Caltech

# Improving Policy Learning via Programmatic Domain Knowledge

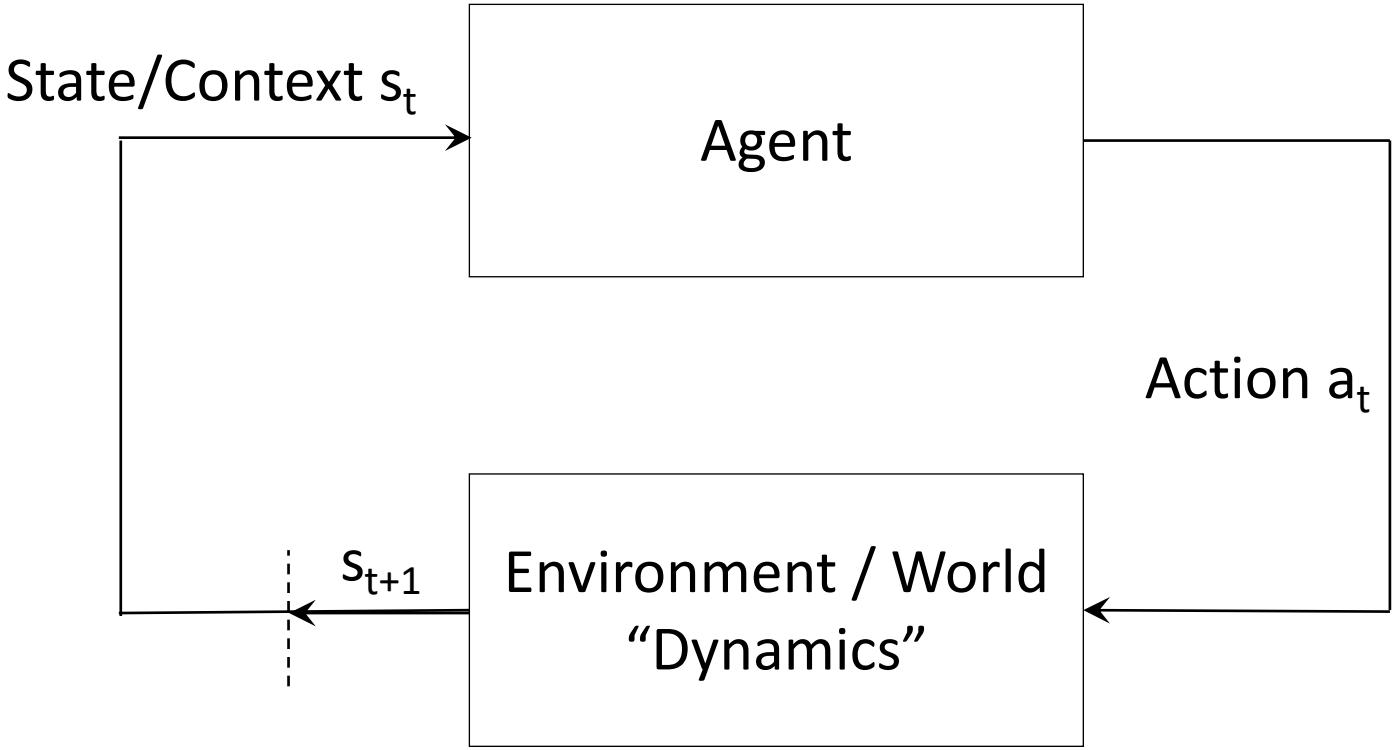
Yisong Yue

# Policy/Controller Learning (Reinforcement & Imitation)

**Goal:** Find "Optimal" Policy

**Imitation Learning:**
Optimize imitation loss

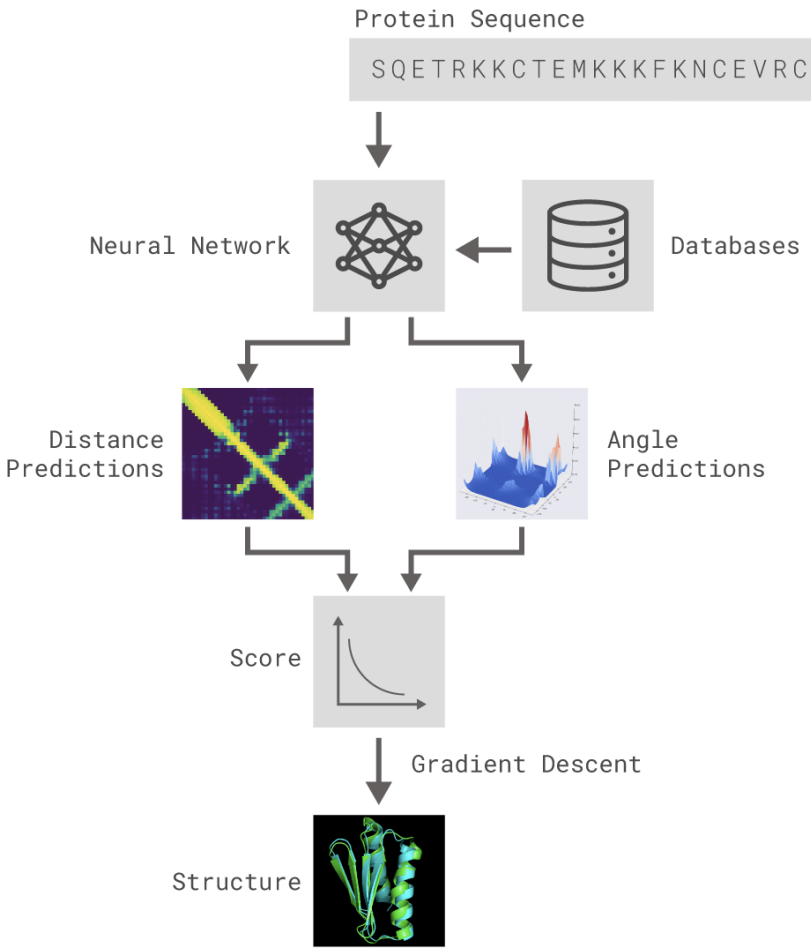
**Reinforcement Learning:**
Optimize environmental reward



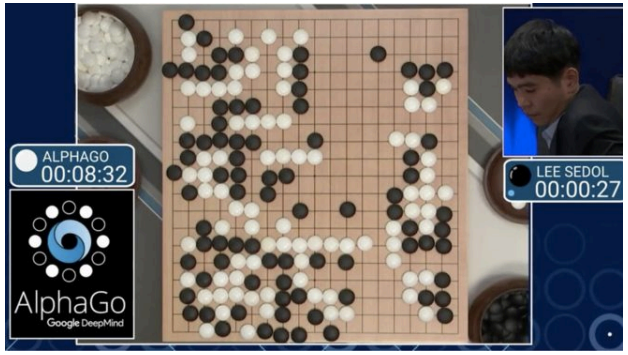

**Learning-based Approach for Sequential Decision Making**

Non-learning approaches include: optimal control, robust control, adaptive control, etc.

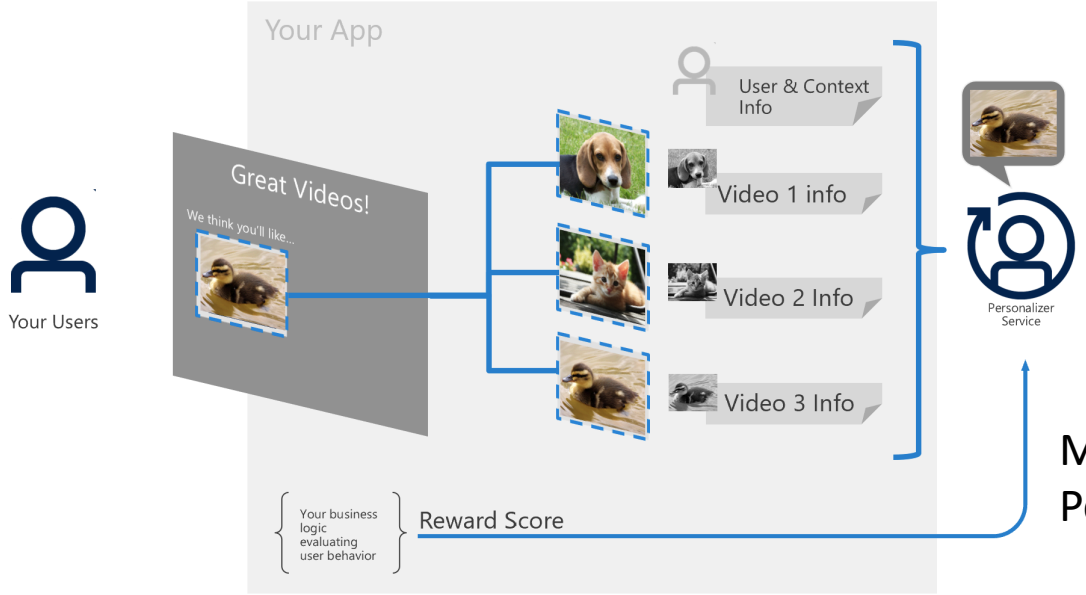
# Many Exciting Success Stories




AlphaFold





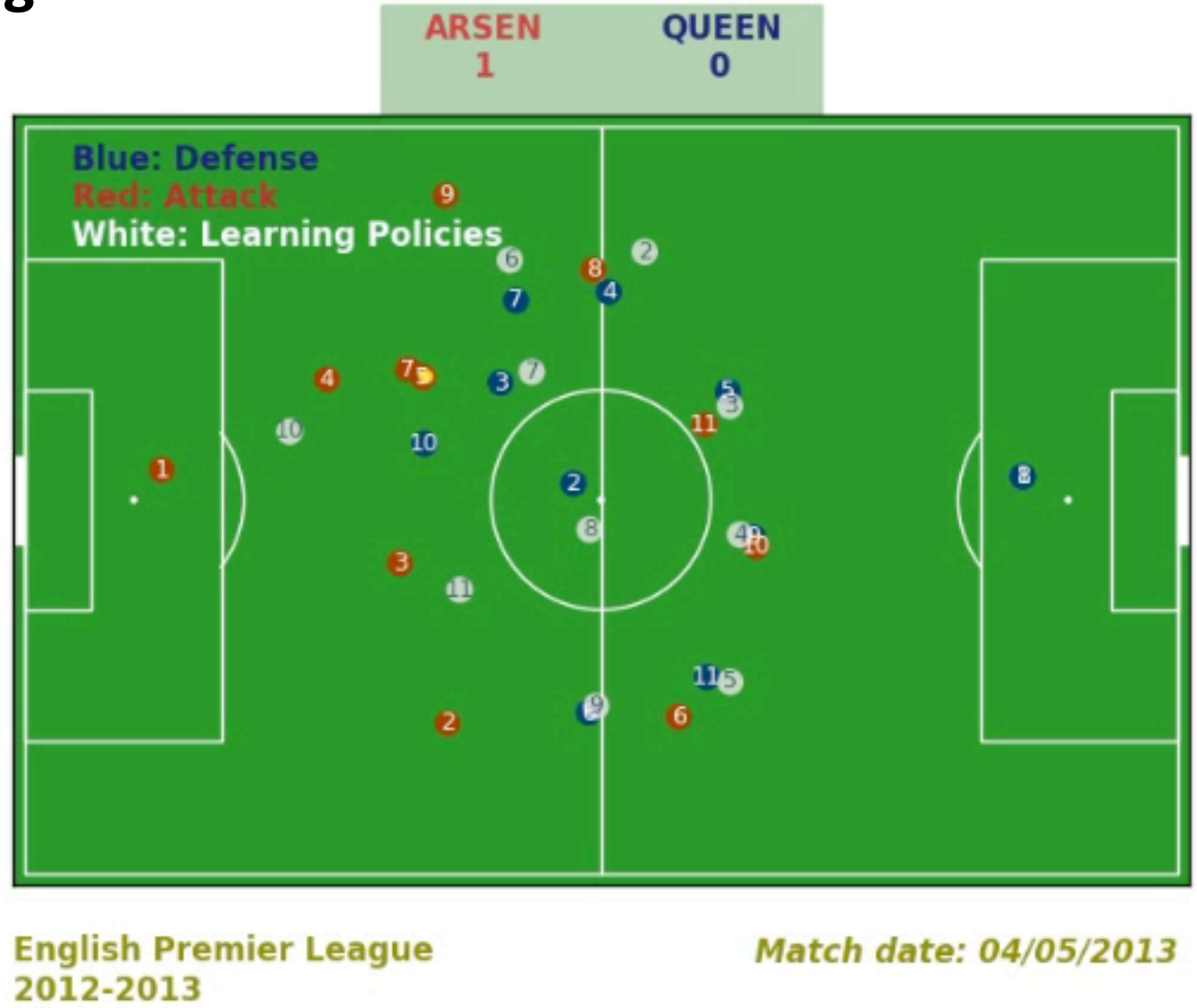



Microsoft Azure Personalizer

https://www.youtube.com/watch?v=9zL7qejW9fE

German

# Data-Driven Ghosting




Hoang Le

**Data-Driven Ghosting using Deep Imitation Learning**
Hoang Le et al. SSAC 2017

https://www.youtube.com/watch?v=WI-WL2cj0CA

# Imitation Learning Tutorial

https://sites.google.com/view/icml2018-imitation-learning/


**Yisong Yue**

yyue@caltech.edu

@YisongYue

yisongyue.com

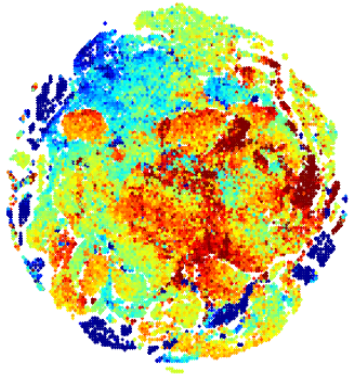
**Hoang M. Le**

Hoang.Le@Microsoft.com

@HoangMinhLe

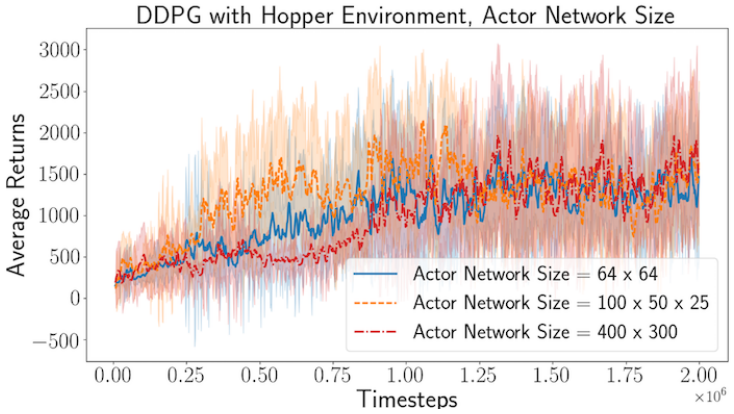
hoangle.info

# Challenges with “Black Box” Policy Learning

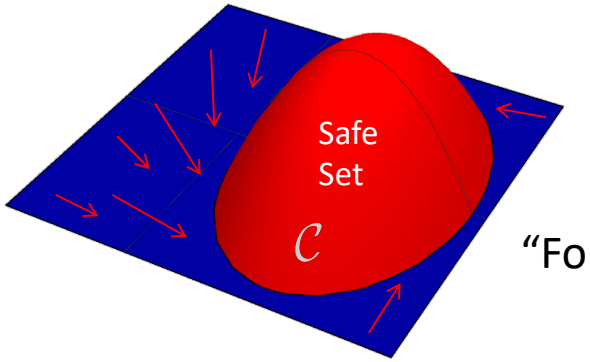
https://arxiv.org/abs/1611.00094

Interpretability

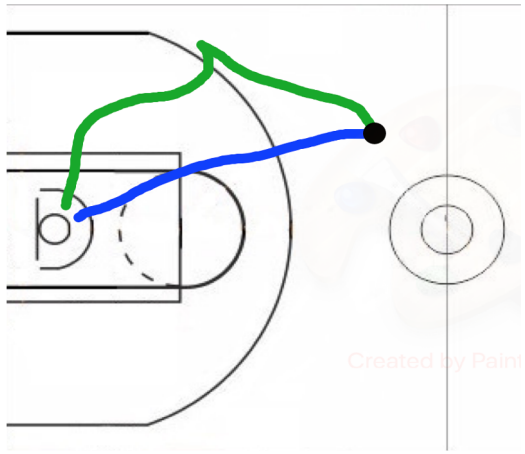



Sample Efficiency

https://arxiv.org/abs/1709.06560




“Formal” Interpretability



Controllability

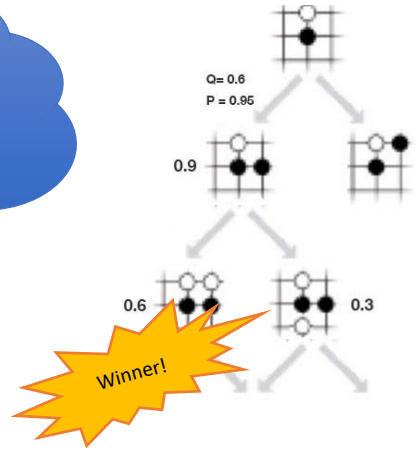
https://arxiv.org/abs/1910.01179

***Modulo Optimization Concerns

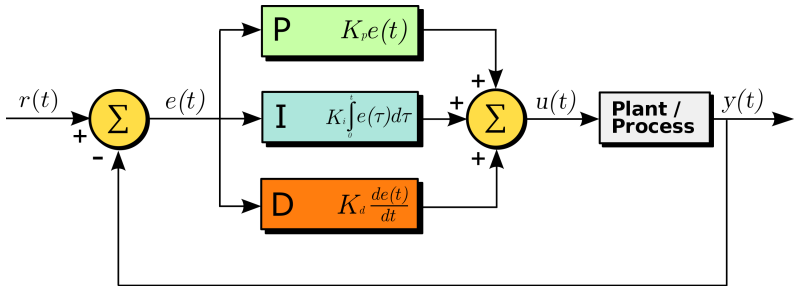
# Data vs Priors

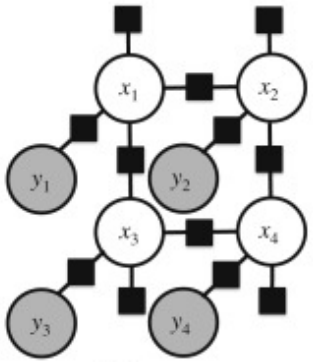
Raw Data

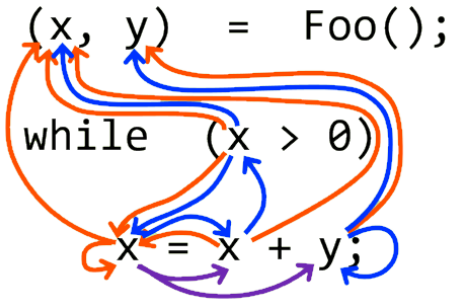
https://medium.com/@jonathan_hui/monte-carlo-tree-search-mcts-in-alphago-zero-8a403588276a



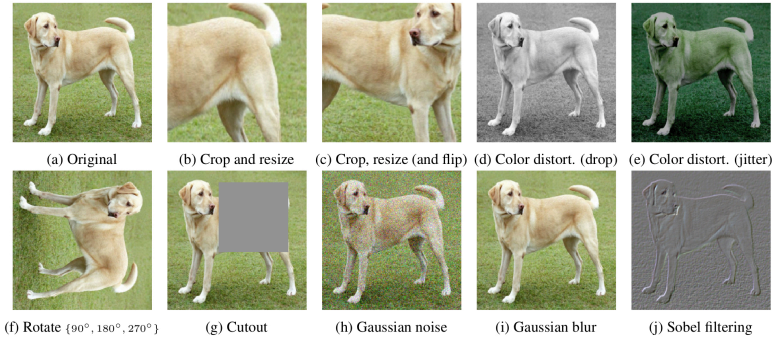

Environmental Reward



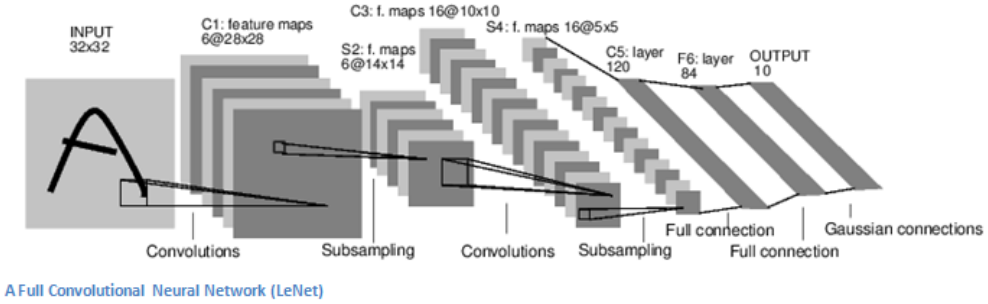



https://alexpolozov.com/blog/program-synthesis-2018/




Data Augmentation

https://arxiv.org/abs/2002.05709




Inductive Biases (“Priors”)

# Programmatic Domain Knowledge

**aka Symbolic**

- Domain Specific Language (DSL)

$$\begin{aligned} \pi(s) &::= a \mid Op(\pi_1(s), \ldots, \pi_k(s)) \mid \textbf{if } b \textbf{ then } \pi_1(s) \textbf{ else } \pi_2(s) \mid \oplus_\theta(\pi_1(s), \ldots, \pi_k(s)) \\ b &::= \phi(s) \mid BOp(b_1, \ldots, b_k) \end{aligned}$$

https://arxiv.org/abs/1907.05431

$$\textbf{if } (s[\texttt{TrackPos}] < 0.011 \textbf{ and } s[\texttt{TrackPos}] > -0.011)$$
$$\textbf{then } \texttt{PID}_{\langle \texttt{RPM}, 0.45, 3.54, 0.03, 53.39 \rangle}(s) \textbf{ else } \texttt{PID}_{\langle \texttt{RPM}, 0.39, 3.54, 0.03, 53.39 \rangle}(s)$$

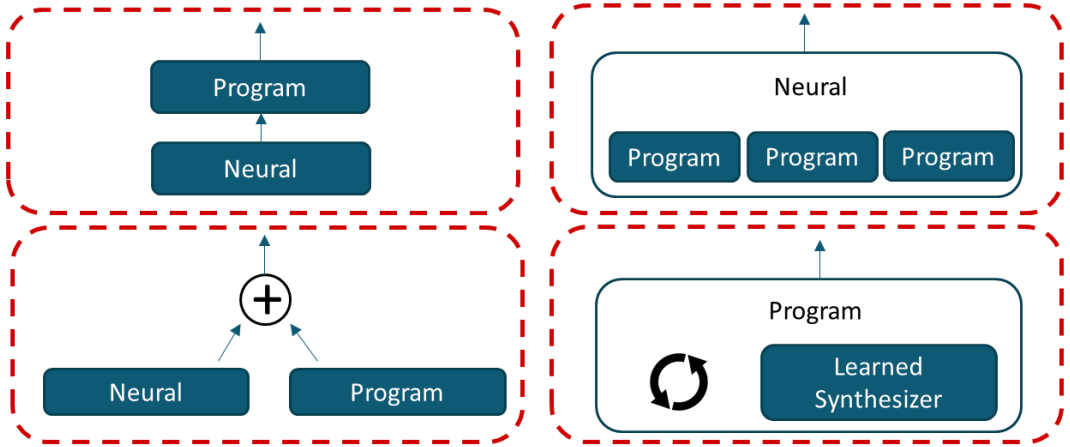
$$\begin{aligned} \alpha ::= \; & x \mid c \mid \oplus(\alpha_1, \ldots, \alpha_k) \mid \oplus_\theta(\alpha_1, \ldots, \alpha_k) \mid \textbf{if } \alpha_1 \textbf{ then } \alpha_2 \textbf{ else } \alpha_3 \mid \textbf{sel}_S \; x \\ & \textbf{map } (\lambda x_1.\alpha_1) \; x \mid \textbf{fold } (\lambda x_1.\alpha_1) \; c \; x \mid \textbf{mapprefix } (\lambda x_1.\alpha_1) \; x \end{aligned}$$

https://arxiv.org/abs/2007.121

$$\textbf{map}(\textbf{multiply}(\textbf{add}(\mathit{OffenseAffine}(x), \mathit{BallAffine}(x)), \textbf{add}(\mathit{OffenseAffine}(x), \mathit{BallAffine}(x)))$$

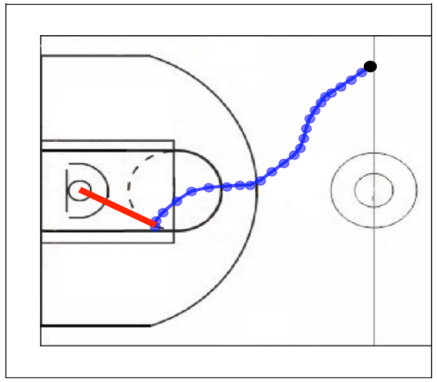
# Programmatic Domain Knowledge

- Programmatic Priors

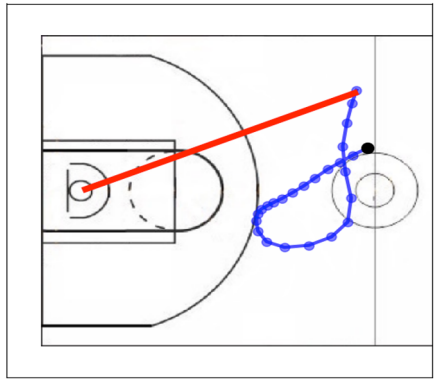



"Neurosymbolic" learning

- Programmatic Supervision



Small Distance



Large Distance

"Data Programming" for sequential behavioral data

https://arxiv.org/abs/1605.07723

# Program Learning

**Loss Function**

Reinforcement
Imitation
Etc…

$$\pi^* = \underset{\pi \in \Pi}{\operatorname{argmin}}\, J(\pi)$$

**Programmatic Policy Class**

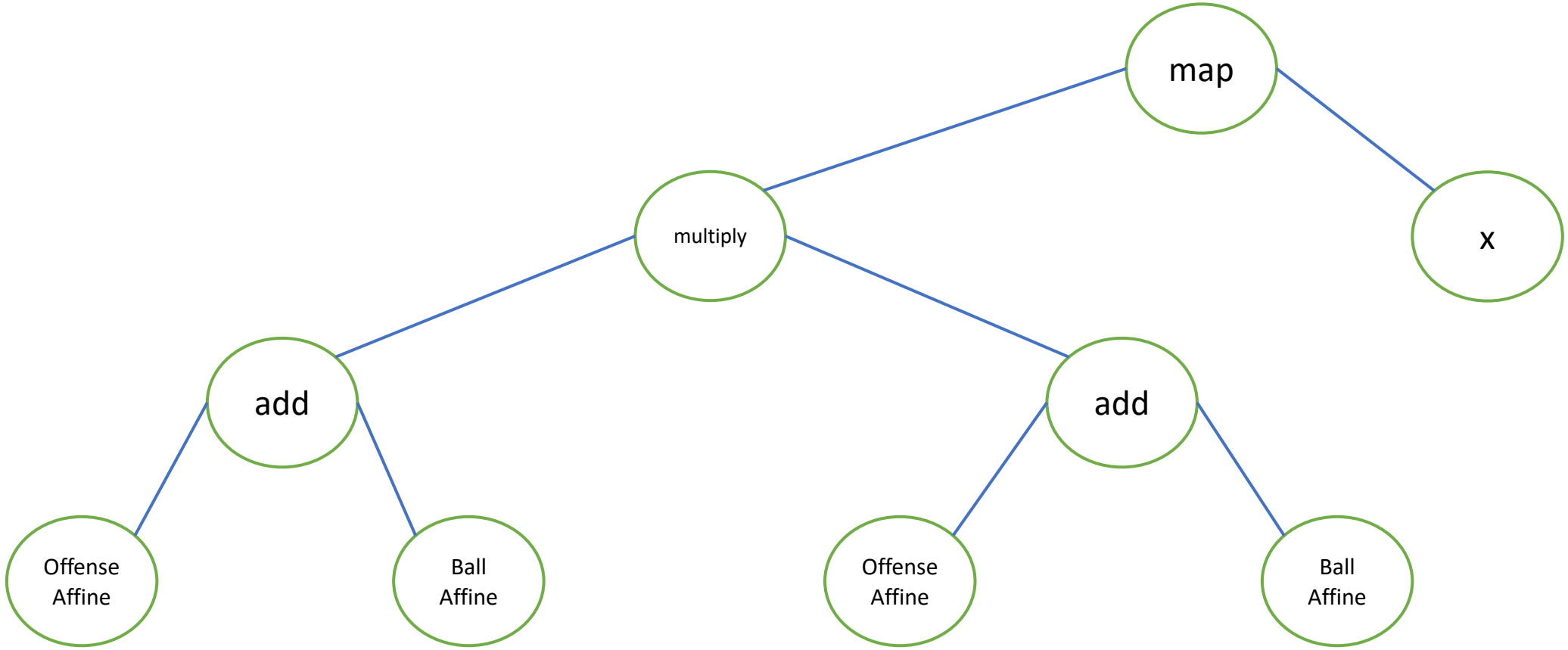
**Challenges**

- Policy class has discrete structures
- Fully differentiable learning may not be possible

***Similar challenges in Neural Architecture Search

# Program "Architecture"

$$\mathbf{map}(\mathbf{multiply}(\mathbf{add}(\mathit{OffenseAffine}(x), \mathit{BallAffine}(x)), \mathbf{add}(\mathit{OffenseAffine}(x), \mathit{BallAffine}(x)))$$



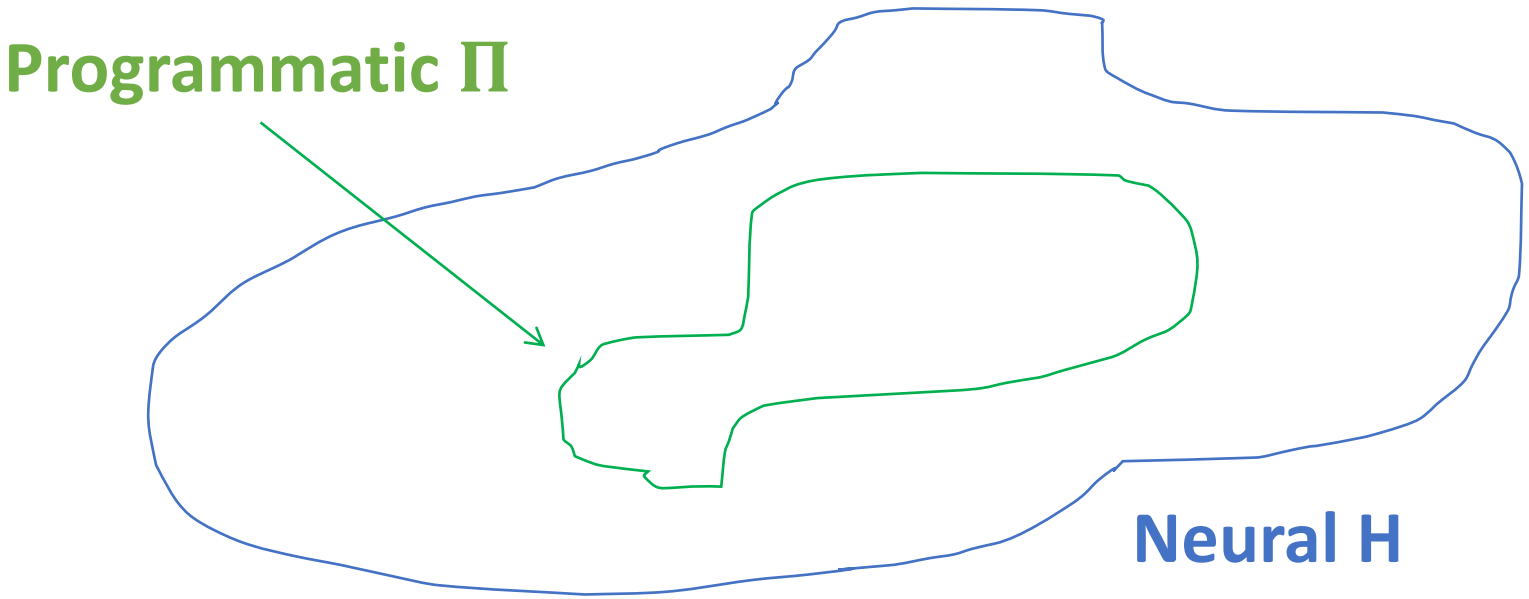
# Discrete Search for Program Synthesis

- A* Search
- Branch and Bound
- Local Pruning
- Transformations (e.g., simplify programs, identify isomorphisms, etc.)
- Oracle Calls to Satisfiability Sub-Problems
- Reinforcement Learning
- Etc…

**Our Philosophy:**

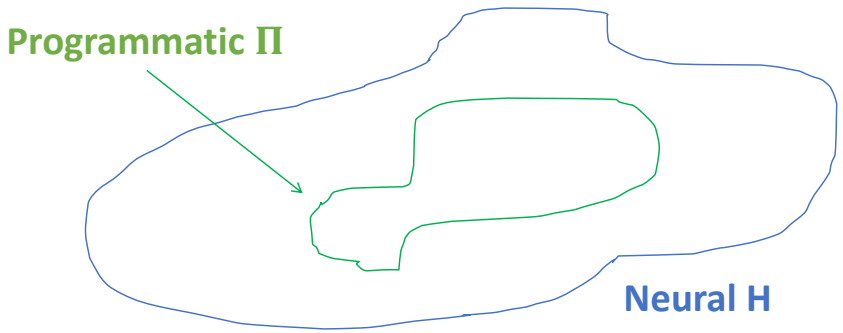
- Compatible with all of the above
- Maximally leverages deep learning
- General framing of problem setting

# Motivating Observation/Assumption:
# Functional Representational Power




**"Neural Relaxation"** Every programmatic policy can be (approximately) represented by some neural policy.

# Implication
## (abstract form)




Slack due to approximation error or training ability

Neural

$$\forall \pi \in \Pi, \exists h \in H \text{ s.t. } J(h) \leq J(\pi) + \epsilon$$

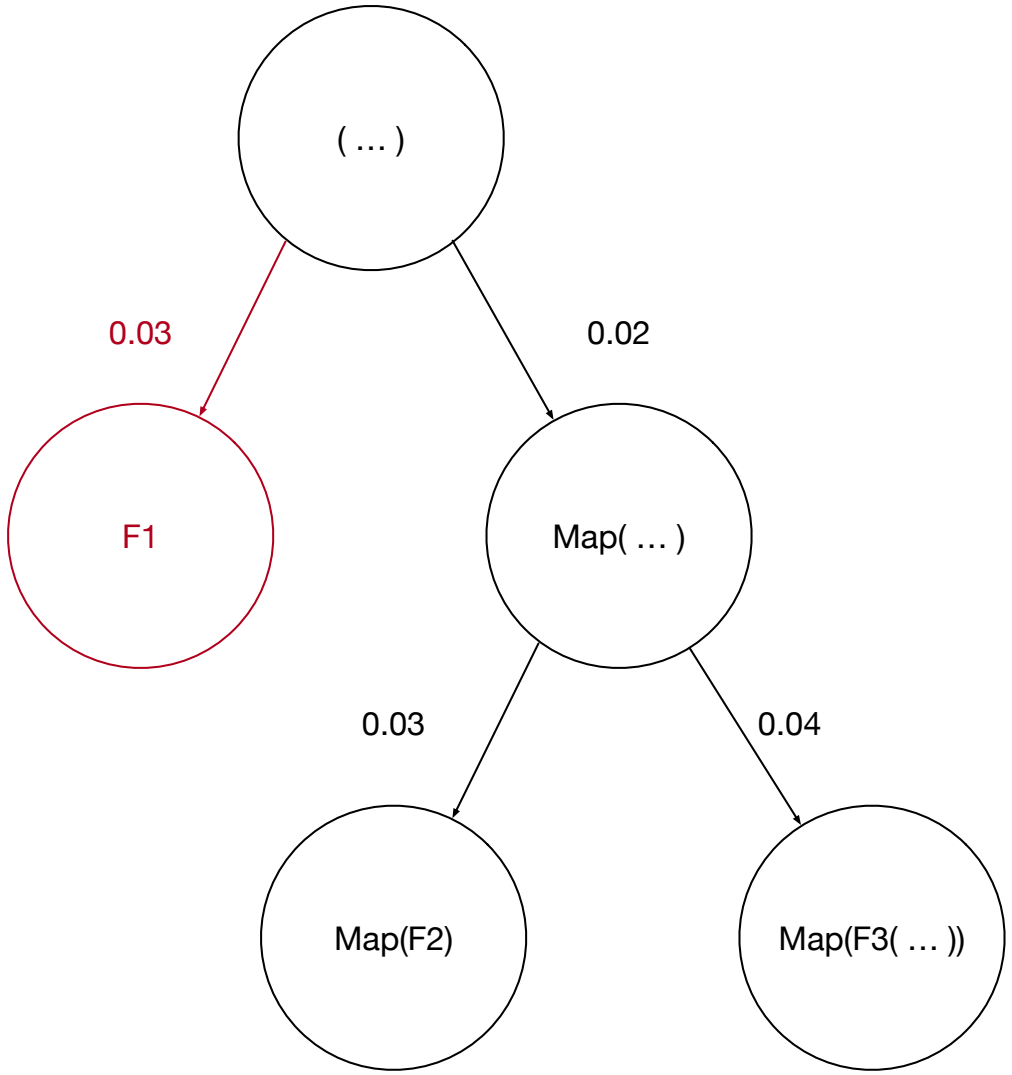
Programmatic

Loss Function
(Classification/Regression/Imitation/Reinforcement/etc.)

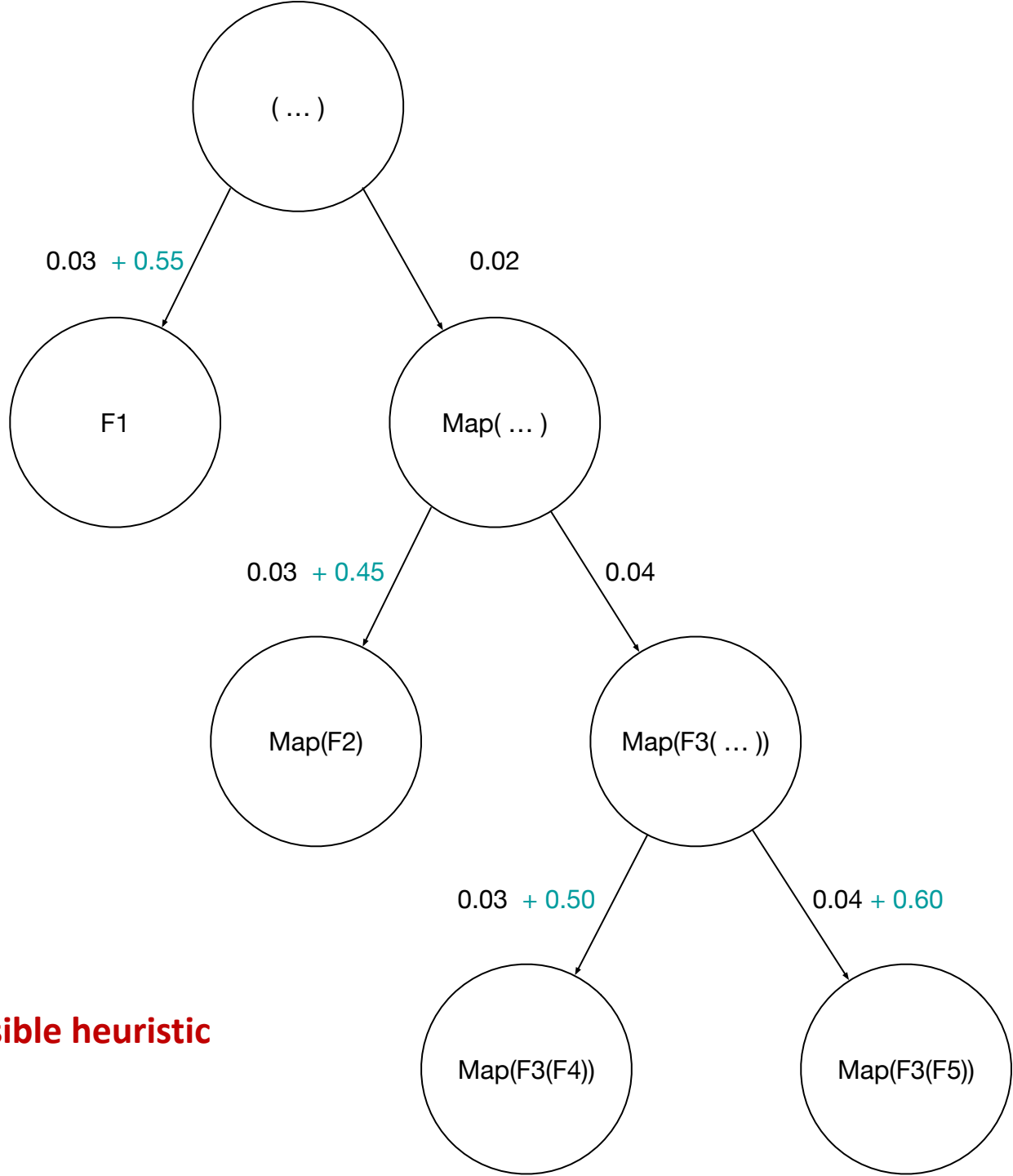
**"Neural Relaxation"** Every programmatic policy can be (approximately) represented by some neural policy.

# Setting 1: Top-Down Program Induction

- Nodes: (Partial) Programs
  - Goal nodes (sinks): complete programs
- Edges: Single expansion of program
- Edge costs: $s(v_{child}) - s(v_{parent})$
  - Cost of expansion
  - I.e., complexity measure



# Setting 1: Top-Down Program Induction



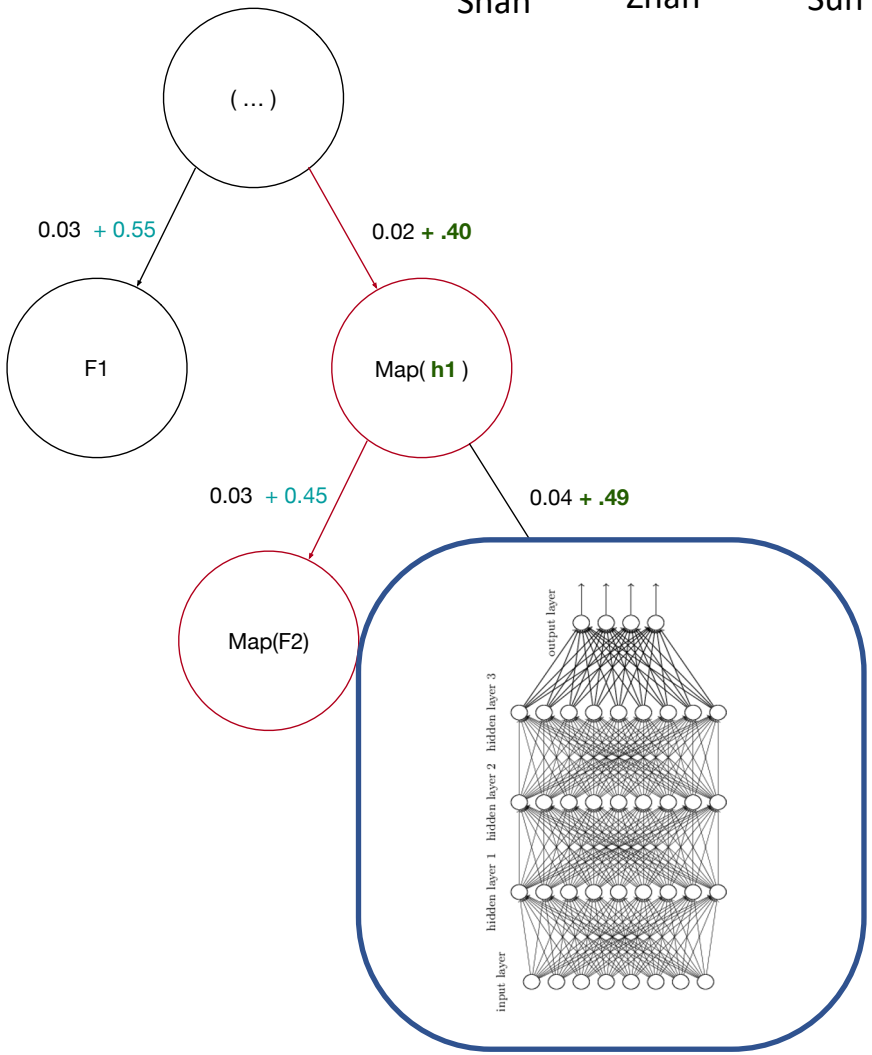

- Training Loss included in edge costs to sink nodes (complete programs)
- Optimal search requires a complete traversal of the graph

**Often exponentially large!**
**Popular approaches (e.g., A*) require admissible heuristic**

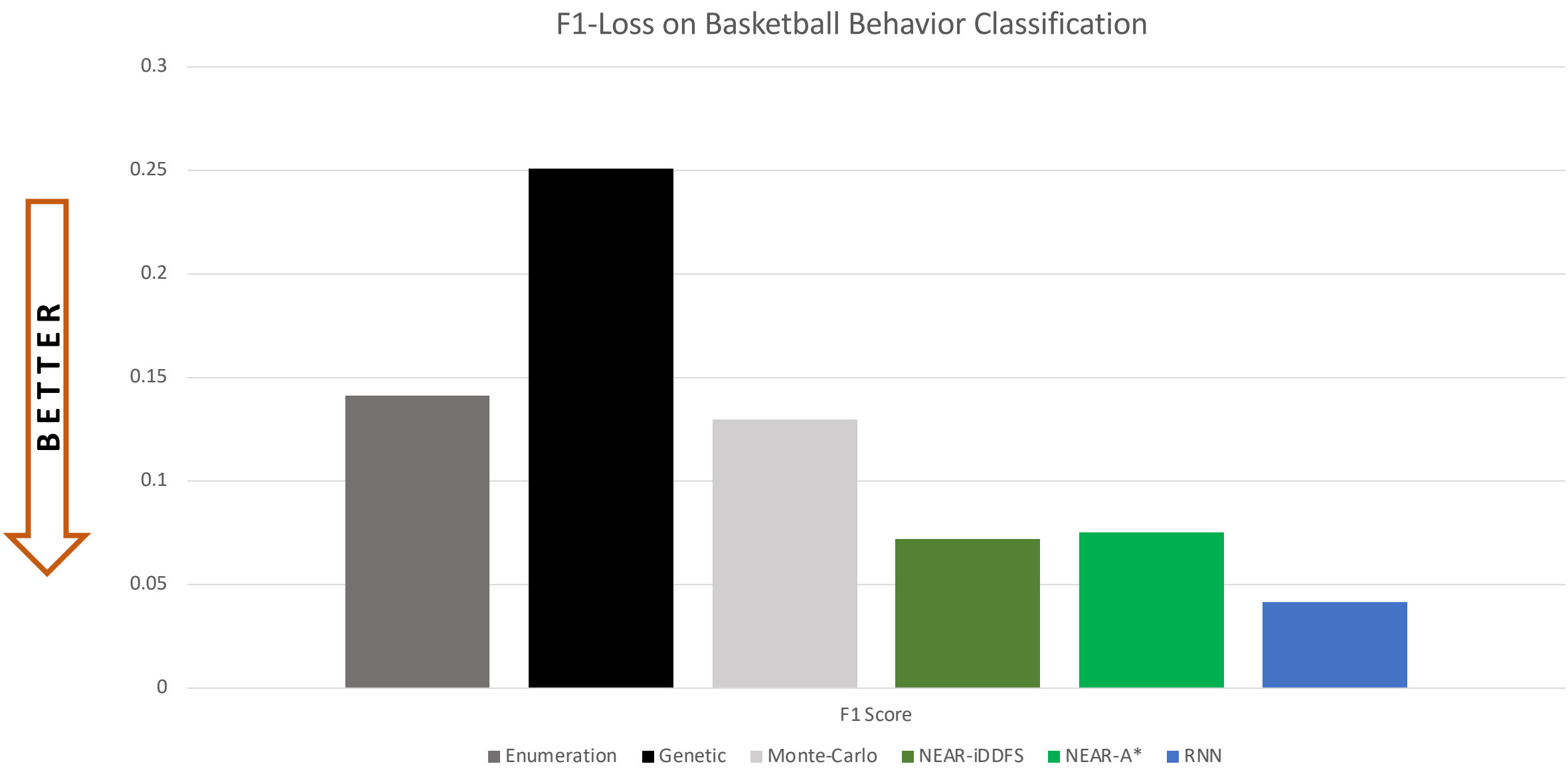
# NEAR: Neural Admissible Relaxations

Ameesh Shah
Eric Zhan
Jennifer Sun

- Neural completion of partial program
  - Sub-routine is a “black-box” neural net
  - “Neural Relaxation”
- Can evaluate complete programs
  - Lower loss: $J(best\ neural) \leq J(best\ program) + \epsilon$
  - **Exactly an $\epsilon$-Admissible Heuristic!**
- A* is guaranteed to find $\epsilon$-optimal program!
  - Also applies to Branch & Bound, etc.



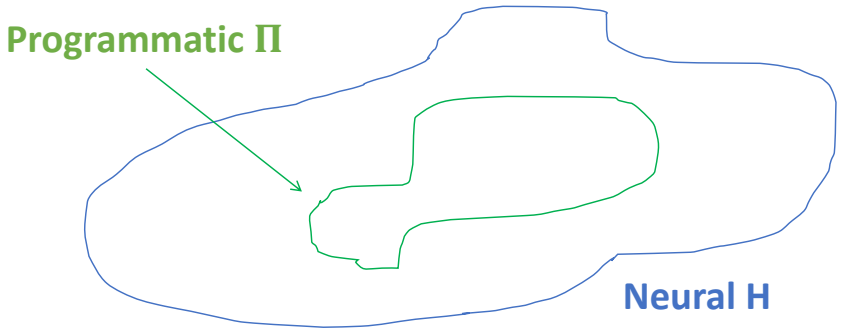

**Learning Differentiable Programs with Admissible Neural Heuristics,** Ameesh Shah, Eric Zhan, et al., NeurIPS 2020



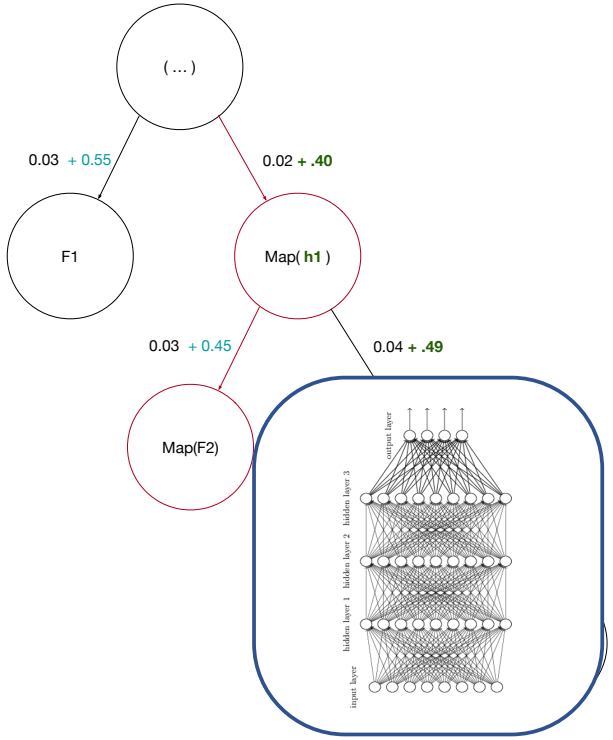

**Learning Differentiable Programs with Admissible Neural Heuristics,** Ameesh Shah, Eric Zhan, et al., NeurIPS 2020

# Story So Far

- Programmatic Policy Class (Inductive Bias)



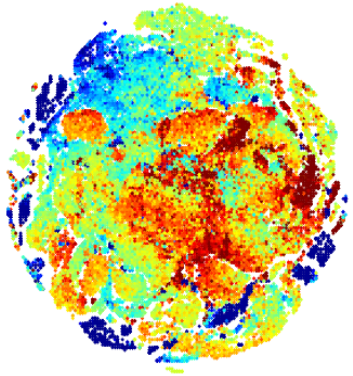

- Neural Relaxations of Programs




- Admissible Heuristic for Top-Down Induction

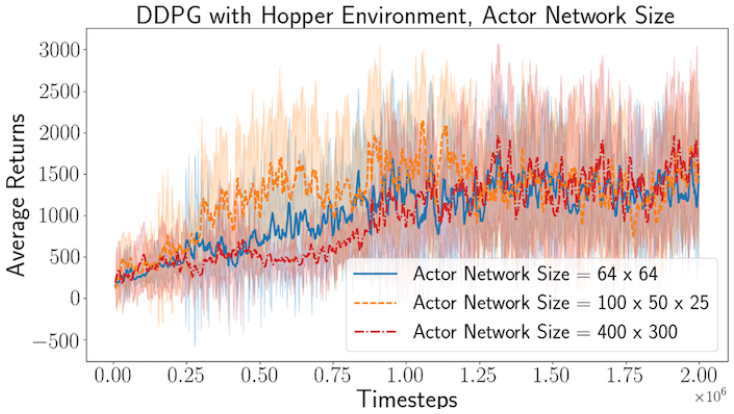
- What's Next?

# Recall: Challenges with “Black Box” Policy Learning

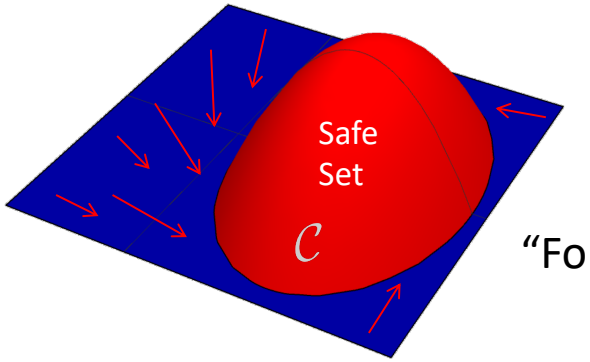
https://arxiv.org/abs/1611.00094

Interpretability

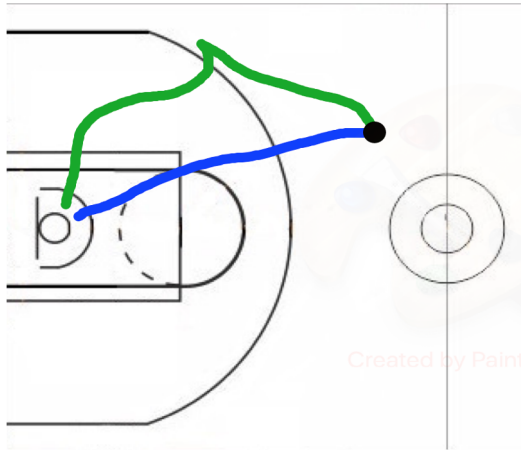



Sample Efficiency

https://arxiv.org/abs/1709.06560




“Formal” Interpretability



Controllability

https://arxiv.org/abs/1910.01179

# Sample-Efficient Policy Learning

- Deep Reinforcement Learning is notoriously data hungry…

See: https://arxiv.org/abs/1709.06560

- …but has been successful (in large part) due to combining stochastic policies with differentiable learning.

- **How to enable this generally w/ programs?**

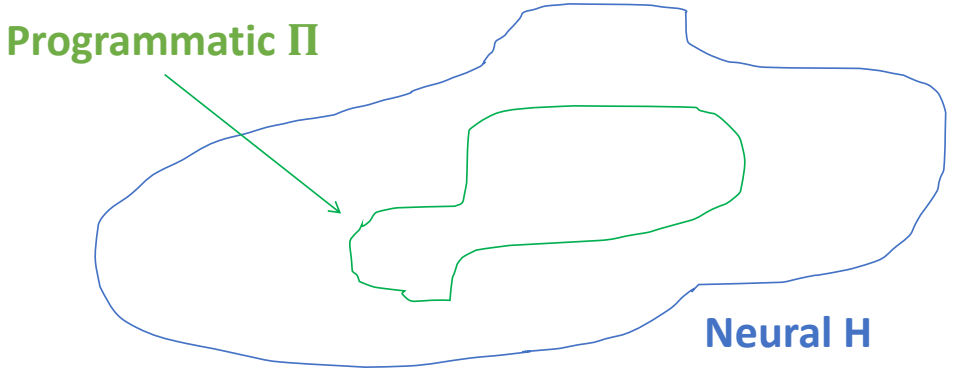
# Setting 2: Program Learning as Constrained Mirror Descent

Abhinav Verma

Hoang Le

$$\pi^* = \underset{\pi \in \Pi}{\operatorname{argmin}}\, J(\pi)$$

**$\Pi$ is a constrained (approximate) subset of $H$!**



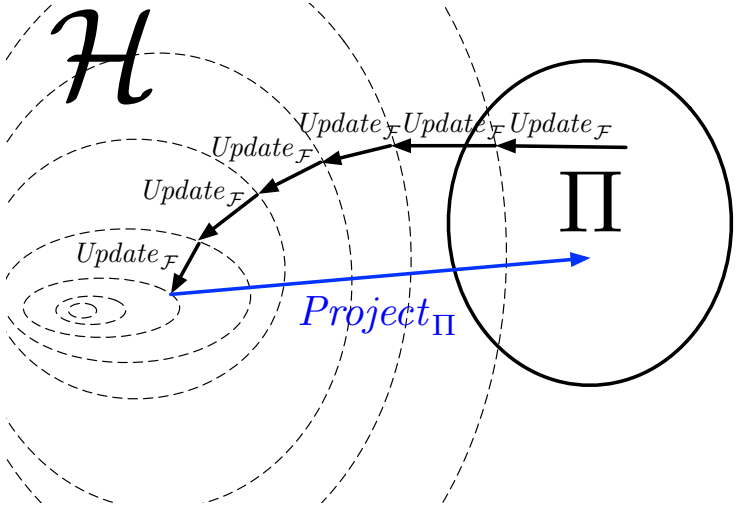

**Imitation-Projected Programmatic Reinforcement Learning,** Abhinav Verma, Hoang Le, et al., NeurIPS 2019

# Constrained Mirror Descent Recipe

- **Update:** gradient-based updates in unconstrained space H
  - "Trust Region" defined by divergence function (e.g., L2, KL, etc.)

- **Project:** project onto constrained space $\Pi$

- **Repeat**



# Useful Notation

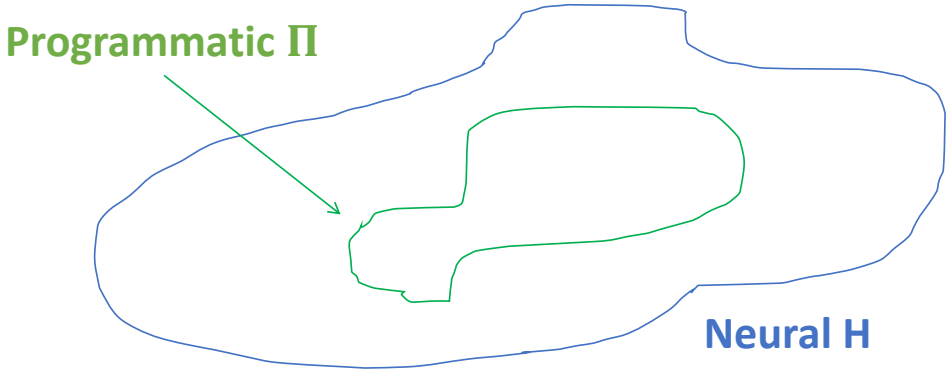
Abhinav Verma

Hoang Le

- **Unconstrained space:** $H = \Pi \oplus F$

**Purely Neural Representation**

- **Unconstrained policy:** $h(s) = \pi(s) + f(s)$



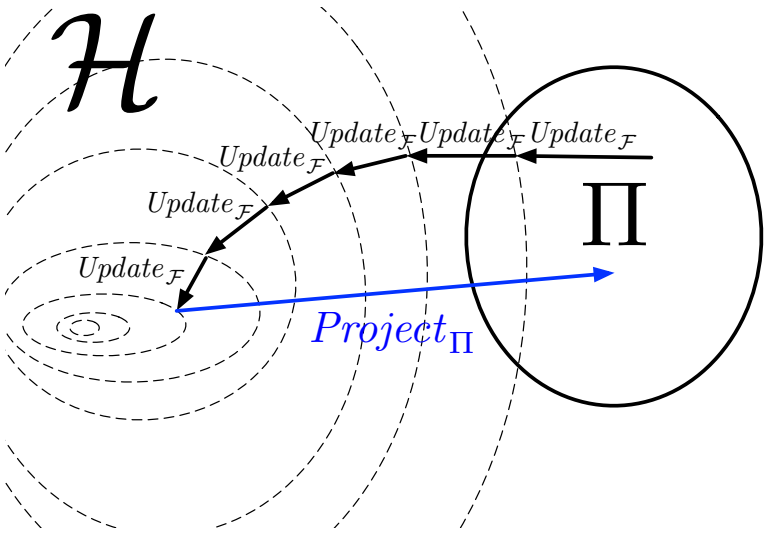

**Imitation-Projected Programmatic Reinforcement Learning,** Abhinav Verma, Hoang Le, et al., NeurIPS 2019

# PROPEL: Imitation-Projected Programmatic Reinforcement Learning

Abhinav Verma

Hoang Le

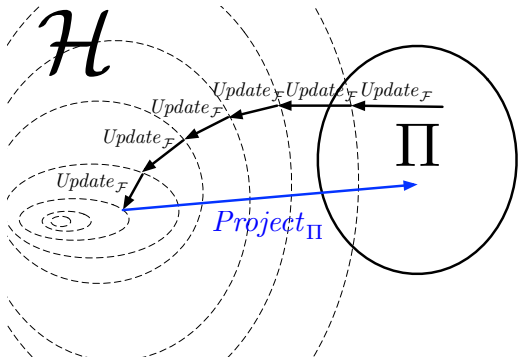
- **Unconstrained policy:** $h(s) = \pi(s) + f(s)$
- **Update:** hold $\pi$ fixed, update $f$
  - Can use any deep RL approach
- **Project:** train new $\pi$ to imitate $h$
  - Can use any program learning approach
- Theoretical analysis in paper




**Imitation-Projected Programmatic Reinforcement Learning,** Abhinav Verma, Hoang Le, et al., NeurIPS 2019

# Aside: Model/Policy Distillation

- Examples:
  - Distill bigger network into smaller network
  - Distill neural net into program
- Most work focus on 1-shot distillation
  - Train more complex model/policy, then distill
- PROPEL suggests iterative distillation
  - Viewed through the lens of constrained mirror descent




Ready
3
4
5
6
7
120
180
240


Ready
3
4
5
6
7
120
180
240

# Zero-Shot Generalization

Test Track

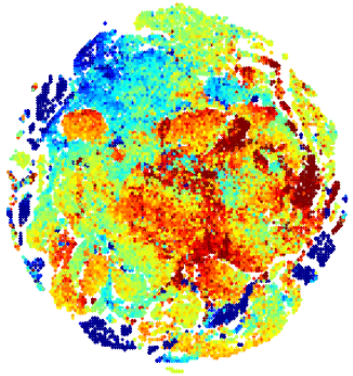
Training Track

| | G-Track | E-Road | Aalborg | Ruudskogen | Alpine-2 |
|---|---|---|---|---|---|
| G-Track | - | 124 / CR | CR / CR | CR / CR | CR / CR |
| E-Road | 102 / 92 | - | CR / CR | CR / CR | CR / CR |
| Aalborg | 201 / 91 | 228 / CR | - | 217 / CR | CR / CR |
| Ruudskogen | 131 / CR | 135 / CR | CR / CR | - | CR / CR |
| Alpine-2 | 222 / CR | 231 / CR | 184 / CR | CR / CR | - |

Reporting: PROPEL / DDPG in time to completion

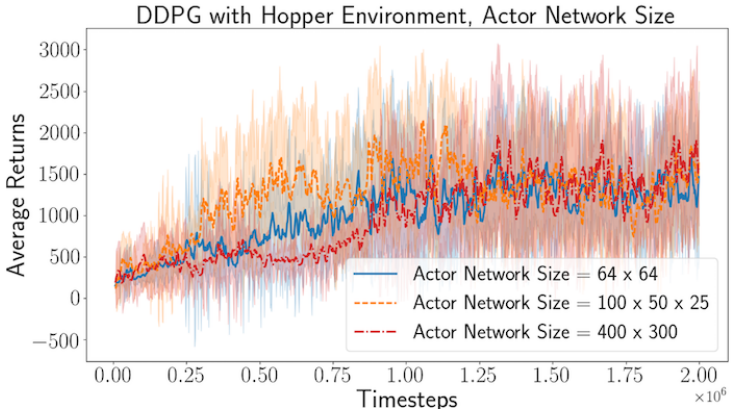
**Imitation-Projected Programmatic Reinforcement Learning,** Abhinav Verma, Hoang Le, et al., NeurIPS 2019

# Recall: Challenges with “Black Box” Policy Learning

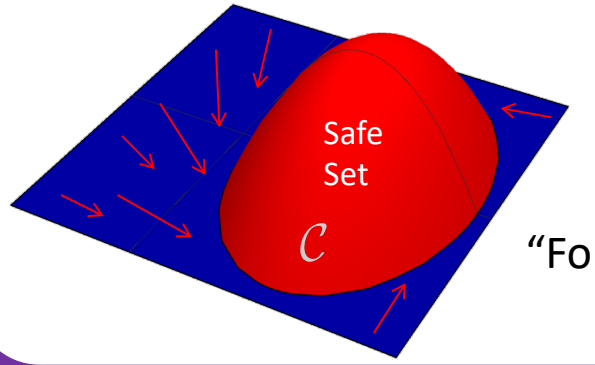
https://arxiv.org/abs/1611.00094

Interpretability

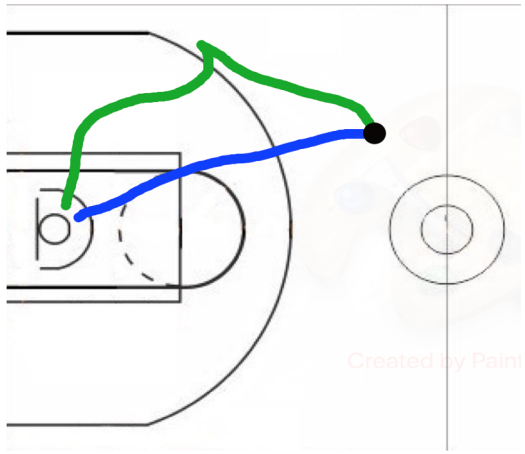



Sample Efficiency

https://arxiv.org/abs/1709.06560




“Formal” Interpretability



Controllability

https://arxiv.org/abs/1910.01179

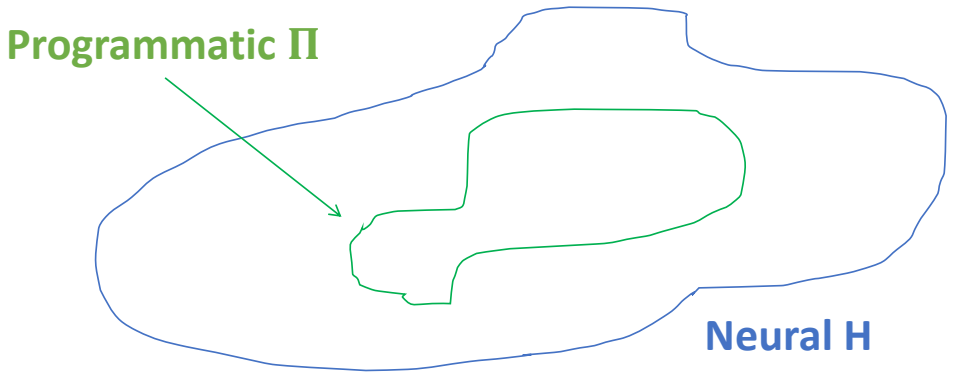
# Certifiable Policy Learning

- Unconstrained space: $H = \Pi \oplus F$

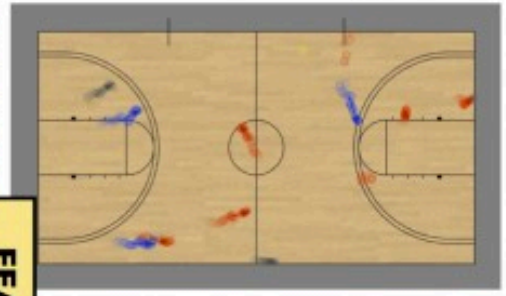
  **Purely Neural Representation**

- Partially constrained policy: $h(s) = \pi(s) + \lambda f(s)$ **$\lambda$ small**

- Suppose all $\pi \in \Pi$ are "certified"
  - **Then $h$ is approximately certified!**



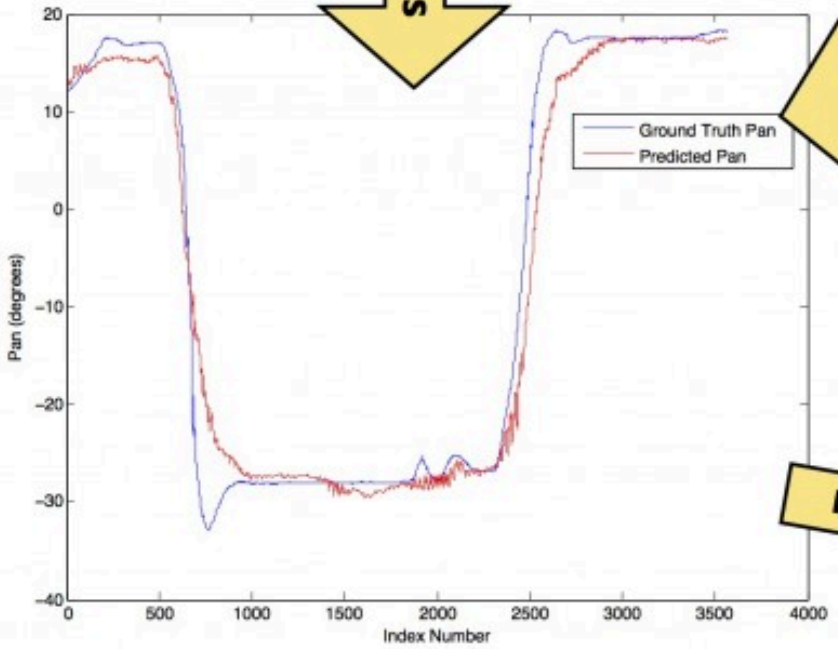
## Realtime Player Detection and Tracking



FEATURES

## Human Operated Camera

TRAIN




## Learned Regressor

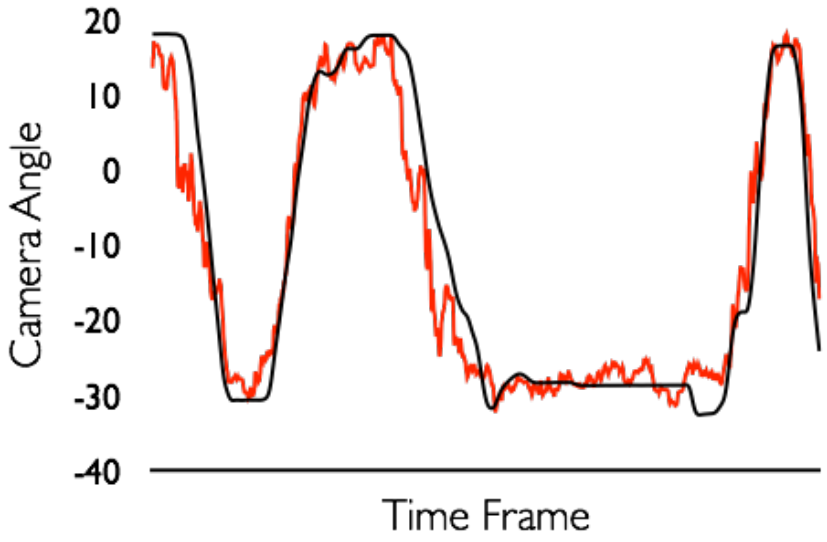
PREDICT

## Autonomous Robotic Camera

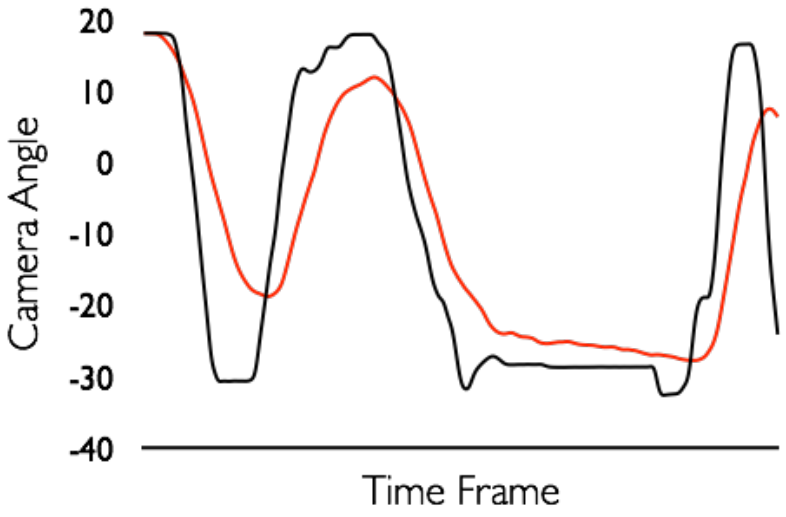
Disney Research

# Naïve Approach

- Supervised learning of demonstration data
  - Train predictor per frame
  - Predict per frame

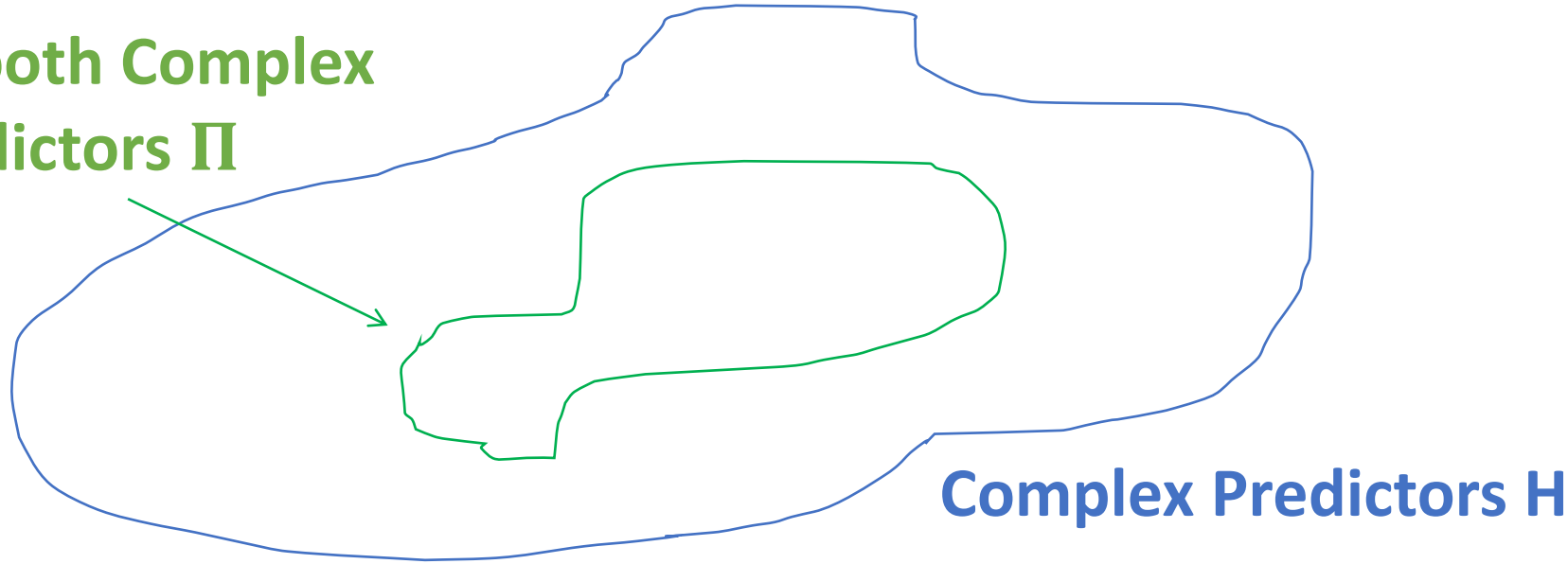



**In practice, 2-step smoothing:**



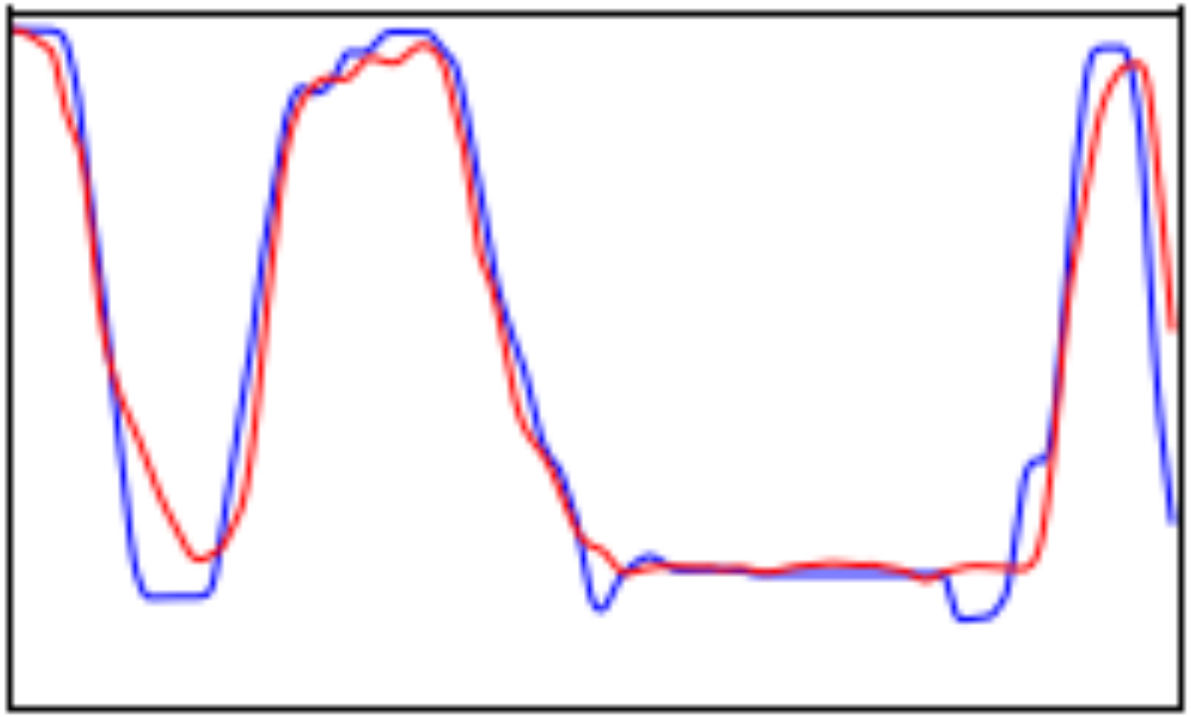
# Provably Smooth Policies

Hoang Le



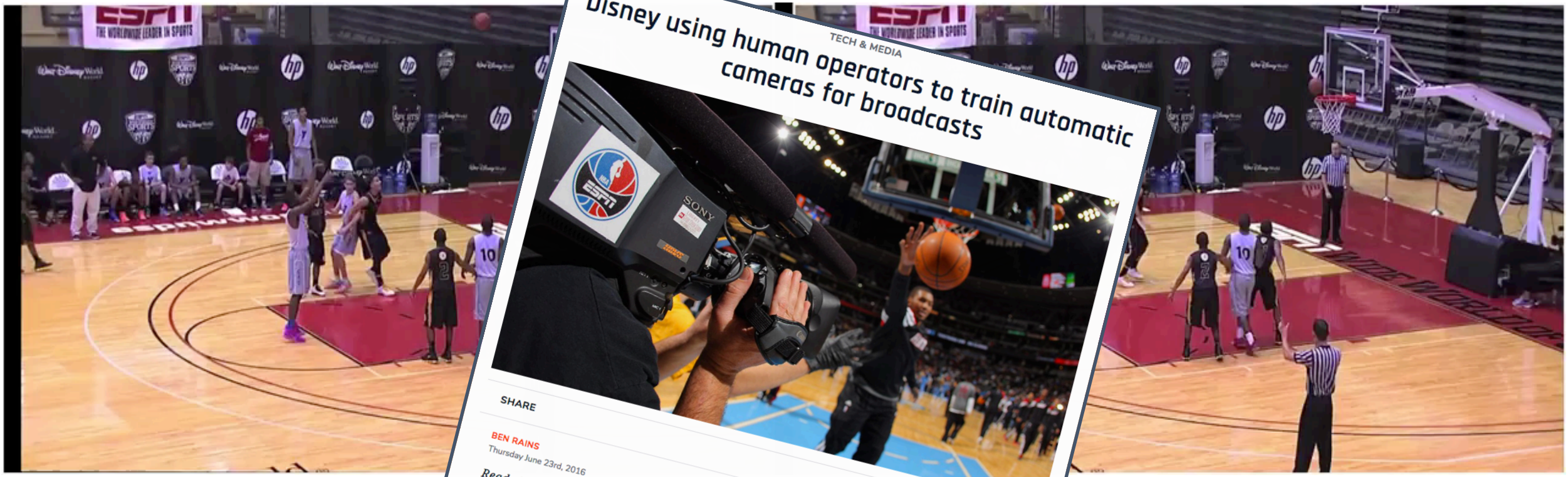

**Smooth Imitation Learning for Online Sequence Prediction**
Hoang Le, Andrew Kang, Yisong Yue, Peter Carr. ICML 2016

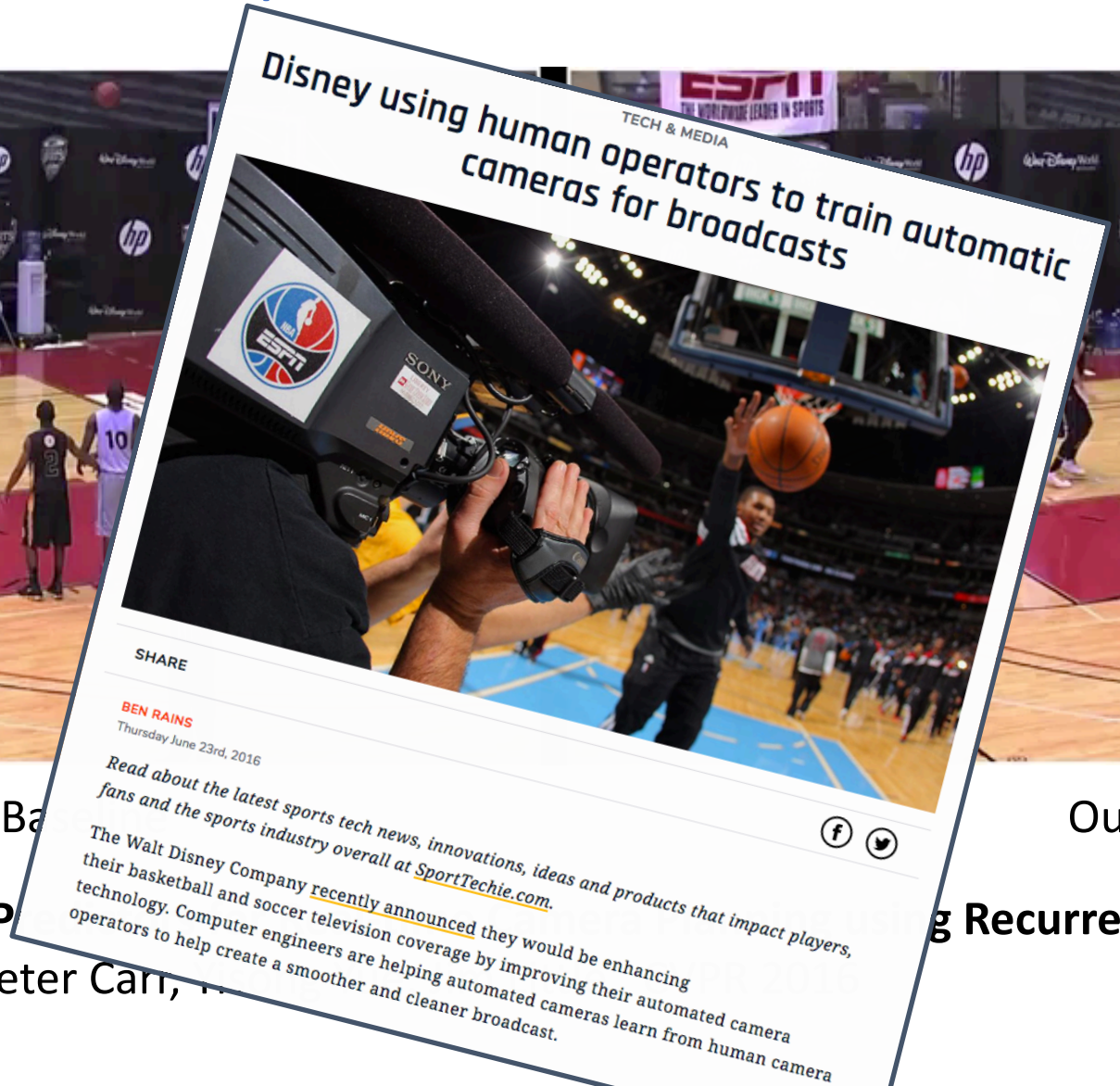
# Our Results



**Smooth Imitation Learning for Online Sequence Prediction**
Hoang Le, Andrew Kang, Yisong Yue, Peter Carr. ICML 2016

# Qualitative Comparison





TECH & MEDIA

## Disney using human operators to train automatic cameras for broadcasts

SHARE

BEN RAINS
Thursday June 23rd, 2016

*Read about the latest sports tech news, innovations, ideas and products that impact players, fans and the sports industry overall at SportTechie.com.*

The Walt Disney Company recently announced they would be enhancing their basketball and soccer television coverage by improving their automated camera technology. Computer engineers are helping automated cameras learn from human camera operators to help create a smoother and cleaner broadcast.

2-Step Ba

Our Approach

**Learning Online Smooth P**

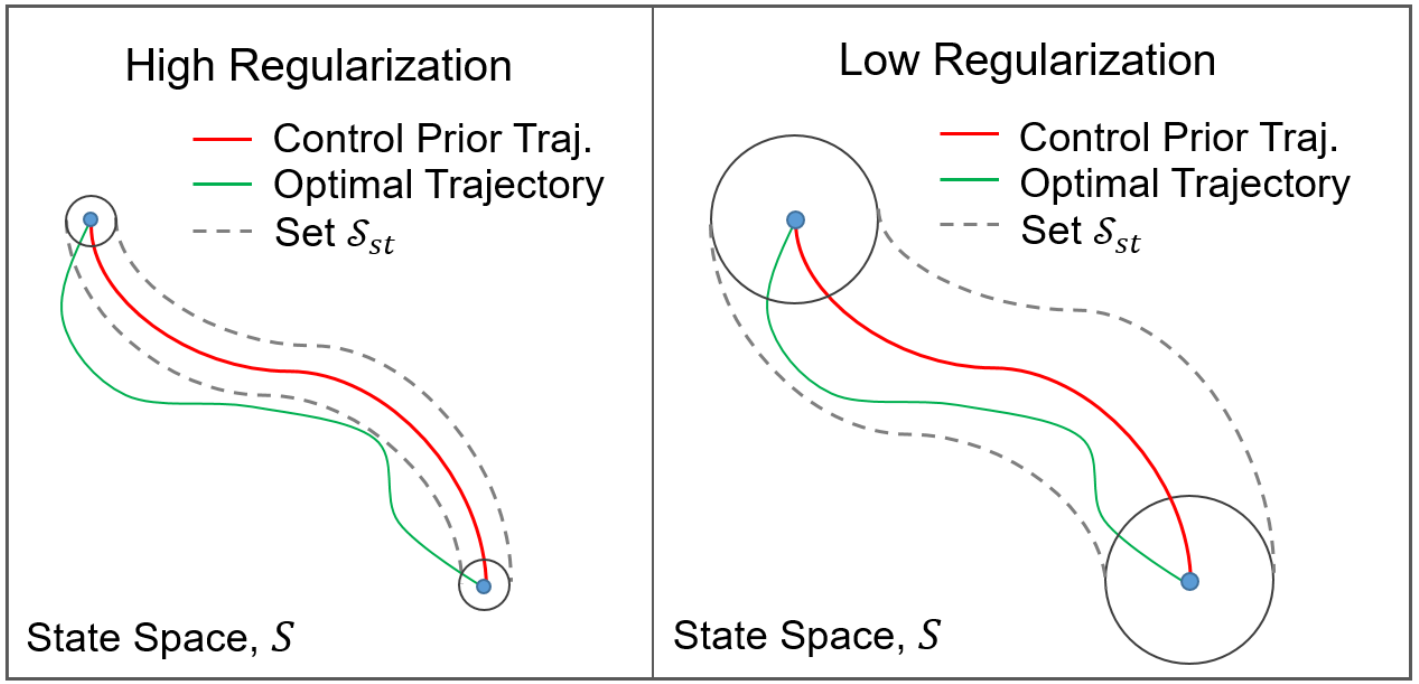
**g Recurrent Decision Trees**

Jianhui Chen, Hoang Le, Peter Carr,

# Control-Theoretic Guarantees

Richard Cheng

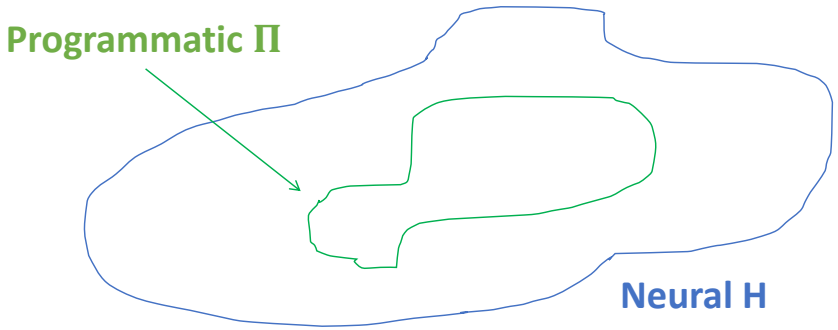
- (Relaxed) Lyapunov stability bounds:

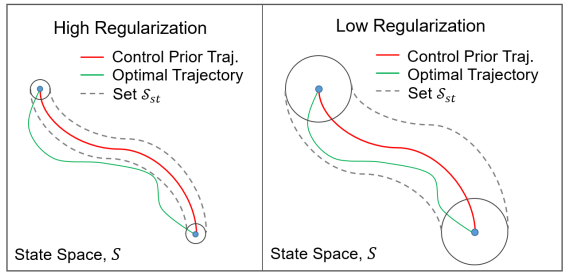


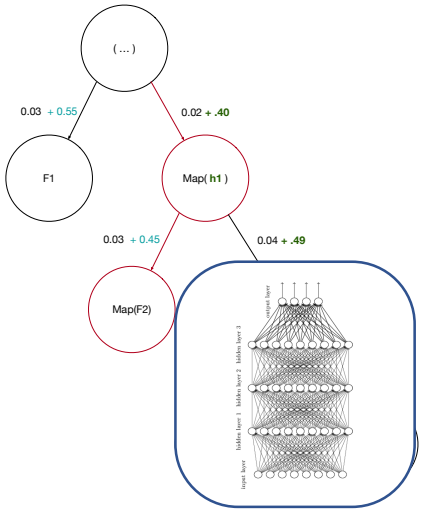

**Control Regularization for Reduced Variance Reinforcement Learning**
Richard Cheng, Abhinav Verma, Gabor Orosz, Swarat Chaudhuri, Yisong Yue, Joel Burdick. ICML 2019

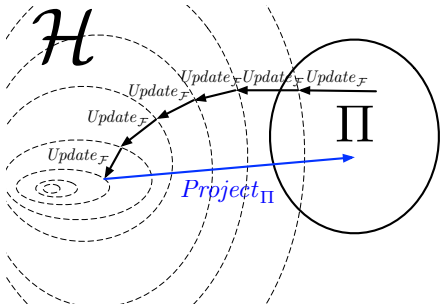
# Summary: Programmatic Inductive Bias

- Represent Policies as Programs
- Generalization / Sample-Efficiency
- Certifiable Guarantees
- Neural Relaxation Perspective
- **New Learning Approaches!**



# 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

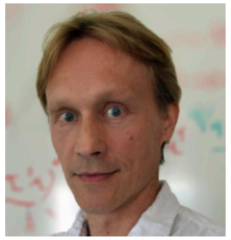
Isil Dillig
Associate Professor, UT Austin

Osbert Bastani
Research Assistant Professor, University of Pennsylvania

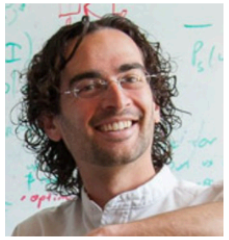
Michael Carbin
Assistant Professor, MIT

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

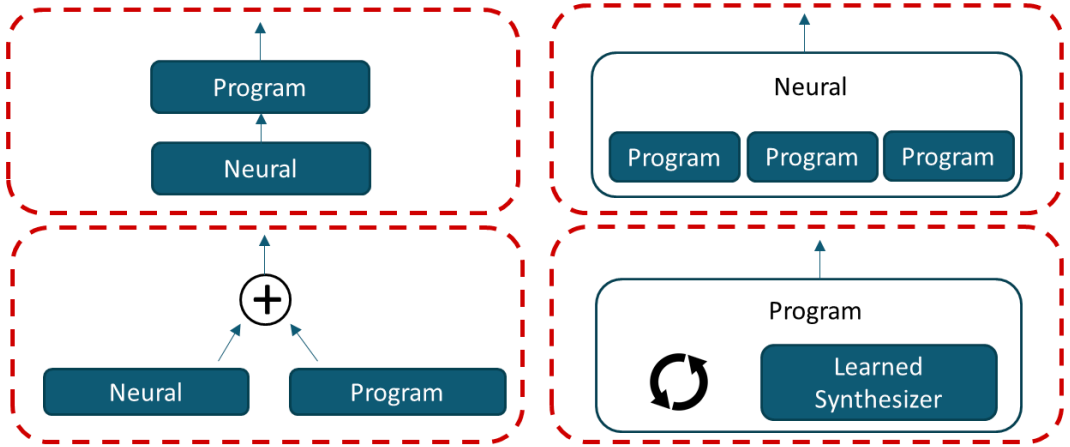
Justin Gottschlich
Head of Machine Programming Research, Intel

Satish Chandra
Engineering Manager, Facebook

Susmit Jha
Principal Computer Scientist, SRI International

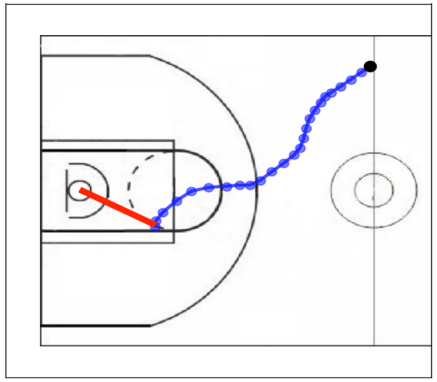
# Programmatic Domain Knowledge

- Programmatic Priors

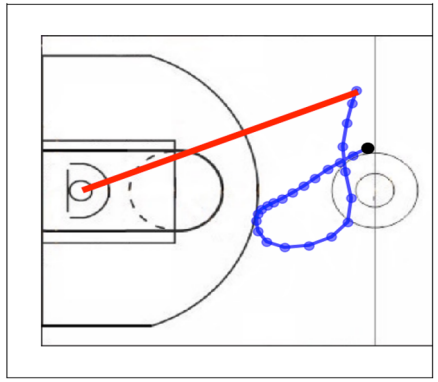



"Neurosymbolic" learning

- Programmatic Supervision



Small Distance

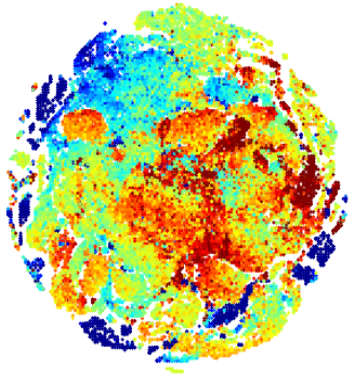


Large Distance

"Data Programming" for sequential behavioral data

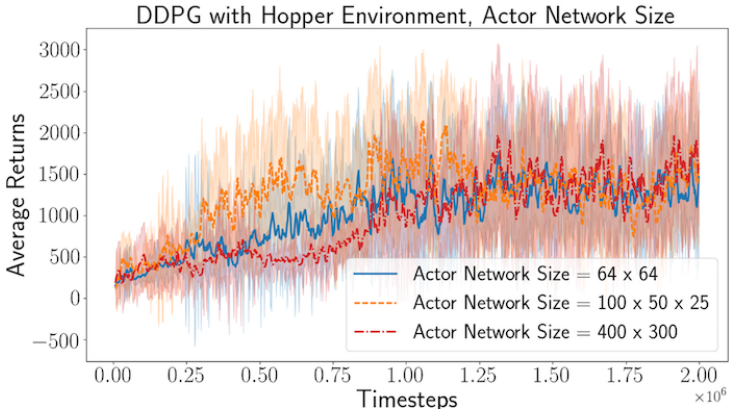
https://arxiv.org/abs/1605.07723

# Recall: Challenges with “Black Box” Policy Learning

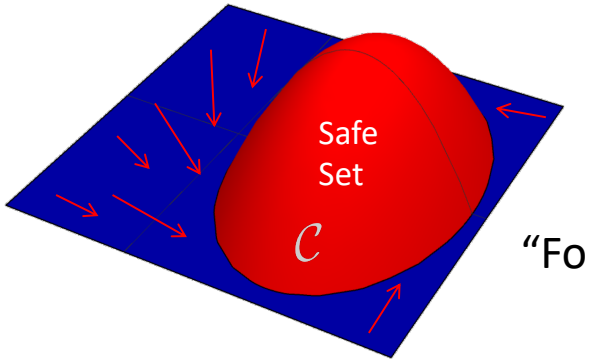
https://arxiv.org/abs/1611.00094

Interpretability

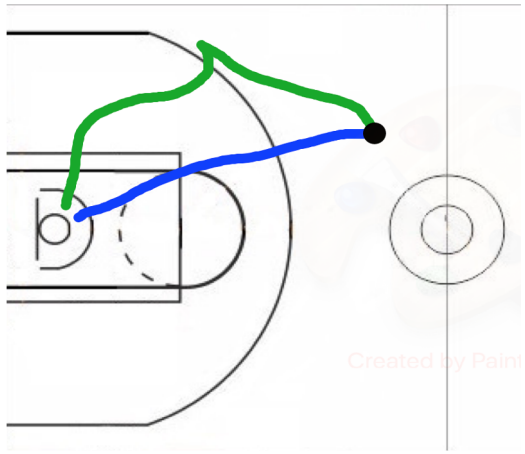



Sample Efficiency

https://arxiv.org/abs/1709.06560



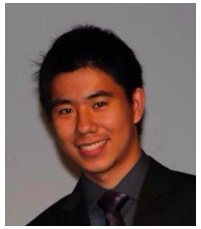

“Formal” Interpretability



Controllability

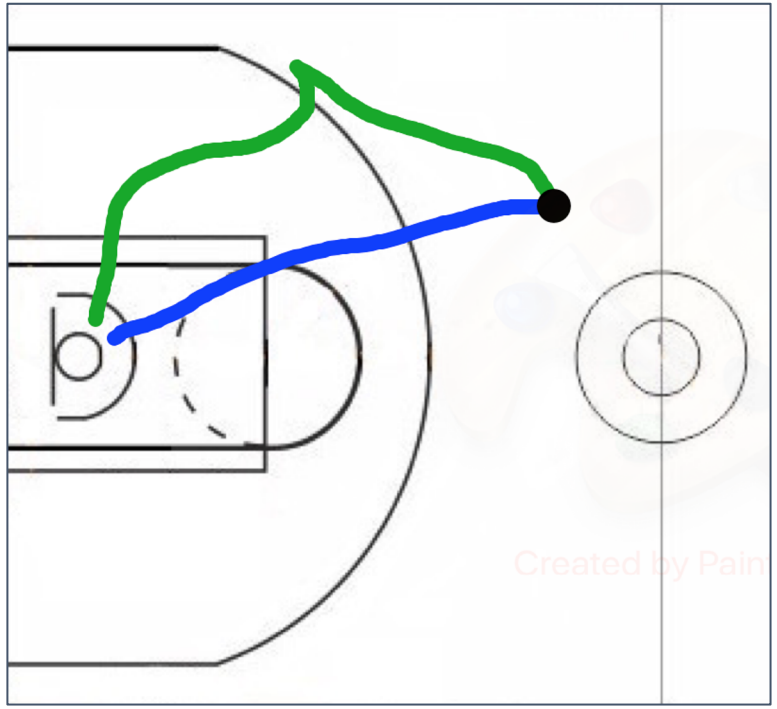
https://arxiv.org/abs/1910.01179

# Style-Calibrated Generation



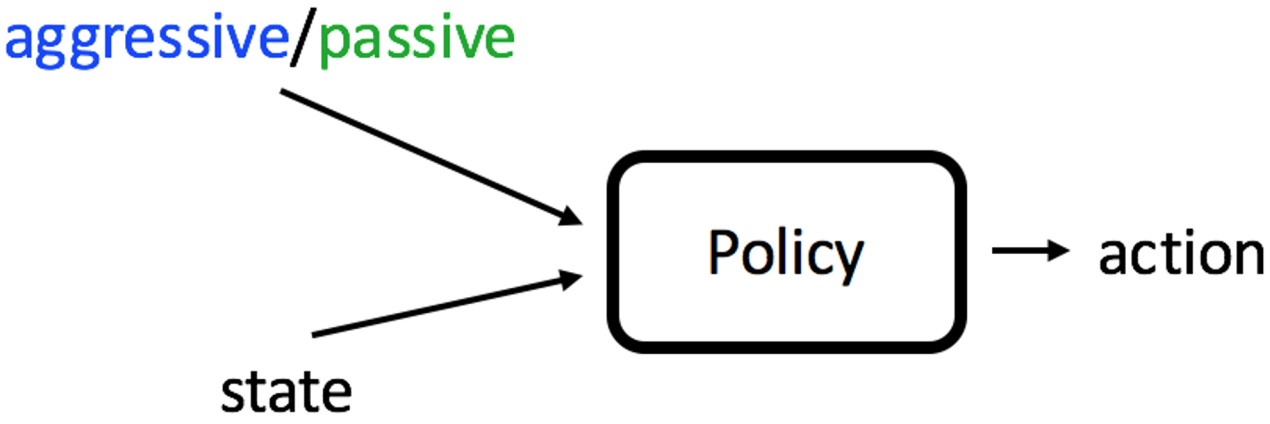
Eric Zhan

**Goal**: controllable generation of behaviors



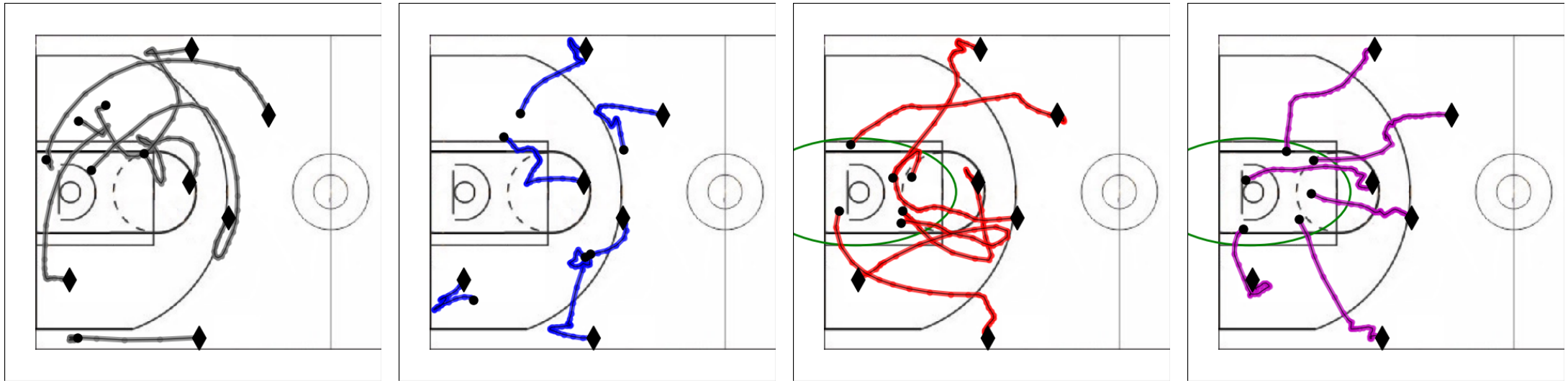
Both trajectories move towards the basket

- Blue style is more aggressive
- Green style is more passive




**Learning Calibratable Policies using Programmatic Style-Consistency,** Eric Zhan et al., ICML 2020

# Many Possible Styles



(a) Expert demonstrations (b) Style: SPEED (c) Style: DESTINATION (d) Both styles

**Learning Calibratable Policies using Programmatic Style-Consistency**
Eric Zhan, Albert Tseng, Yisong Yue, Adith Swaminathan, Matthew Hausknecht, ICML 2020

# Key Challenges

**Challenge #1**: expert labels for behavior styles are hard to obtain

**Challenge #2:** joint style space scales combinatorially wrt. # styles

**Our solution:**

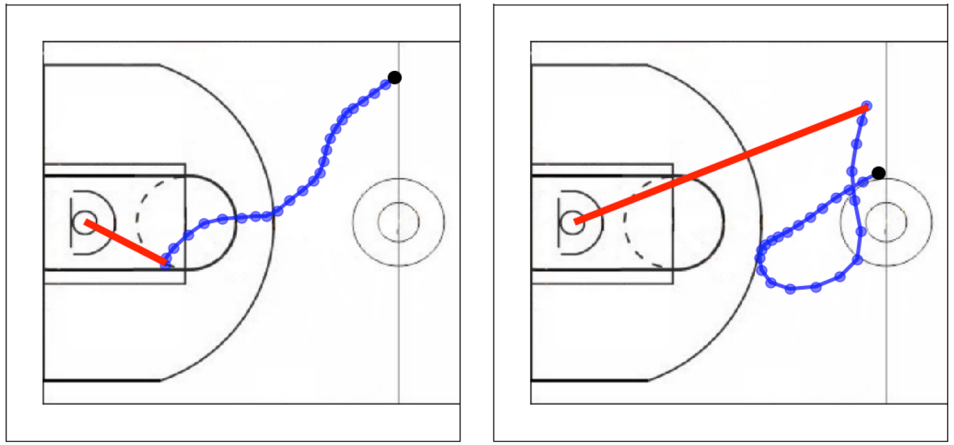
- Expert-written style-labeling functions (programmatic supervision)
- Style-consistency learning objective

# Labeling Functions for Behavior Styles

Behavior styles via programmatic labeling functions: $\lambda(\tau) = \mathbf{y}$

- e.g. small/large distance to basket

$$\lambda(\tau) = \mathbb{1}\{\|\mathbf{s}_{T+1} - \mathbf{s}_{\text{basket}}\|_2 > b\}$$

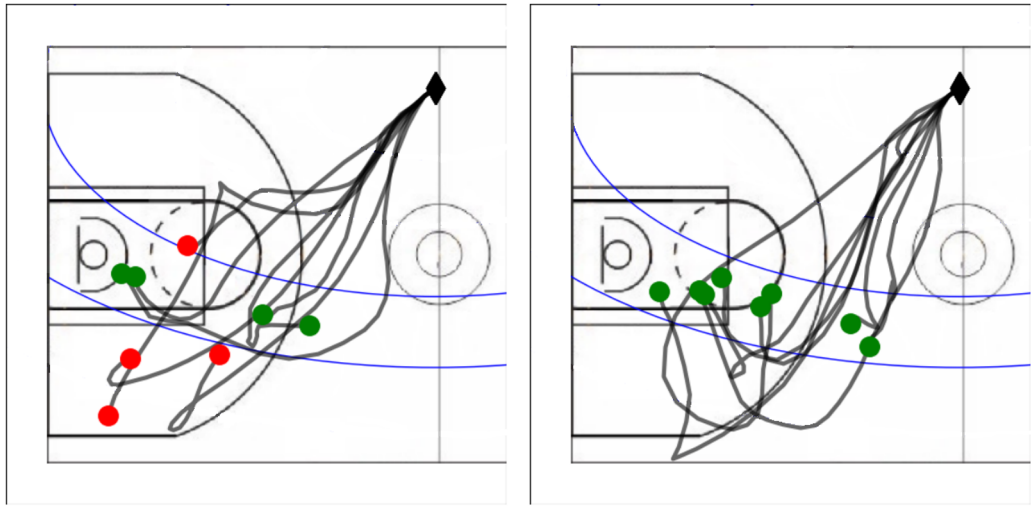


Advantages:

- Fast to compute
- Compactly encodes domain knowledge
- Readily available

"Data Programming" for sequential behavioral data
https://arxiv.org/abs/1605.07723

# Style-Consistency



not consistent consistent

**Want to directly optimize for style-consistency!**

# Style-Consistency Objective

## (can be extended to multiple styles)

Eric Zhan

Trajectory generated by $\pi(\cdot \mid y)$ should be consistent with $y$

$$\pi^* = \arg\min_{\pi} \underbrace{\mathbb{E}_{(\tau,\mathbf{y})\sim\mathcal{D}_\lambda}\left[\mathcal{L}^{\text{imitation}}\big(\tau, \pi(\cdot \mid \mathbf{y})\big)\right]}_{\textbf{Standard Imitation Learning}} + \underbrace{\mathbb{E}_{\mathbf{y}\sim p(\mathbf{y}),\tau\sim\pi(\cdot|\mathbf{y})}\left[\mathcal{L}^{\text{style}}\big(\lambda(\tau), \mathbf{y}\big)\right]}_{\textbf{Style Consistency}}$$

See paper for learning algorithm

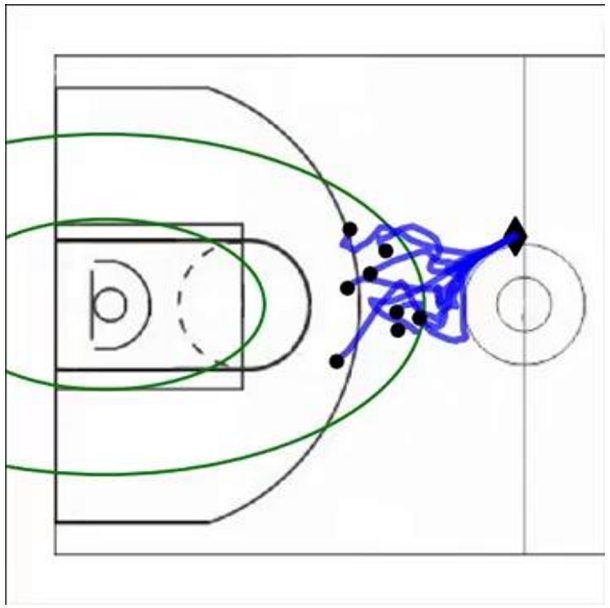
**Example:** $\mathcal{L}^{\text{style}}\big(\lambda(\tau), \mathbf{y}\big) = \mathbb{1}\{\lambda(\tau) \neq \mathbf{y}\}$

**Learning Calibratable Policies using Programmatic Style-Consistency**
Eric Zhan, Albert Tseng, Yisong Yue, Adith Swaminathan, Matthew Hausknecht, ICML 2020

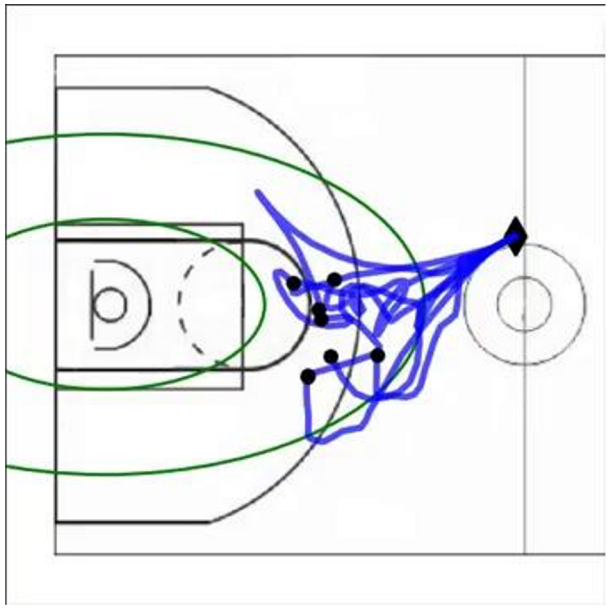
# Experimental Results for Basketball

Style #1: [medium] distance to basket

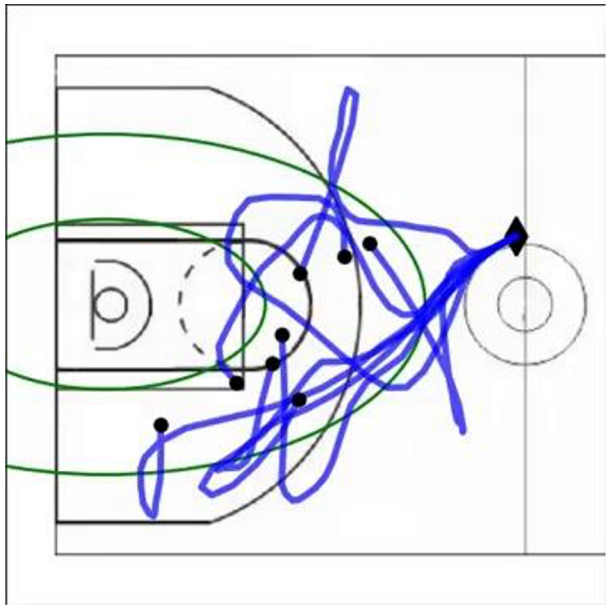
Style #2: [slow, medium, fast] speed



slow speed



medium speed



fast speed

# Experimental Results for Basketball

| Model | 2 styles | 3 styles | 4 styles | 5 styles |
|---|---|---|---|---|
| CTVAE | 0.71 | 0.58 | 0.50 | 0.37 |
| CTVAE-info | 0.69 | 0.58 | 0.51 | 0.32 |
| CTVAE-mi | 0.72 | 0.56 | 0.51 | 0.30 |
| CTVAE-style | **0.93** | **0.88** | **0.88** | **0.75** |

Joint style space of
3^5 = 243 joint styles!

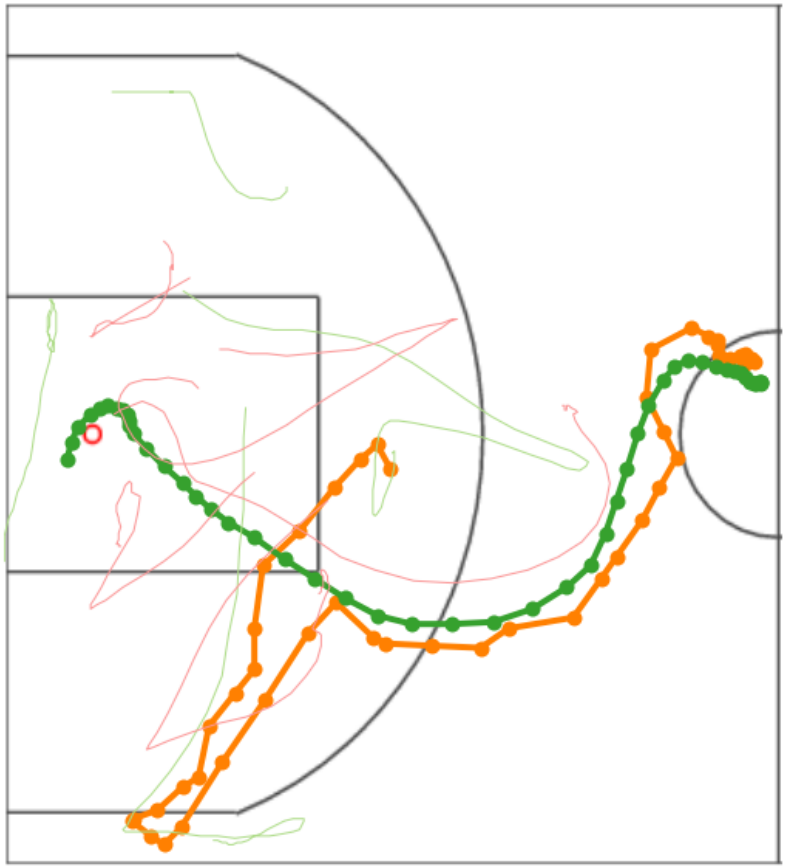
Median style-consistency over 5 seeds, 4000 trajectories each (can interpret as accuracy)

- Using CTVAE as base imitation learner
- Baselines use “indirect” learning objectives
- Evaluate on matching all styles simultaneously

**Learning Calibratable Policies using Programmatic Style-Consistency**
Eric Zhan, Albert Tseng, Yisong Yue, Adith Swaminathan, Matthew Hausknecht, ICML 2020

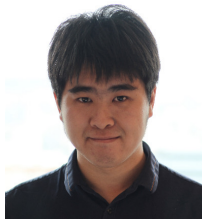
# Another Motivation: Strategy vs Tactics

- Long-term Goal:
  - Curl around basket
- Tactics
  - Drive left w/ ball
  - Pass ball
  - Cut towards basket



Stephan Zheng

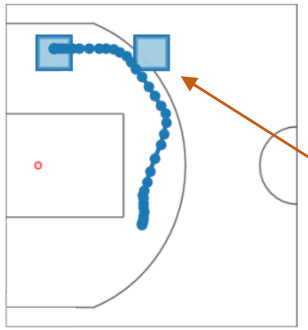
Eric Zhan



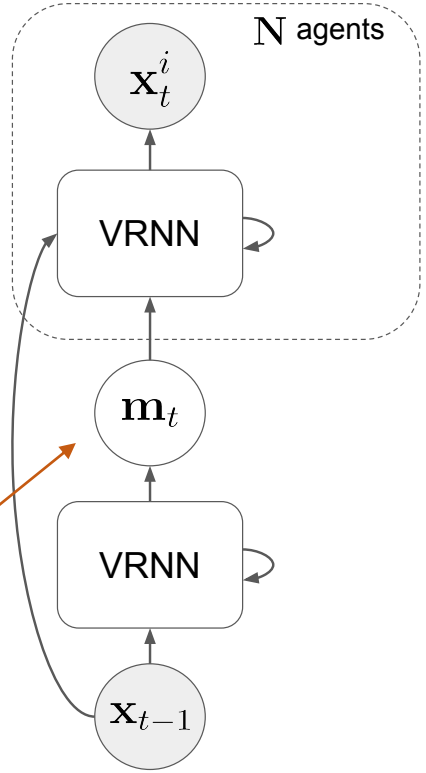
Yukai Liu

# Generative + Hierarchical

- **Generative Imitation Learning**
  - No single "correct" action
- **Hierarchical**
  - Make predictions at multiple resolutions



Macro-goals



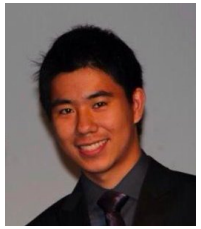

Programmatically Generated

**Generating Long-term Trajectories using Deep Hierarchical Networks**
Stephan Zheng, Yisong Yue, Patrick Lucey. NeurIPS 2016

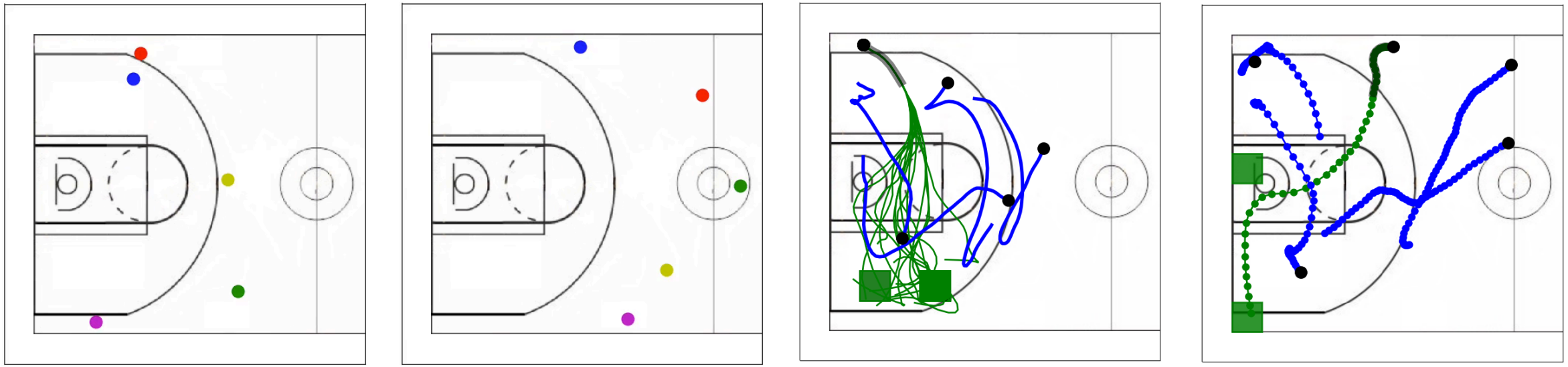
**Generating Multi-agent Trajectories using Programmatic Weak Supervision**
Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, Patrick Lucey. ICLR 2019

**NAOMI: Non-Autoregressive Multiresolution Sequence Imputation**
Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, Yisong Yue. NeurIPS 2019

# Simulation Results



Eric
Zhan

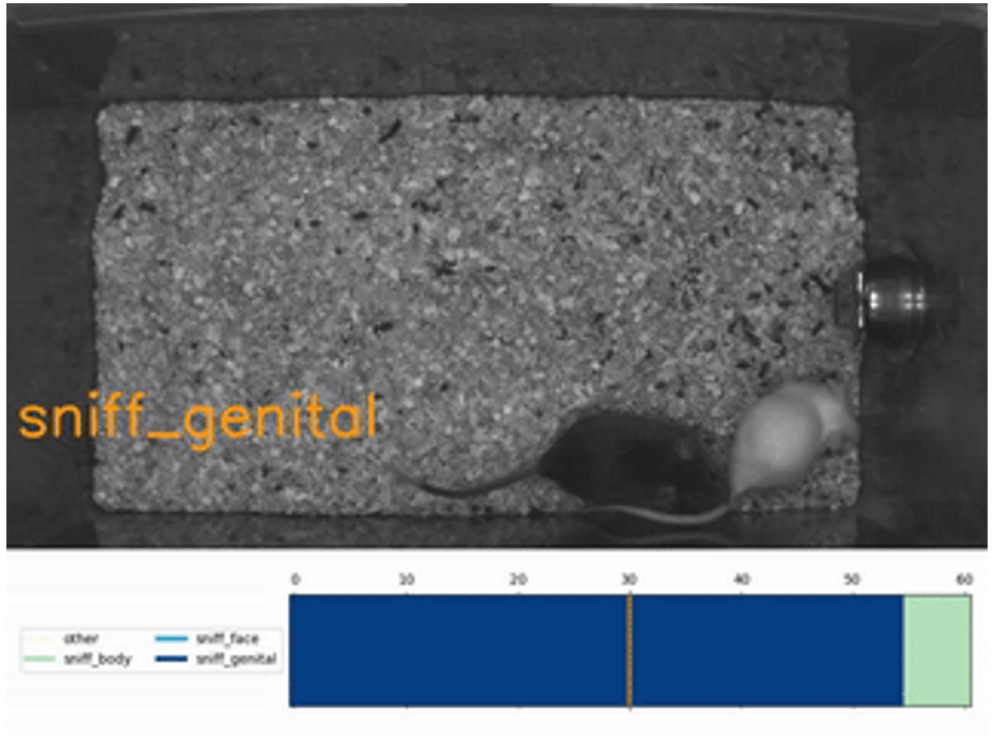


**Generating Multi-agent Trajectories using Programmatic Weak Supervision**
Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, Patrick Lucey. ICLR 2019

https://www.youtube.com/watch?v=0q1j22yMipY

# Another Project: Programs as Auxiliary Supervision

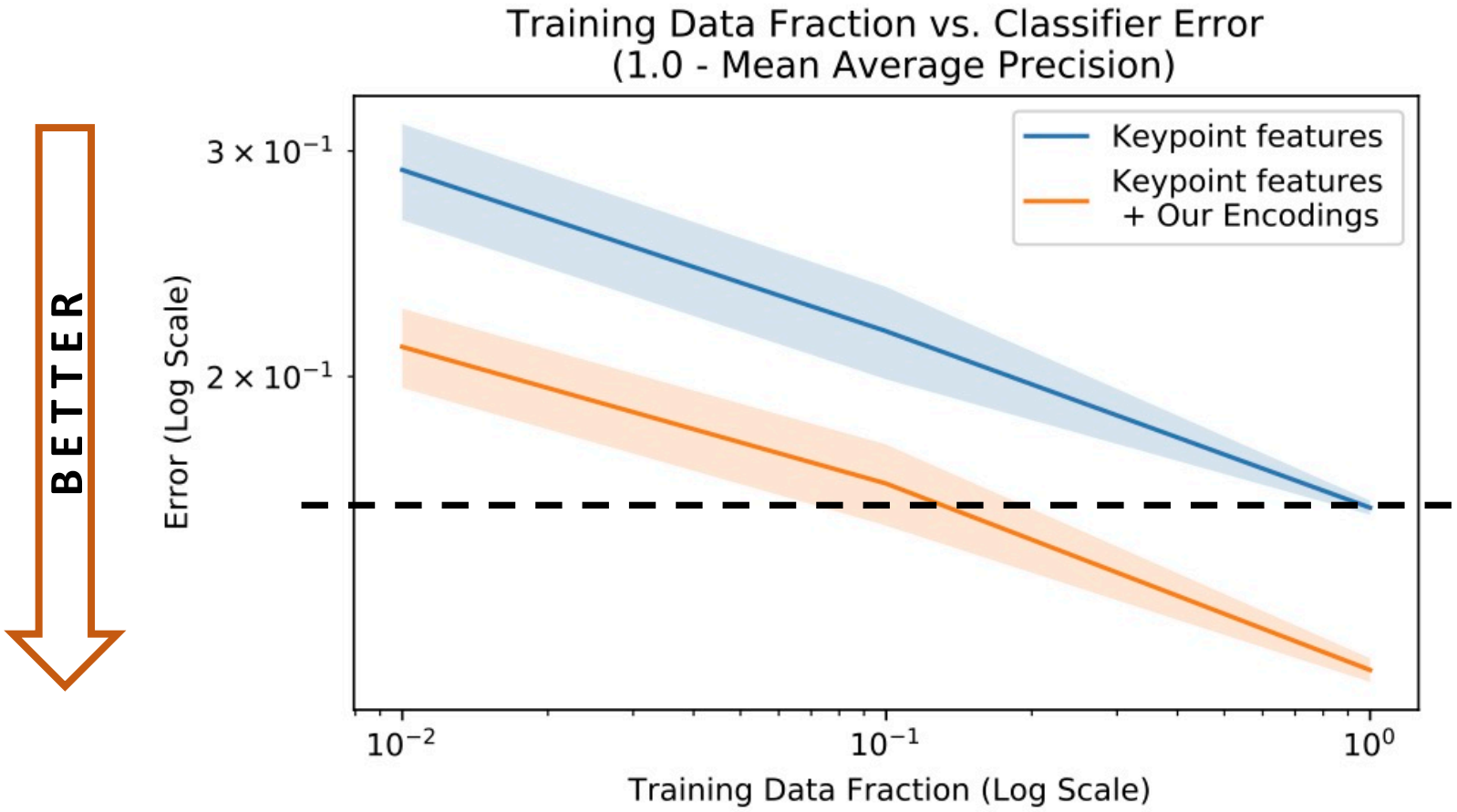
- Biggest bottleneck in behavior modeling is annotation




https://github.com/annkennedy/bento

# Task Programming

Jennifer Sun

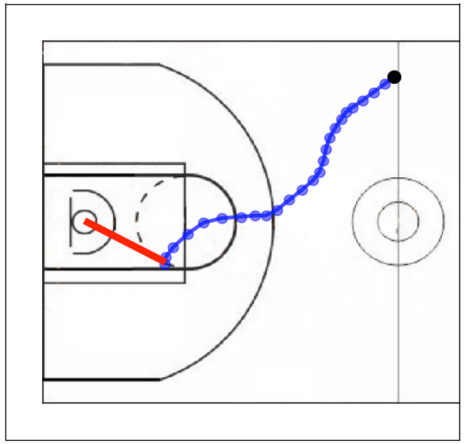


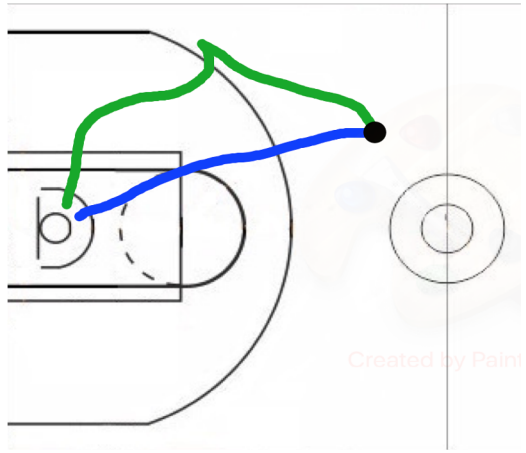

**Can lead to 10x annotation efficiency!**

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

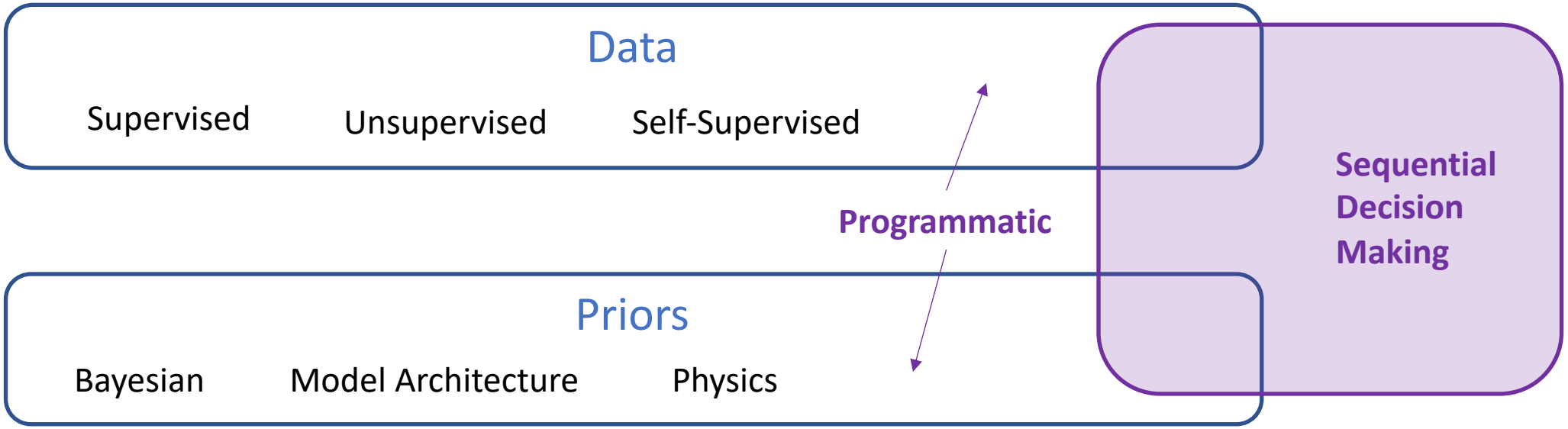
# Summary: Programmatic Supervision

- Expert-written labeling functions
  - (Ongoing work: automatically discovered)
- Programmatic supervision
- New learning objectives
- Controllable generation
- Annotation-efficient downstream tasks

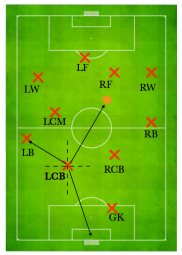


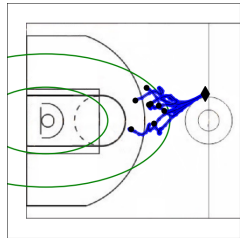


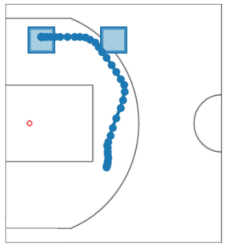
# Programmatic Domain Knowledge




**Many cool applications!**





# Collaborators


Eric Zhan


Stephan Zheng


Hoang Le


Ameesh Shah


Richard Cheng


Abhinav Verma

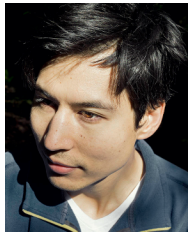

Jennifer Sun

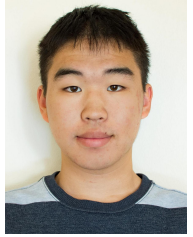

Jimmy Chen


Andrew Kang


Long Sha

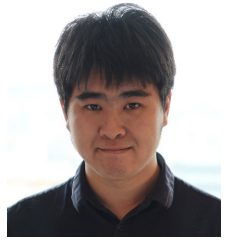

Adith Swaminathan

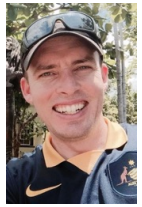

Matthew Hausknecht


Albert Tseng


Ann Kennedy


Rose Yu

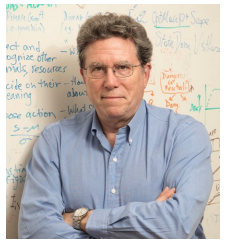

Yukai Liu


Patrick Lucey


Joel Burdick


Gabor Orosz


Swarat Chaudhuri


David Anderson


Pietro Perona


Jim Little


Peter Carr

# References

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

**Imitation-Projected Programmatic Reinforcement Learning,** Abhinav Verma, Hoang Le, et al., NeurIPS 2019
- https://bitbucket.org/averma8053/propel/src/master/

**Smooth Imitation Learning for Online Sequence Prediction,** Hoang Le, et al., ICML 2016
- https://sites.google.com/view/smooth-imitation-learning

**Learning Smooth Online Predictors for Real-Time Camera Planning using Recurrent Decision Trees,** Jianhui Chen, et al., CVPR 2016

**Control Regularization for Reduced Variance Reinforcement Learning**, Richard Cheng et al., ICML 2019
- https://github.com/rcheng805/CORE-RL

**Learning Calibratable Policies using Programmatic Style-Consistency,** Eric Zhan et al., ICML 2020
- https://github.com/ezhan94/calibratable-style-consistency
- http://basketball-ai.com/

**Generating Long-term Trajectories using Deep Hierarchical Networks,** Stephan Zheng et al., NeurIPS 2016

**Generating Multi-agent Trajectories using Programmatic Weak Supervision,** Eric Zhan et al., ICLR 2019
- https://github.com/ezhan94/multiagent-programmatic-supervision

**NAOMI: Non-autoregressive Multiresolution Sequence Imputation**, Yukai Liu et al., NeurIPS 2019
- https://github.com/felixykliu/NAOMI

**Task Programming: Learning Data Efficient Behavior Representations,** Jennifer J. Sun, et al., CVPR 2021
- https://sites.google.com/view/task-programming