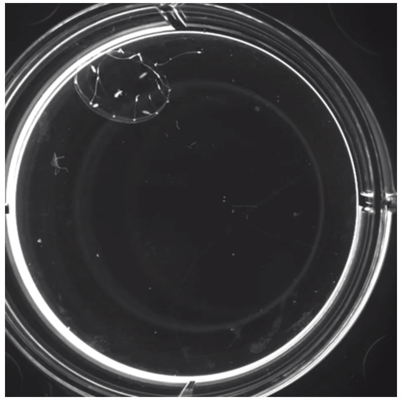
Can lead to 10x annotation efficiency!

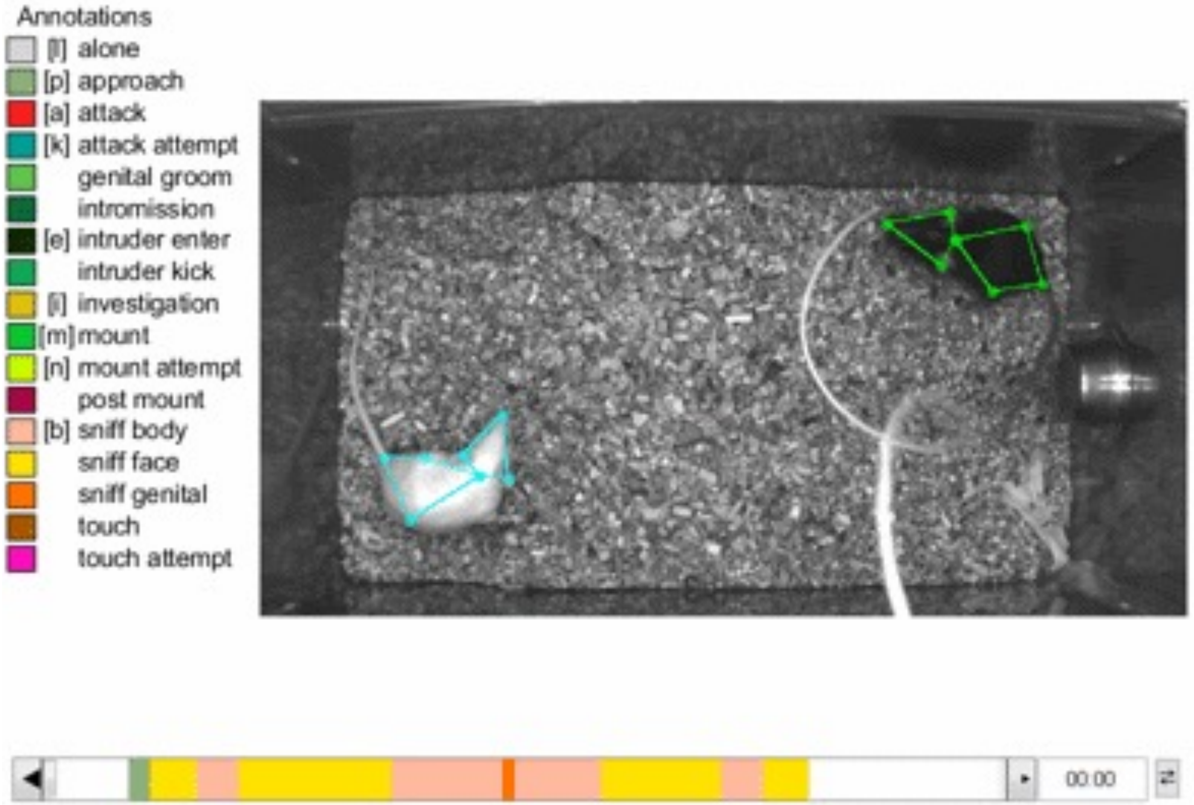
Task Programming: Learning Data Efficient Behavior Representations,
Jennifer J. Sun, Ann Kennedy, Eric Zhan, David J. Anderson, Yisong Yue, Pietro Perona, CVPR 2021 ****Best Student Paper Award*

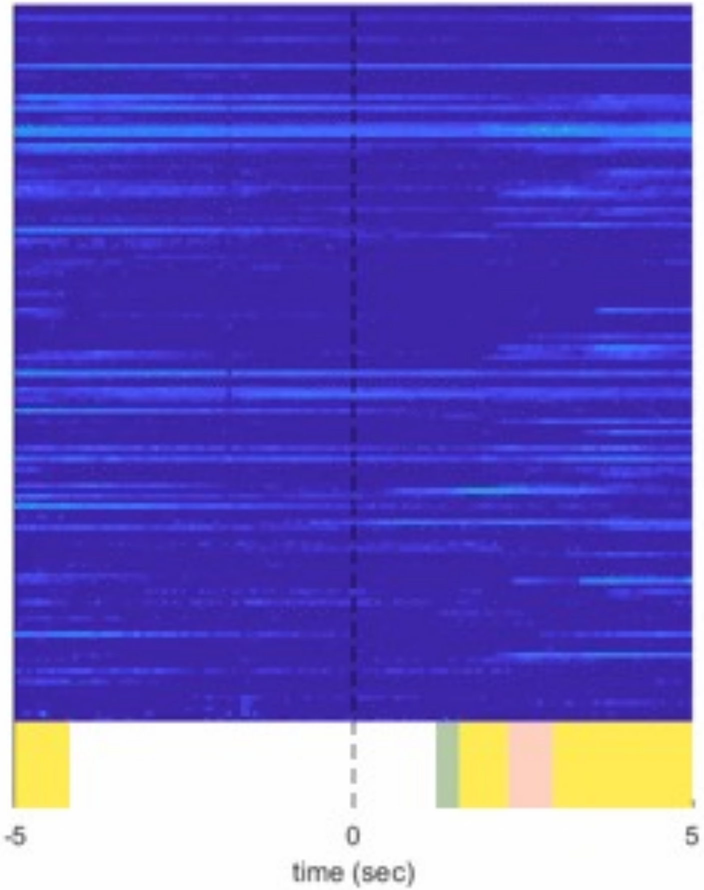
Close Collaboration with Domain Experts

David Anderson
(Caltech)

Ann Kennedy
(Northwestern)







Neurosymbolic Survey






*****Coming out soon!**

Neurosymbolic Programming

Swarat Chaudhuri
UT Austin
swarat@cs.utexas.edu

Kevin Ellis
Cornell University
kellis@cornell.edu

Oleksandr Polozov
Microsoft Research
polozov@microsoft.com

Rishabh Singh
Google
rising@google.com

Armando Solar-Lezama
MIT
asolar@csail.mit.edu

Yisong Yue
Caltech
yyue@caltech.edu

ABSTRACT

Neurosymbolic programming is an emerging research area at the interface of deep learning and program synthesis. Like in classic machine learning, the goal here is to learn functions from data. However, these functions are represented as *programs* that use symbolic primitives, often in conjunction with neural network components, and must, in some cases, satisfy certain additional behavioral constraints. The programs are induced using a combination of symbolic search and gradient-based optimization.

Neurosymbolic programming can offer multiple advantages over end-to-end deep learning. Programs can sometimes naturally represent long-horizon, procedural tasks that are difficult to perform using deep networks. Neurosymbolic representations are also, commonly, easier to interpret, analyze, and trust than neural networks.

idea here is to represent ML models as *programs* of the sort humans would write. Sometimes, these programs are built entirely from symbolic primitives. Sometimes, they use a mix of symbolic code and neural modules. The learning problem in this area is to *simultaneously induce* a program's symbolic and neural components, and this problem is solved using a mix of symbolic and statistical techniques. We call this literature *neurosymbolic programming* (Nsp), and this article is an introduction to this area.

Mathematically, a program α_θ in Nsp consists of a *program architecture* α that defines the way in which a program's symbolic and neural modules are composed, and a vector θ of parameters of these modules. The program architectures are required to follow the syntax of a *domain-specific language* (DSL). The learning problem is to induce α and θ so as to minimize an empirical loss function $L(\alpha, \theta)$.

Neurosymbolic Programs

http://www.neurosymbolic.org/

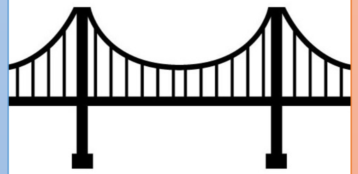
Symbolic Programs

Interpretable

Verifiable

Structured domain knowledge

Data efficient



Neural Networks

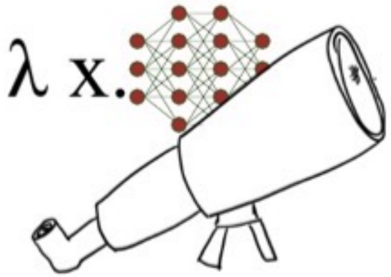
Scalable algorithms

Flexible

Handles messy data

Easy to get started

Thanks!



References:

Learning Differentiable Programs with Admissible Neural Heuristics, Ameesh Shah*, Eric Zhan*, et al., NeurIPS 2020

- https://github.com/trishullab/near

Interpreting Expert Annotation Differences in Animal Behavior, Megan Tjandrasuwita et al., arXiv

Task Programming: Learning Data Efficient Behavior Representations, Jennifer J. Sun, et al., CVPR 2021 ****Best Student Paper Award*

- https://sites.google.com/view/task-programming

Unsupervised Learning of Neurosymbolic Encoders, Eric Zhan*, Jennifer J. Sun*, et al., arXiv