Deep TAMER: Interactive Agent Shaping in High-Dimensional State Spaces

Presenter: Jordi Ramos

[Date]

Preview

Tamer

- **Non expert human**
  - **provides scalar feedback**
  - **agent estimates a hidden function**
- **On-policy**
- **model-based**

Deep Tamer

- **High dimensional state-space**

Motivation and Main Problem

1-5 slides

High-level description of problem being solved

Why is the problem important?

- its significance towards general-purpose robot autonomy
- its potential application and societal impact of the problem

Technical challenges arising from the problem

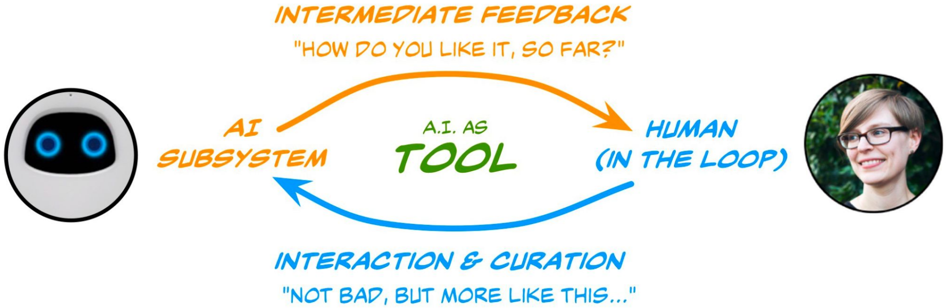
- the role of the AI and machine learning in tackling this problem

High-level idea of why prior approaches didn’t already solve

Key insight(s) (try to do in 1-3) of the proposed work

Motivation and Main Problem

- Increase the agent learning speed
    - leveraging the input of human trainers
- Humans can provide real-time, scalar-valued feedback
- Reduce training data required by RL methods
- Agents are still doing the learning



- Picture from Stanford HAI

Problem Setting

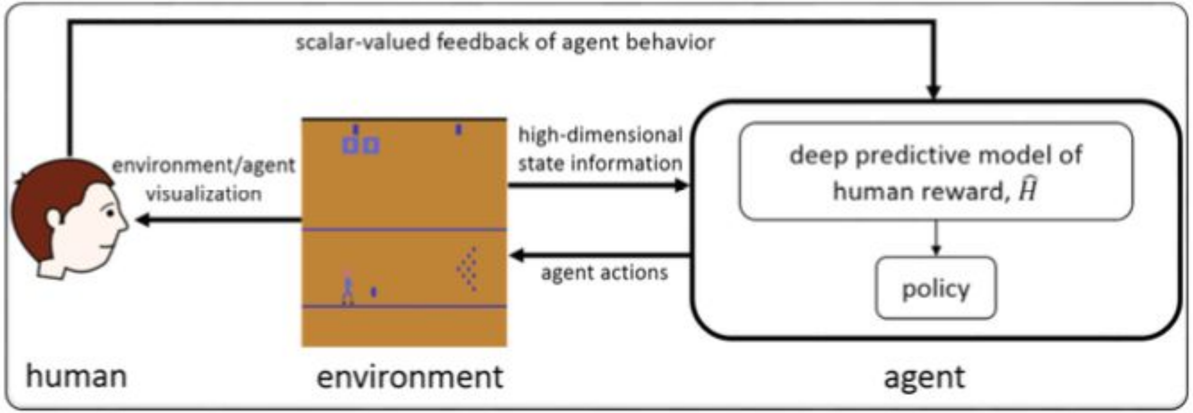
1 or more slides

Problem formulation, key definitions and notations

❖ Be precise -- should be as formal as in the paper

Problem Setting

- Sequatien decision making (Policy)
  - Hand coded
  - Using RL method
- Fast training is of crucial importance
  - Rapid agent learning
- Observer provides scalar value feedback
- Atari Bowling


scalar-valued feedback of agent behavior
environment/agent visualization
high-dimensional state information
agent actions
deep predictive model of human reward, $\hat{H}$
policy
human
environment
agent

Context / Related Work / Limitations of Prior Work

1 or more slides

Which other papers have tried to tackle this problem or a related problem?

- The paper’s related work is a good start, but there may be others
- What is the key limitations of prior work(s)?
- Three works by Stone
  - Only shown to work in low-dimensional state spaces
- Chritiano et al. 2017

Related Work

- Learning from demonstrations - Schaal 1997; Argall et al. 2009; Hussein et al. 2017
  - Human may not always be available
  - hard to generalize
- Inverse reinforcement learning - Ng and Russell 2000; Abbeel and Ng 2004
  - autonomous agents learn reward function
  - performance capped by that of the demonstrator
- Reward shaping - Skinner 1938; Randløv and Alstrøm 1998; Ng, Harada, and Russell 1999; Devlin and Kudenko 2012
  - Human modify low-level reward function
  - Human might not know the how to modify the reward function

Related Work (Ctd.)

- TAMER (Training an Agent Manually via Evaluative Reinforcement)
  - Knox and Stone 2009; 2012; Knox, Stone, and Breazeal 2013; Knox and Stone 2015
  - Only shown to work in low-dimensional state spaces
- Deep reinforcement learning from human preferences - Christiano et al. 2017
  - deep learning queries humans to compare behavior examples
  - Differences:
    - the use of a simulator
    - the use of powerful hardware

Proposed Approach / Algorithm / Method

1-5 slides

Describe algorithm or framework (pseudocode and flowcharts can help)

- ❖ What is the optimization objective?
- ❖ What are the core technical innovations of the algorithm/framework?

Implementation details should be left out here, but may be discussed later if its relevant for limitations / experiments

Problem Formulation

- Human’s a reward function: $H(\cdot,\cdot) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$

- Estimate hidden function from human: $\pi(\mathbf{s}) = \max_{\mathbf{a}} \hat{H}(\mathbf{s},\mathbf{a})$.

scalar-valued feedback signal

- Defines a loss function for supervised learning:

$$\ell(\hat{H}\ ;\ \mathbf{x},\mathbf{y}) = w(t^s, t^e, t^f)\left[\hat{H}(\mathbf{s},\mathbf{a}) - \boxed{h}\right]^2, \tag{1}$$

  - Agent’s experience: $\mathbf{x} = (\mathbf{s},\mathbf{a}\boxed{, t^s, t^e)}$

interval of time at state s

  - Human’s feedback: $\mathbf{y} = (h, t^f)$

- Deep TAMER tries to minimize the loss:

$$\hat{H}^* = \arg\min_{\hat{H}} \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\ell(\hat{H}\ ;\ \mathbf{x},\mathbf{y})\right] \tag{2}$$

Method

- Deep TAMER tries to minimize the loss:

$$\hat{H}^* = \arg\min_{\hat{H}} \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\ell(\hat{H}\ ;\ \mathbf{x}, \mathbf{y}) \right] \tag{2}$$

- stochastic gradient descent (SGD):

$$\hat{H}_{k+1} = \hat{H}_k - \eta_k \nabla_{\hat{H}} \ell(\hat{H}_k\ ;\ \mathbf{x}_{i_k}, \mathbf{y}_{j_k})\ . \tag{3}$$

Importance Weights

- Defines a loss function for supervised learning:

$$\ell(\hat{H}\,;\,\mathbf{x},\mathbf{y}) = w(t^s,t^e,t^f)\left[\hat{H}(\mathbf{s},\mathbf{a}) - h\right]^2 , \qquad (1)$$

- weighted square loss

- feedback applies to recent agent behaviour

$$w(t^s,t^e,t^f) = \int_{t^f - t^e}^{t^f - t^s} f_{delay}(t)dt \, . \qquad (4)$$

- Continuous uniform distribution over the interval [0.2,4]

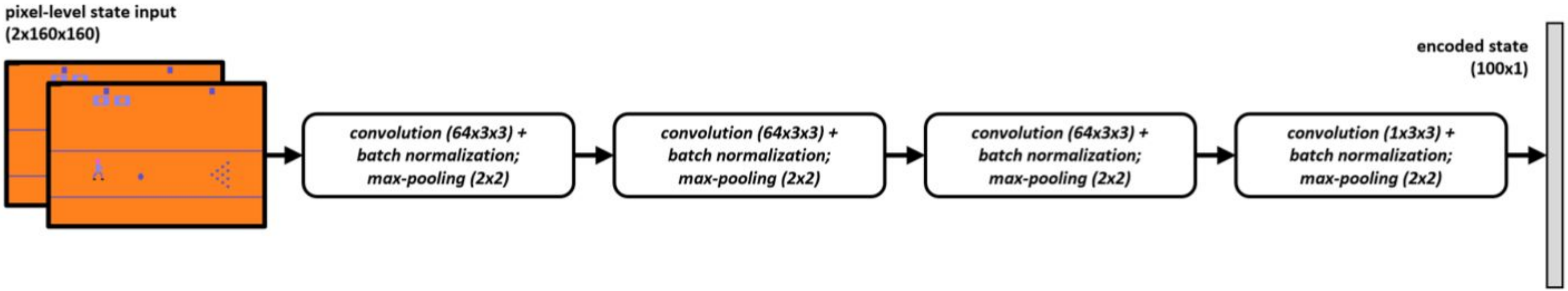
Deep Reward Model

- $\hat{H}$ as a CNN: $\hat{H}(\mathbf{s}, \mathbf{a}) = z(f(\mathbf{s}), \mathbf{a}).$
- Two strategies
  1. pretrain CNN portion using autoencoder
  2. use feedback replay buffer
     - Increase the rate of learning

Deep Autoencoder

- Input dimension = 51,200
- Output dimension = 100
- Pre-trained using:
    - by minimizing reconstruction error

$$(\boldsymbol{\theta}_f^*, \boldsymbol{\theta}_g^*) = \arg \min_{(\boldsymbol{\theta}_f, \boldsymbol{\theta}_g)} \frac{1}{M} \sum_{i=1}^{M} \|\mathbf{s}_i - g(f(\mathbf{s}; \boldsymbol{\theta}_f); \boldsymbol{\theta}_g)\|_2^2 \ . \tag{5}$$



Feedback Replay Buffer

- SGD performs update more rapidly than human feedback
- store all human feedback in a buffer: $\mathcal{D} = \left\{(\mathbf{x}_i, \mathbf{y}_j) \mid w(\mathbf{x}_i, \mathbf{y}_j) \neq 0\right\}$
- sample with replacement
- 1000 feedback signals during 15 mins

Deep TAMER

- two-layer, fully connected neural network
- 16 hidden units per layer
- one output node per action

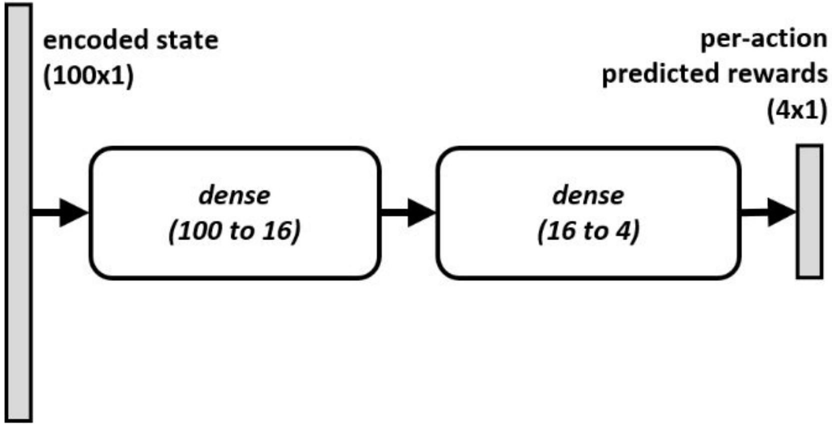
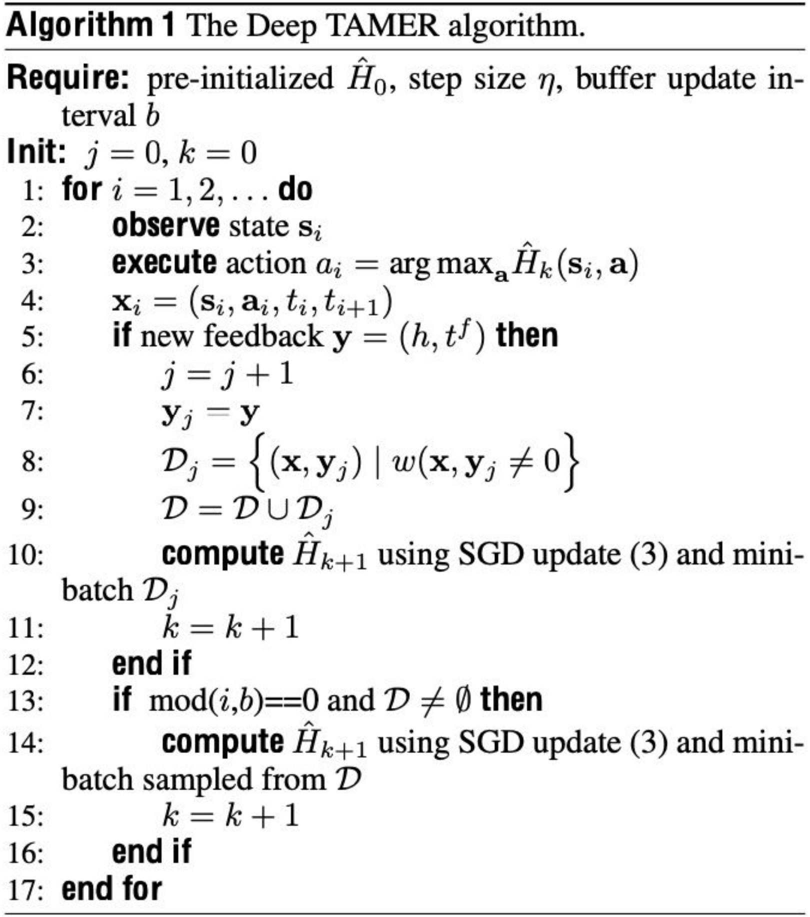


Figure 3: The specific network structure for the fully-connected portion of $\hat{H}$, $z(f(\mathbf{s}), \mathbf{a})$. The action input specifies which component of the output vector is used as the final output of $\hat{H}$.

Deep TAMER

- pre-training the autoencoder
- importance-weighted stochastic optimization
- feedback replay buffer



Algorithm 1 The Deep TAMER algorithm.

```
Require: pre-initialized $\hat{H}_0$, step size $\eta$, buffer update interval $b$
Init: $j = 0, k = 0$
for $i = 1, 2, \ldots$ do
    observe state $\mathbf{s}_i$
    execute action $a_i = \arg\max_{\mathbf{a}} \hat{H}_k(\mathbf{s}_i, \mathbf{a})$
    $\mathbf{x}_i = (\mathbf{s}_i, \mathbf{a}_i, t_i, t_{i+1})$
    if new feedback $\mathbf{y} = (h, t^f)$ then
        $j = j + 1$
        $\mathbf{y}_j = \mathbf{y}$
        $\mathcal{D}_j = \left\{(\mathbf{x}, \mathbf{y}_j) \mid w(\mathbf{x}, \mathbf{y}_j \neq 0\right\}$
        $\mathcal{D} = \mathcal{D} \cup \mathcal{D}_j$
        compute $\hat{H}_{k+1}$ using SGD update (3) and mini-batch $\mathcal{D}_j$
        $k = k + 1$
    end if
    if mod($i$,$b$)==0 and $\mathcal{D} \neq \emptyset$ then
        compute $\hat{H}_{k+1}$ using SGD update (3) and mini-batch sampled from $\mathcal{D}$
        $k = k + 1$
    end if
end for
```

Theory (if relevant)

What are the assumptions made for the theory? Are these reasonable? Realistic?

If the theory build strongly on other prior theory / results, reference those and state them here.

Theory (if relevant, continued)

State main results formally

Give proof sketches

Refer students to the full proofs in paper

Experimental Setup

1-3 slides

Description of the experimental evaluation setting

- What is the domain(s), e.g., datasets, tasks, robot hardware setups?
- What are the baseline(s)?
- What scientific hypotheses are tested?

How did the authors evaluate the success of their approach?

- Clear description of the metrics that will be used

Experimental Results

>1 slide

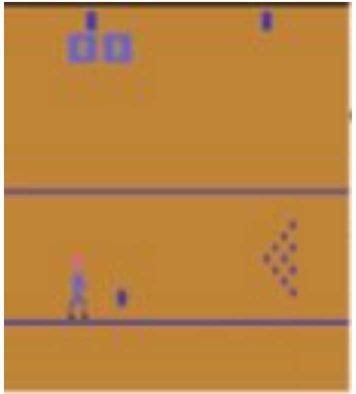
Present the quantitative and qualitative results

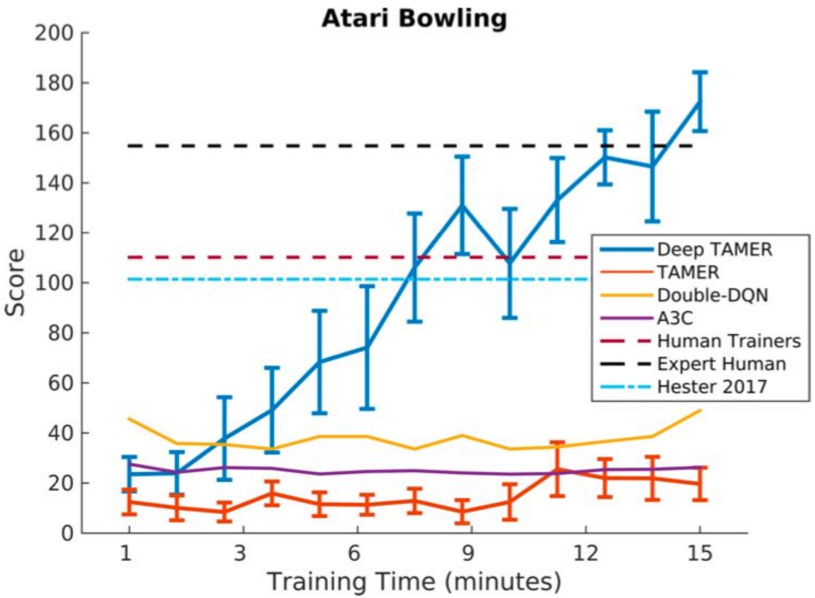
Show figures / tables / plots / robot demos

Pinpoint the most interesting / significant results

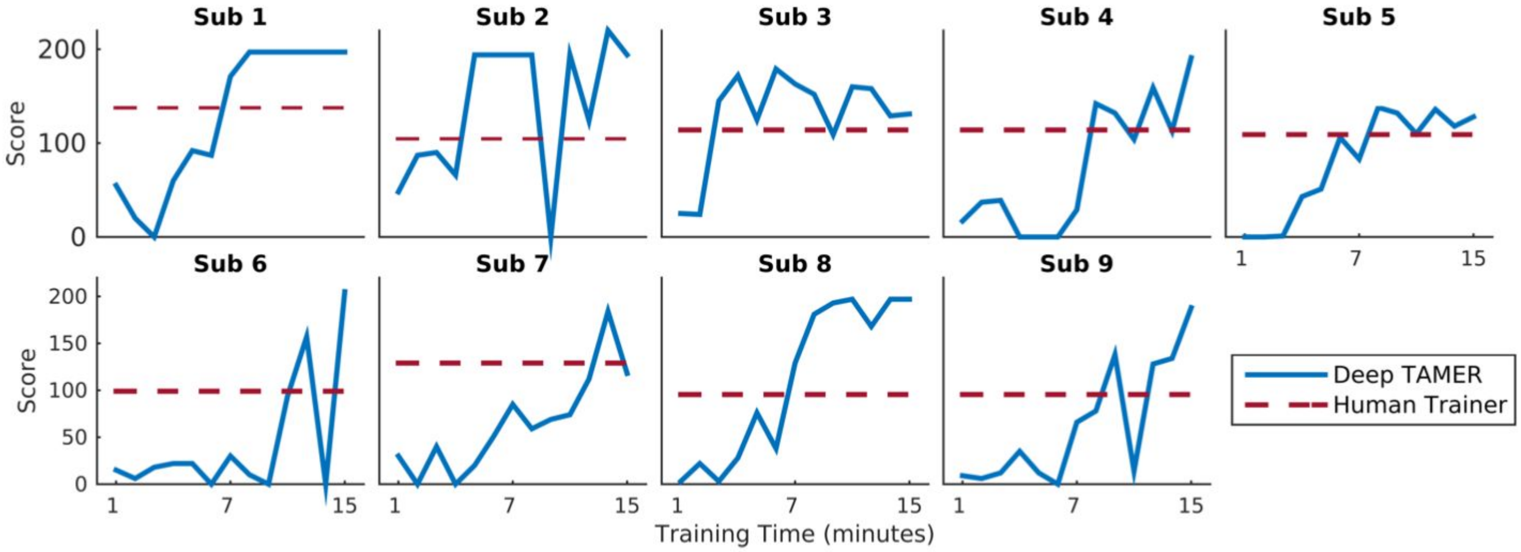
Experimental Results

- Arcade Learning Environment
- 9 trainers trained 15 minutes
- Super-human performance trainer
    - better than expert human





Experimental Results (Ctd.)



Discussion of Results

1-2 slides

What conclusions are drawn from the results by the authors?

- What insights are gained from the experiments?
- What strengths and weaknesses of the proposed method are illustrated by the results?

Are the stated conclusions fully backed by the results and references?

- If so, why? (Recap the relevant supporting evidences from the given results + refs)
- If not, what are the additional experiments / comparisons that can further support/repudiate the conclusions of the paper?

Discussion of Results

- Deep TAMER is the most useful when the task is difficult for a human to perform but not difficult for a human to critique
    - Deep TAMER achieved “super-human performance”
- Better than SOTA methods with less training data
- It is expensive to have human trainers
    - What happens after 15 minutes? Why didn’t they train for longer?
    - Could other methods perform better in the long run?

Critique / Limitations / Open Issues

1-2 slides

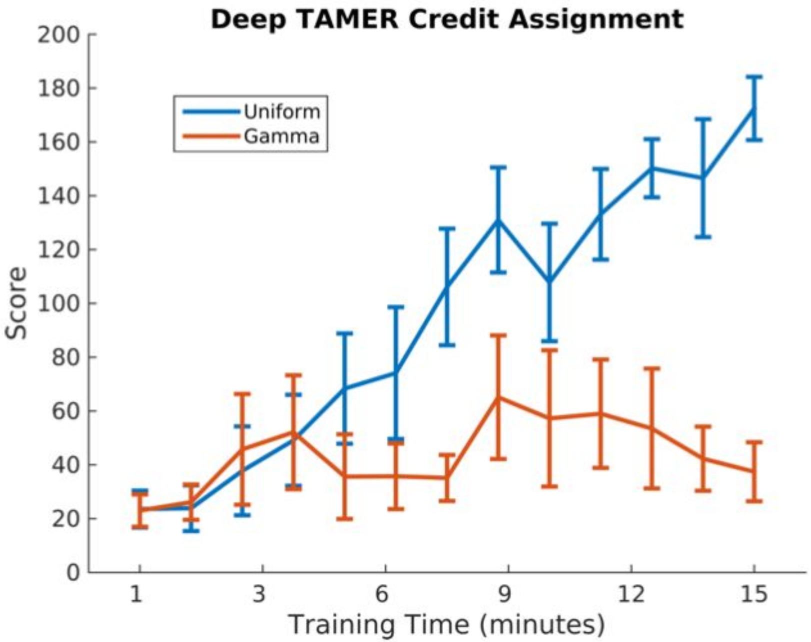
What are the key limitations of the proposed approach / ideas? (e.g. does it require strong assumptions that are unlikely to be practical? Computationally expensive? Require a lot of data?)

Are there any practical challenges in deploying the approach on physical robots in the real world? Are there any safety or ethical concerns of using such approach?

If follow-up work has addressed some of these limitations, include pointers to that. But don’t limit your discussion only to the problems / limitations that have already been addressed.

Importance Weighting Study

- Comparison of different credit assignment distribution
  - Uniform [0.28, 4.0] vs. Gamma(2.0, 0.28)
- Too expensive to explore other hyperparameters searches
- Other hyperparameters are based on tuition
  - # of input frames
  - encoder structure



Critique / Limitations / Open Issues

- Difficult to obtain large amount of human interaction data
- Model is very task-specific
- Hyperparameters choice are based on tuition
- Does not generalize well

Future Work for Paper / Reading

1-2 slides

What interesting questions does it raise for future work?

- ❖ Your own ideas for future work
- ❖ Hyperparameters

Future Work for Paper

- Hyperparameters search
  - variation of DL methods
- Train for longer duration
- Other tasks
  - other Atari games
  - simple robotics task
    - in simulation & real life
- Combination of DEEP TAMER and other SOTA RL

Extended Readings

1-2 slides

Pointers to papers that use this paper as a reference and/or other very related papers that others may want to read

Point classmates to where they can go for further reading on this paper/reading

Summary

1 slide

Approximately one bullet for each of the following

- Problem the reading is discussing
- Why is it important and hard
- What is the key limitation of prior work
- What is the key insight(s) (try to do in 1-3) of the proposed work
- What did they demonstrate by this insight? (tighter theoretical bounds, state of the art performance on X, etc)

Summary - Deep TAMER

- Extension of the TAMER framework
    - learning from real-time human interaction
- Enables success with high-dimensional state spaces
    - deep neural network that approximate the human trainer’s reward function
- Only examined on Atari Bowling
- Accelerates learning and saves training data
    - After just 15 mins of human training performs better than
        - expert human
        - state-of-the-art deep reinforcement learning techniques