From Demonstrations and Knowledge Engineering to a DNN Agent in a Modern Open-World Video Game

Igor Borovikov and Ahmad Beirami

Electronic Arts Digital Platform - Data & AI

{iborovikov, abeirami}@ea.com

AAAI-Make, Stanford, 2019

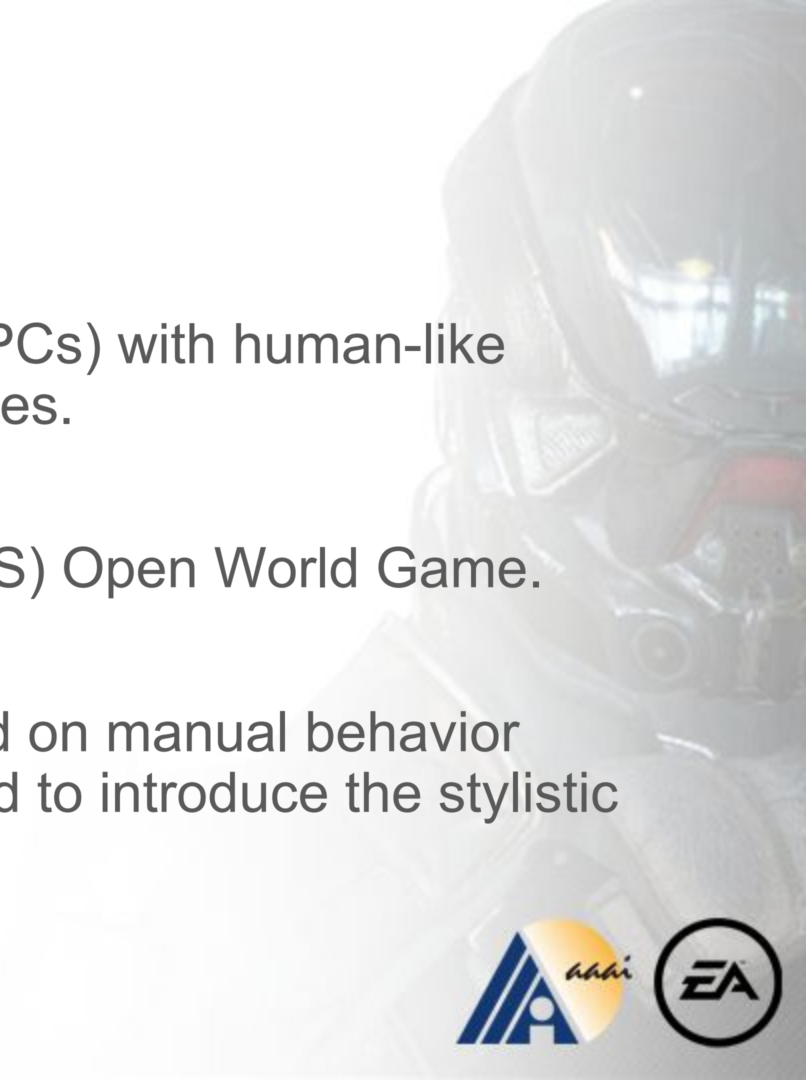
Extended version

Content Overview

1. Motivation: goals and constraints
2. Our approach:

   demonstrations + knowledge engineering => bootstrap => DNN model
3. Results
4. Future work

Motivation

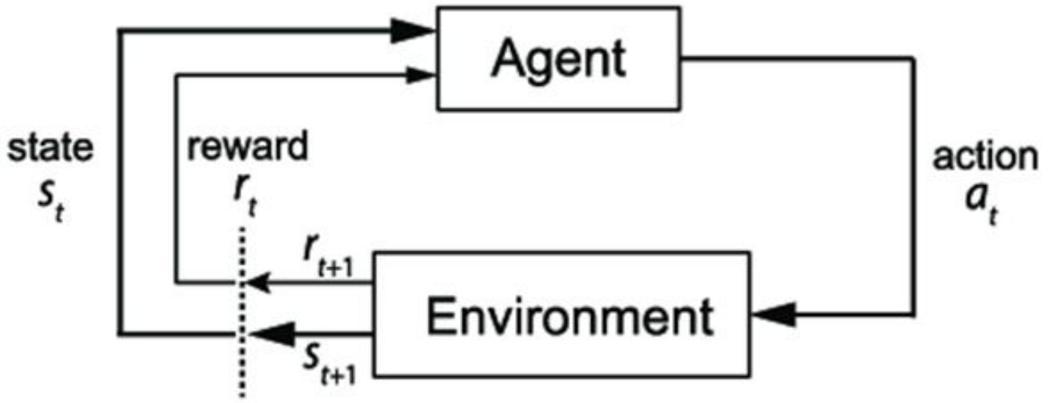


Goal: Train Non-Player Characters (NPCs) with human-like behavior for playing modern video games.

Case Study: First Person Shooter (FPS) Open World Game.

Challenge: Traditional approach based on manual behavior engineering is hard to scale up; it’s hard to introduce the stylistic component.

A Game: Markov Decision Process with an agent in the control loop



[Drawing from Sutton and Barto, Reinforcement Learning: An Introduction, 1998]

Question

Can we use Reinforcement and/or Imitation Learning for training NPCs?

The cutting edge: RL ***solves*** games

Atari games (DQN)

AlphaGo (MCTS)

AlphaZero (meta learning)

Capture the Flag (adaptive learning)

Starcraft II (a combination of everything)

The cutting edge: RL ***solves*** games

Atari games (DQN)

AlphaGo (

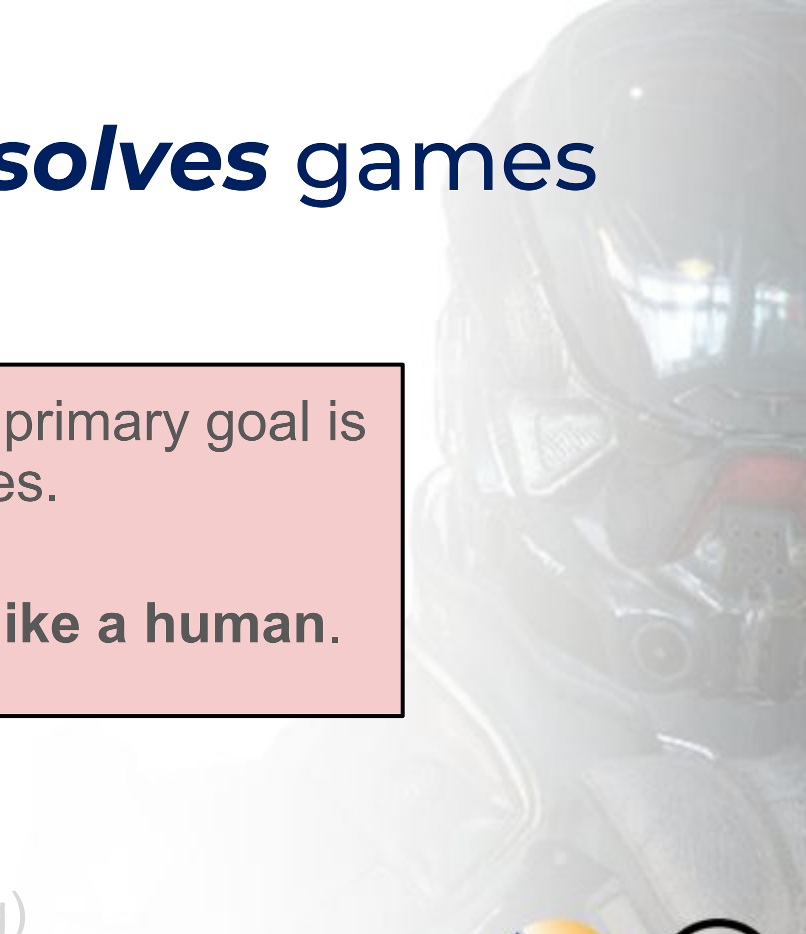
AlphaZero

Capture the Flag (adaptive learning)

Starcraft II (a combination of everything)

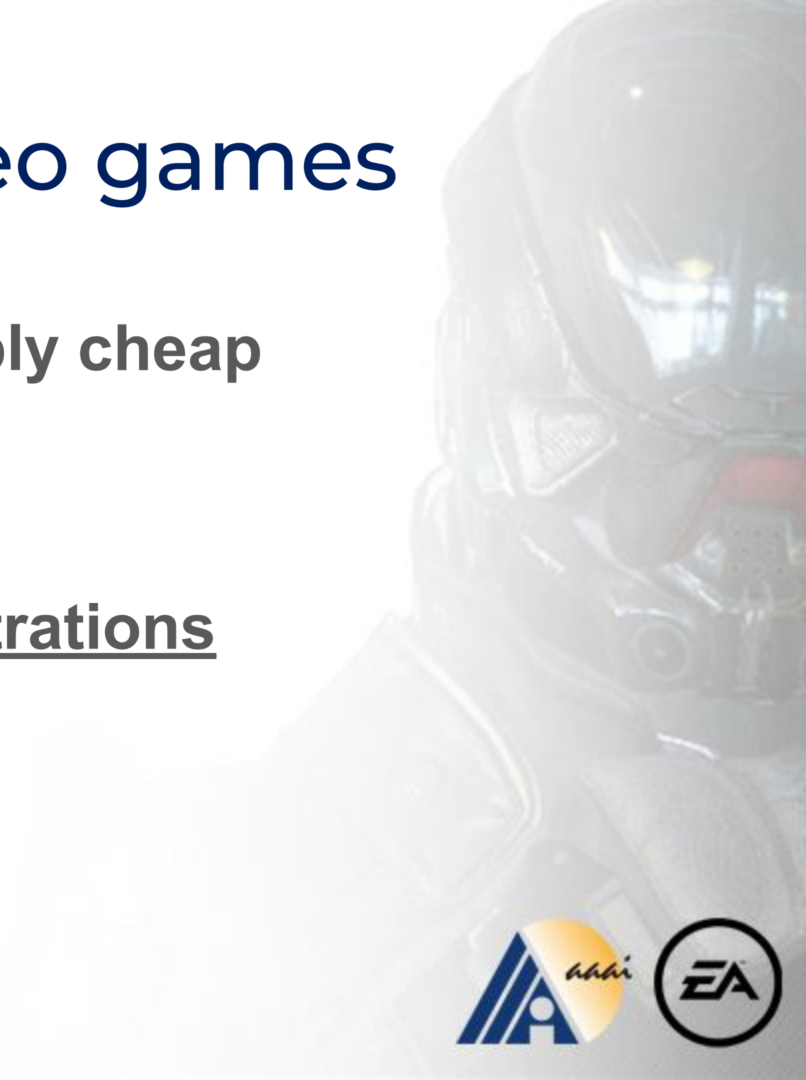
But in a game company our primary goal is to **create**, not to **solve** games.

We need an agent **playing like a human**.



RL challenges for video games

- Training at scale but **reasonably cheap**
- Relatively **quick turnaround**
- Learning from expert **demonstrations**
- Tunability for style/behavior?
- Explainability?

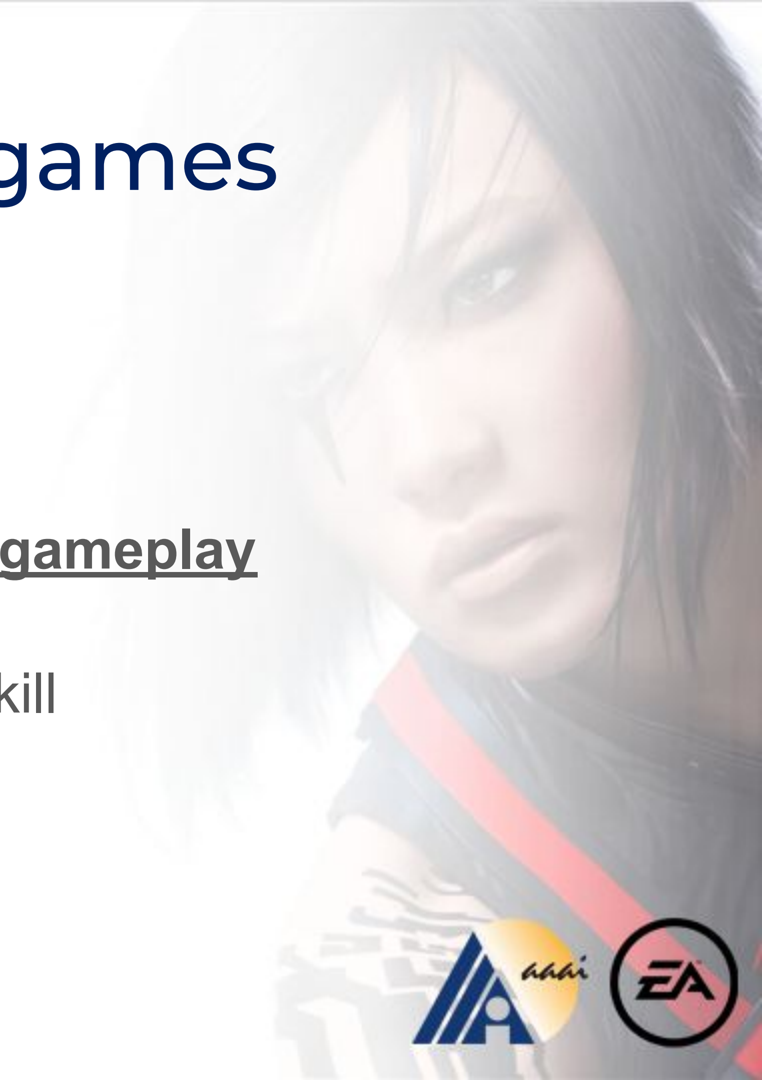


IL challenges in video games

Demonstrations are expensive (humans have to spend time)

<u>They may not cover all aspects of the gameplay</u>

May not deliver the target performance skill

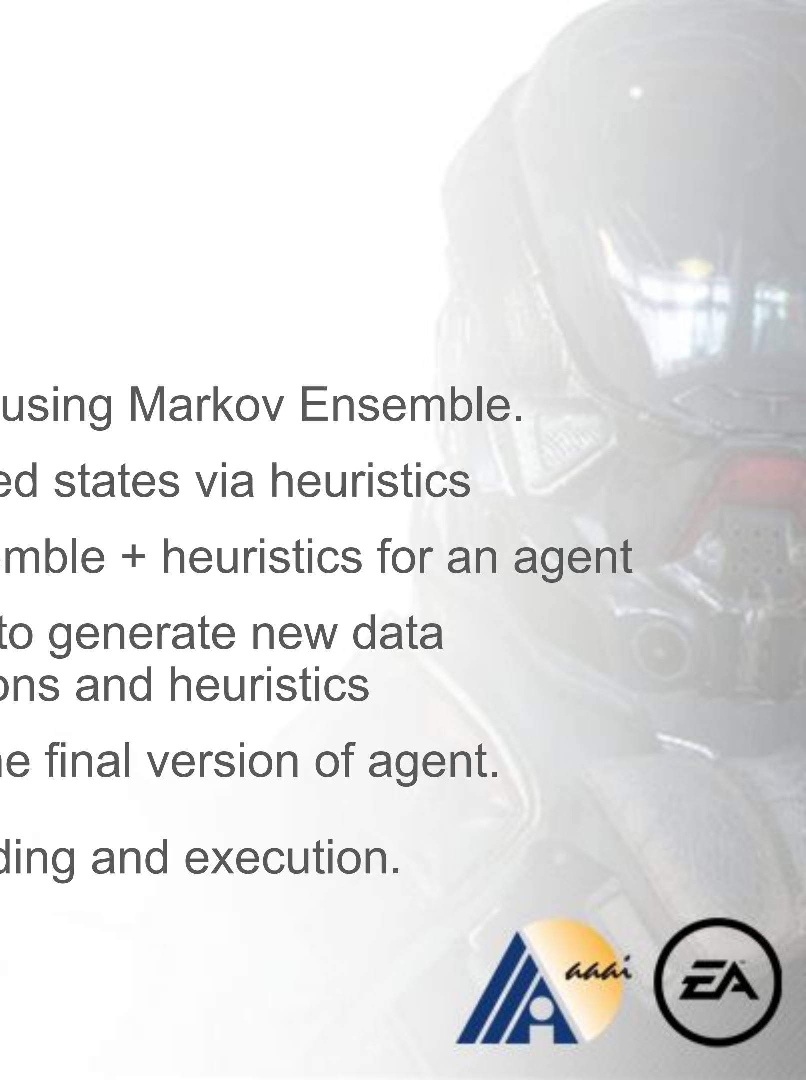


Our game is not a black box (for us)

Direct access to the game code, developers and testers:

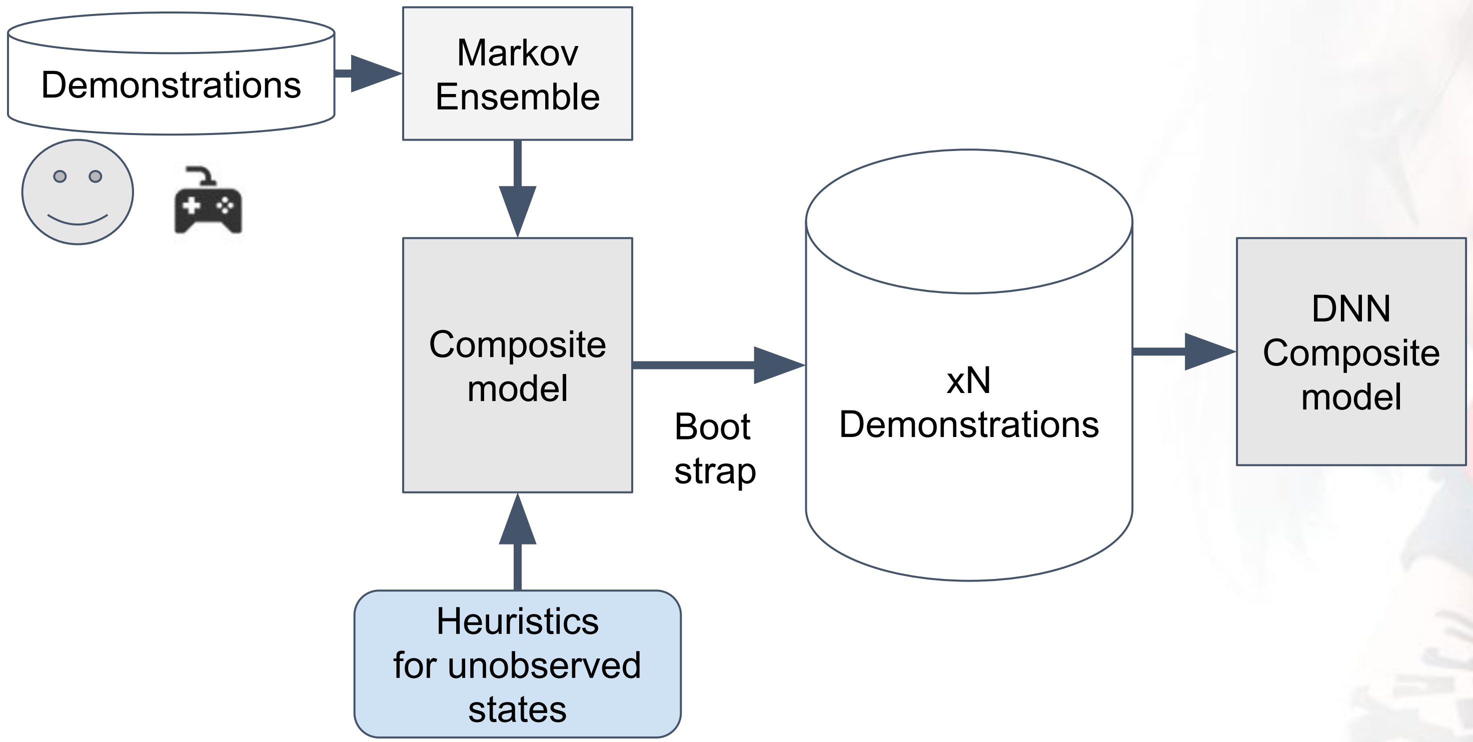
- No need to learn features from rendered images which are in flux due to art changes
- Can extract game state directly as a vector of relevant variables (e.g., character position, orientation, health, ammo, collision probes)
- Obtaining demonstrations is relatively easy and somewhat cheap

Our approach



1. Instrument the game (no frame buffer!)
2. Build a base model from demonstrations using Markov Ensemble.
3. Address undefined behavior in unobserved states via heuristics
4. Construct composite model Markov Ensemble + heuristics for an agent
5. The composite model controls the game to generate new data integrating both generalized demonstrations and heuristics
6. Use new data to train a DNN model for the final version of agent.
7. Single DNN model gives speed up in loading and execution.

Our approach



Markov Ensemble: Range of Orders

Demonstration is a sequence of recorded states s_i and actions a_i

$$S=\{(s_i,\ a_i)\}_{i=1,\ldots,N}$$

Markov Model M_m of order m defines probability of transitions:

$$(s_i,\ a_i,\ldots,a_{i-m}) \rightarrow a_{i+1}$$

m is in the range from 0 to ~30 i.e. covers up to 1 sec of gameplay

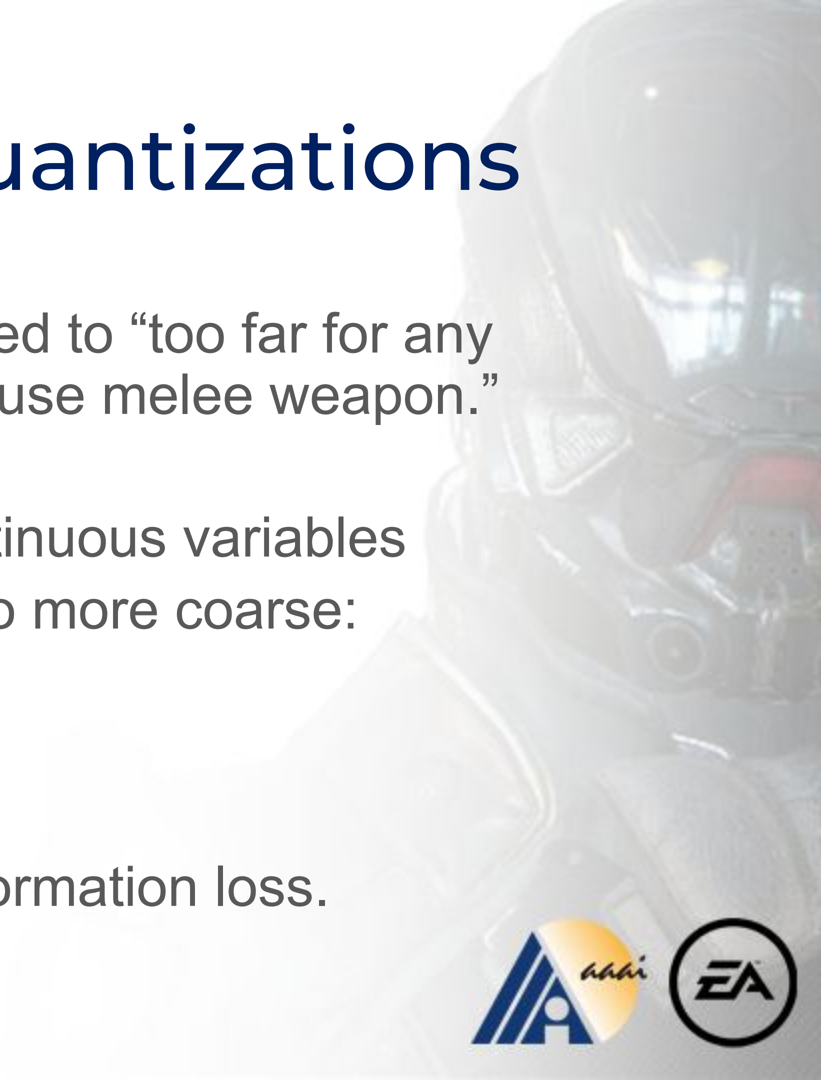
Markov Ensemble: Quantizations

E.g., distance can be roughly discretized to “too far for any action”, “can use sniper rifle”, …, “can use melee weapon.”

A hierarchy *Q** of quantizations of continuous variables in (s_i, a_i) from low error (fine grained) to more coarse:

$$Q^* = \{Q_0, \ldots, Q_k\}$$

The coarsest level (Q_0) = complete information loss.

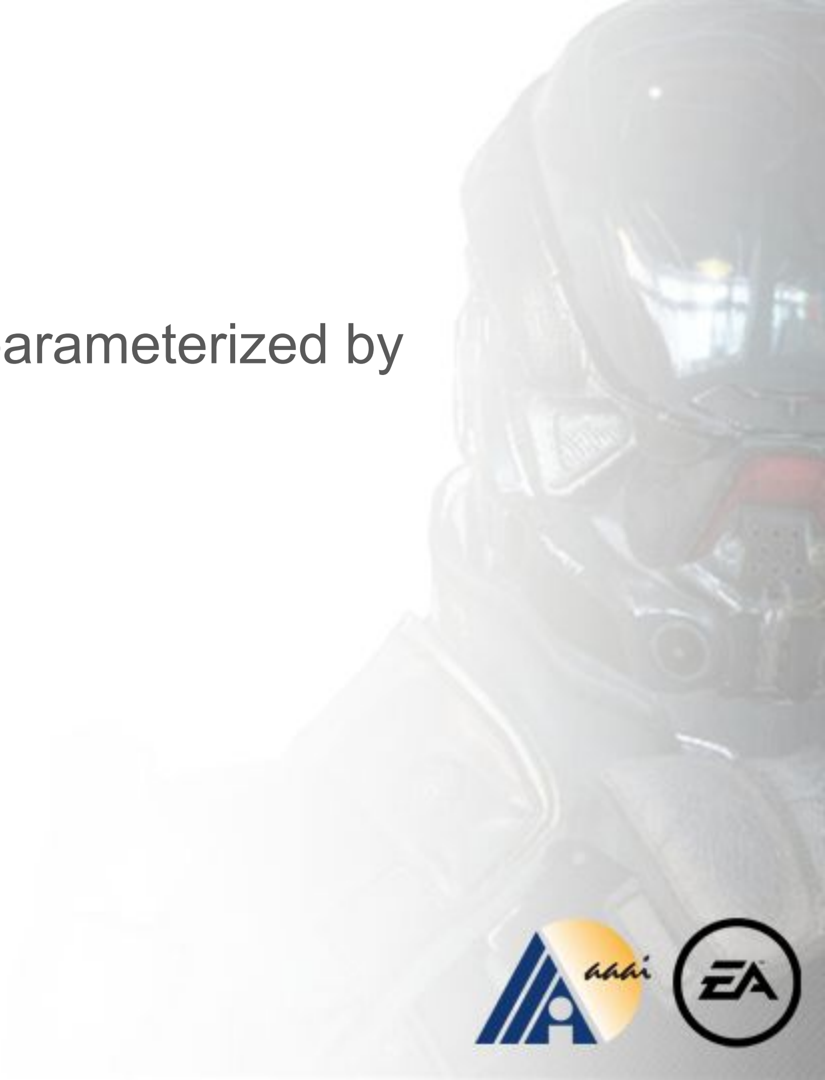


Markov Ensemble

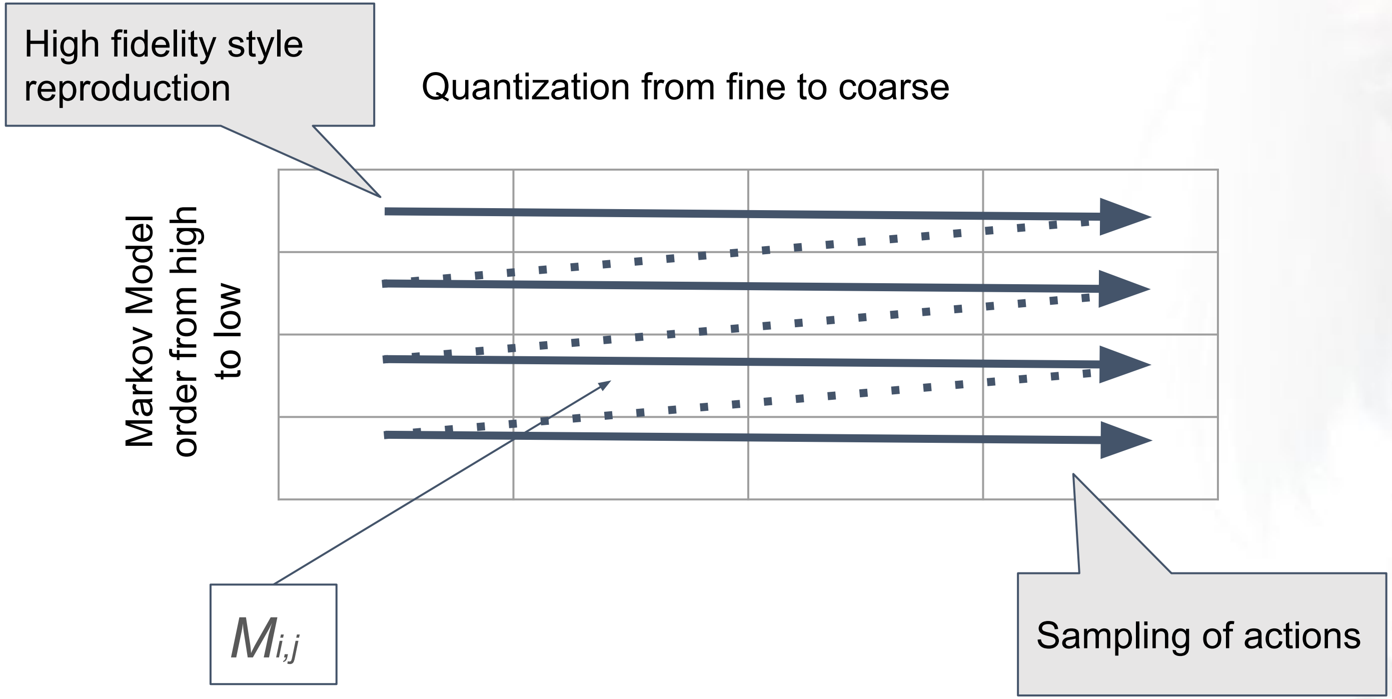
Markov Models from demonstrations parameterized by

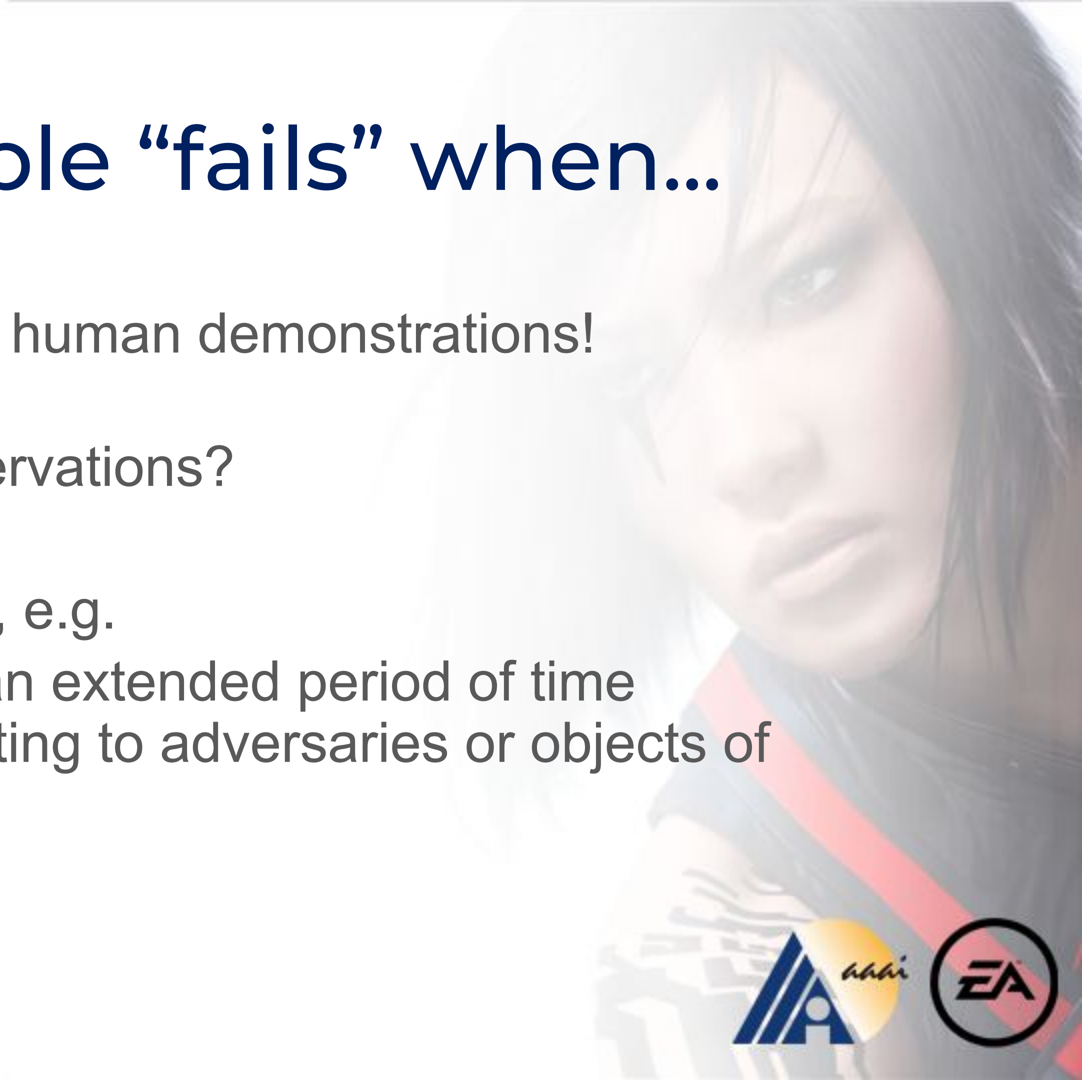
- Order $m=0,...,N$
- Quantization $k=0,...,K$

$$M_{m,k}\,(Q_k(s,\ a_i,...,a_{i-j})) \longrightarrow a_{i+1}$$



Markov Ensemble


High fidelity style reproduction
Quantization from fine to coarse
Markov Model order from high to low
$M_{i,j}$
Sampling of actions



Markov Ensemble “fails” when…

A state was not observed in human demonstrations!

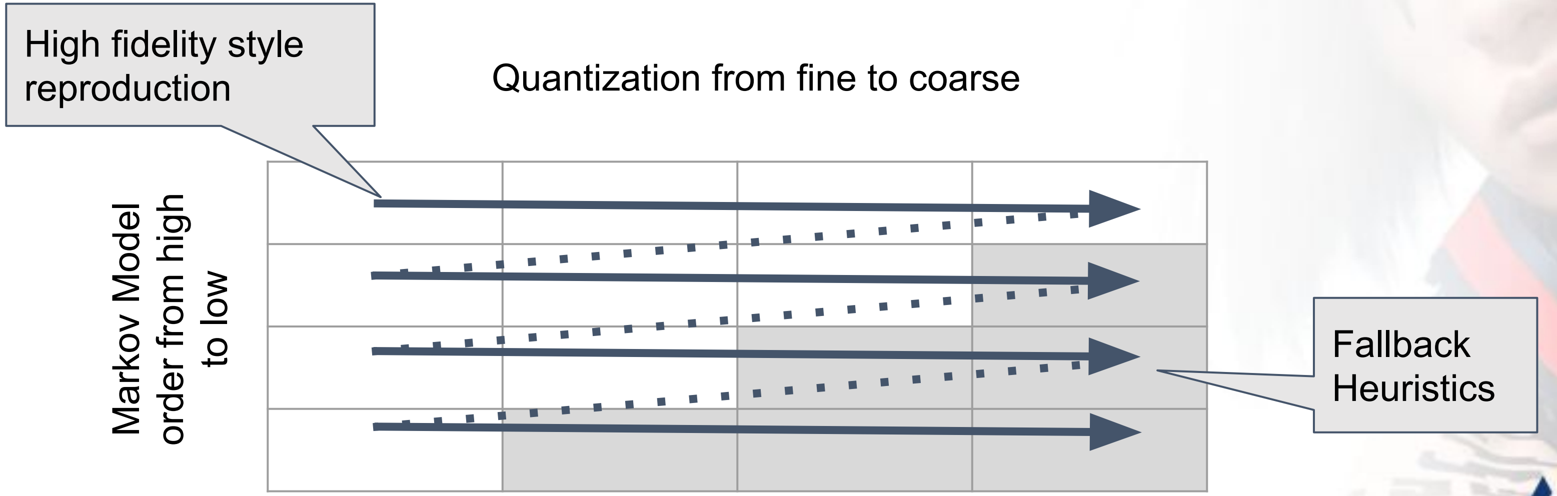
Why a state was not in observations?

Humans proactively avoid it, e.g.

- Running into a wall for an extended period of time
- Wandering without reacting to adversaries or objects of interest

Solution

- Limit Markov Ensemble to finer quantization and longer actions sequences.
- Provide a fallback as a heuristic.


High fidelity style reproduction
Quantization from fine to coarse
Markov Model order from high to low
Fallback Heuristics

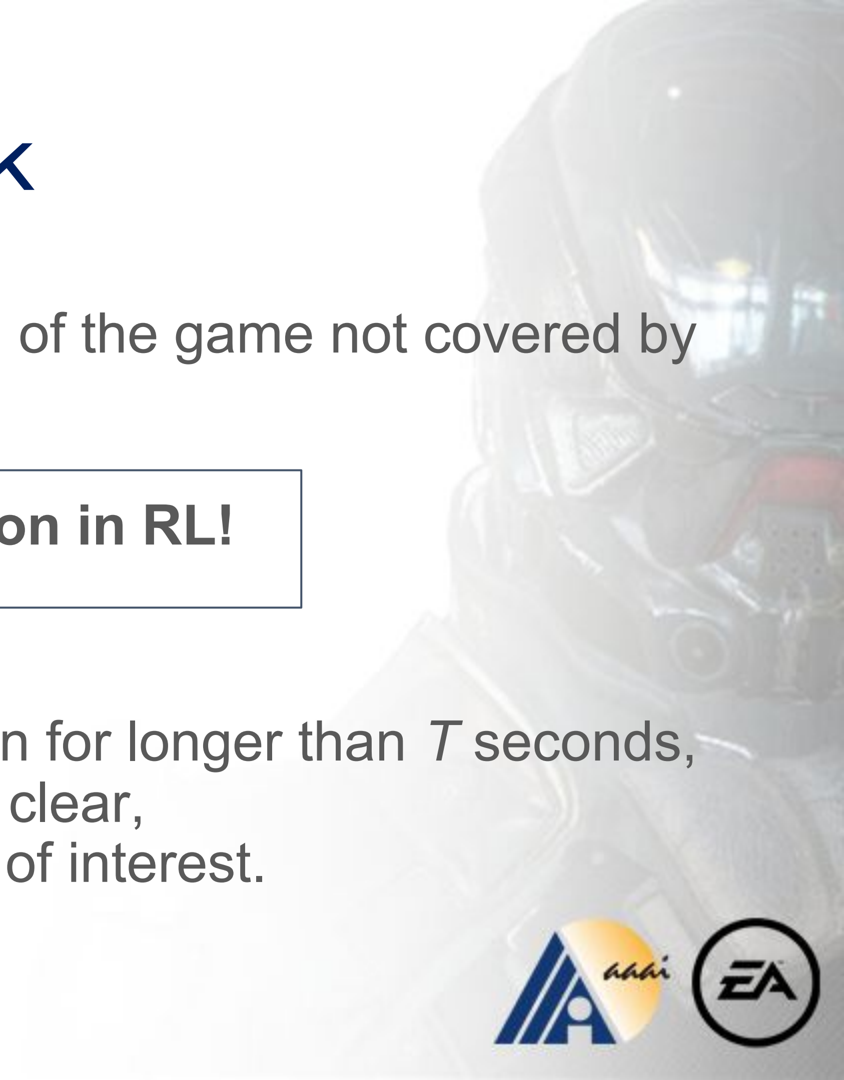
Heuristics for fallback

Capture **intuitive human knowledge** of the game not covered by demonstrations.

Too costly to discover by exploration in RL!

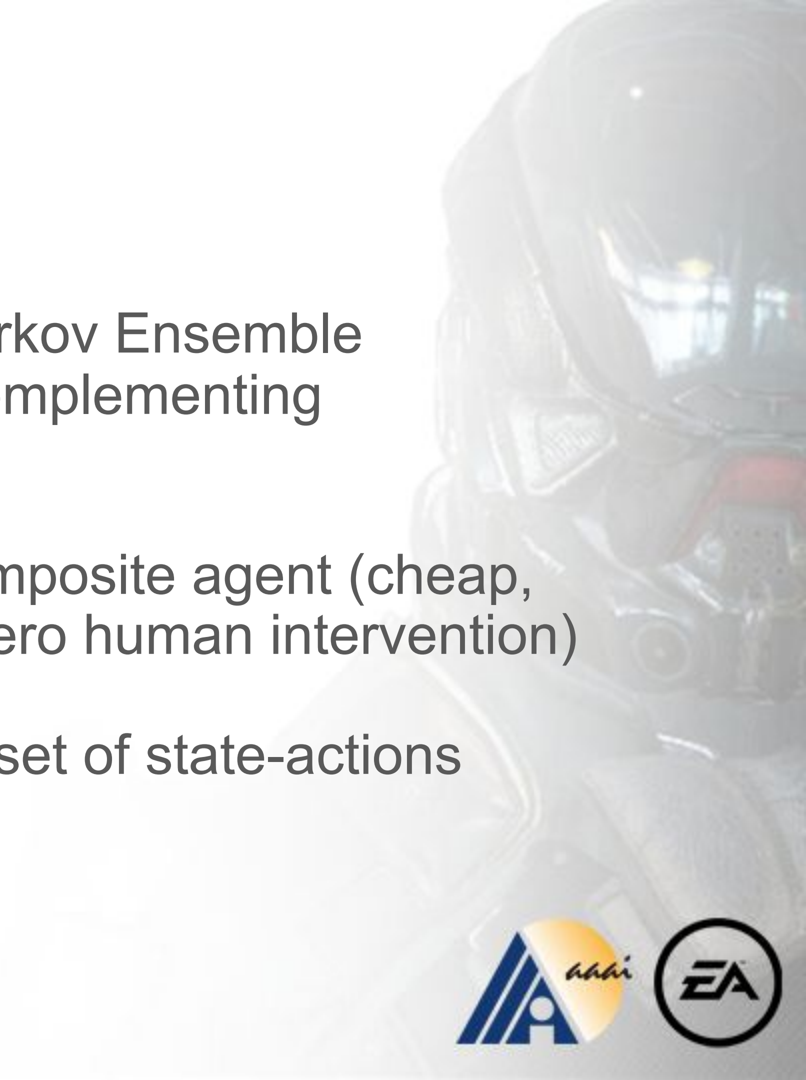
Examples:

- If blocked in the direction of motion for longer than *T* seconds, keep turning until new direction is clear,
- Turn to an adversary or an object of interest.



Bootstrap

- Build an agent which combines Markov Ensemble (demonstrations) and heuristics (complementing demonstrations)
- Run the game controlled by the composite agent (cheap, uses only computing time, nearly zero human intervention)
- Collect the new bootstrapped data set of state-actions



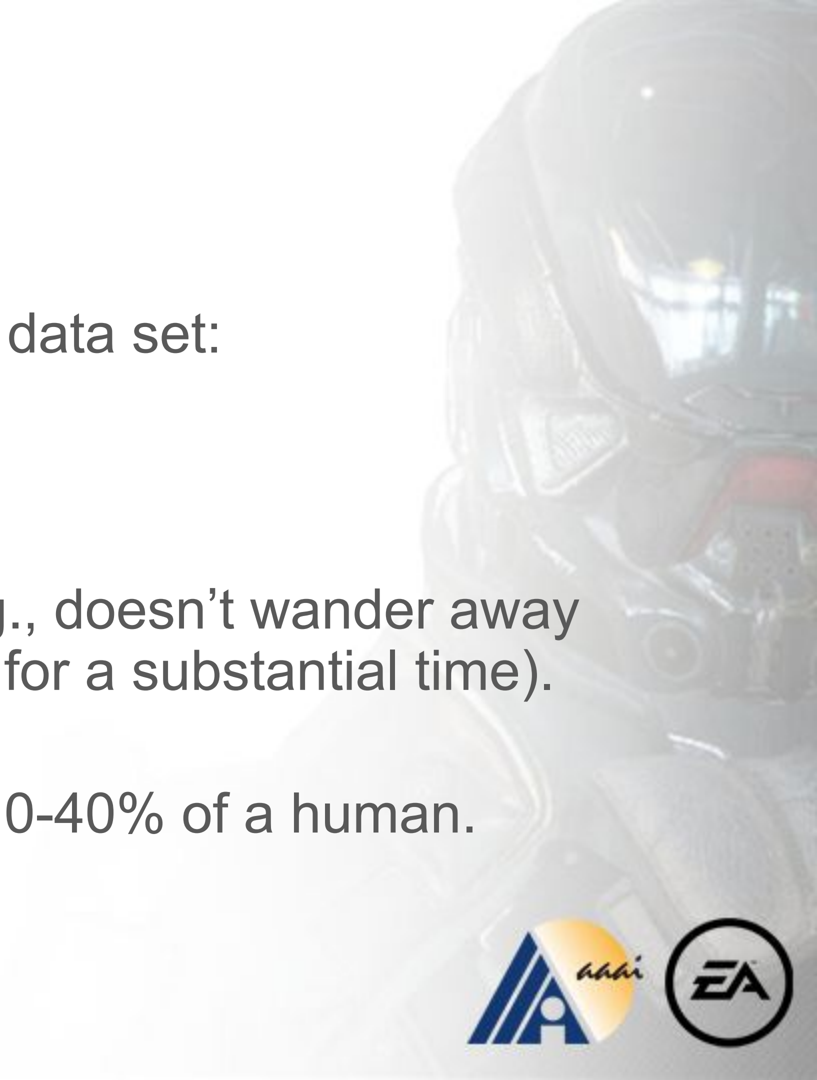
Train the final model

Train a DNN model from bootstrapped data set:

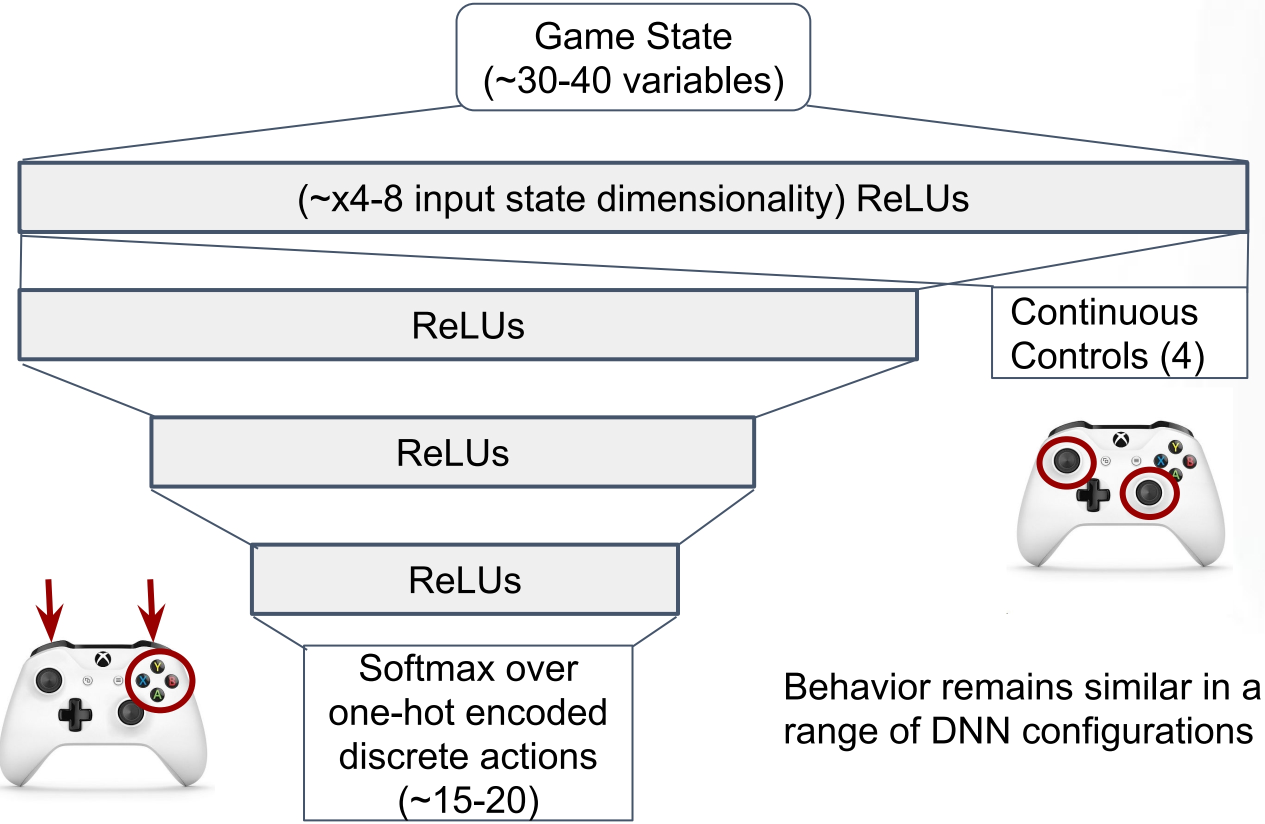
$$DNN(s_i, a_i, \ldots, a_{i-m}) \rightarrow a_{i+1}$$

Results in a “reasonable” behavior (e.g., doesn’t wander away from adversary, doesn’t run into a wall for a substantial time).

Performance: kill-death-ratio is about 10-40% of a human.

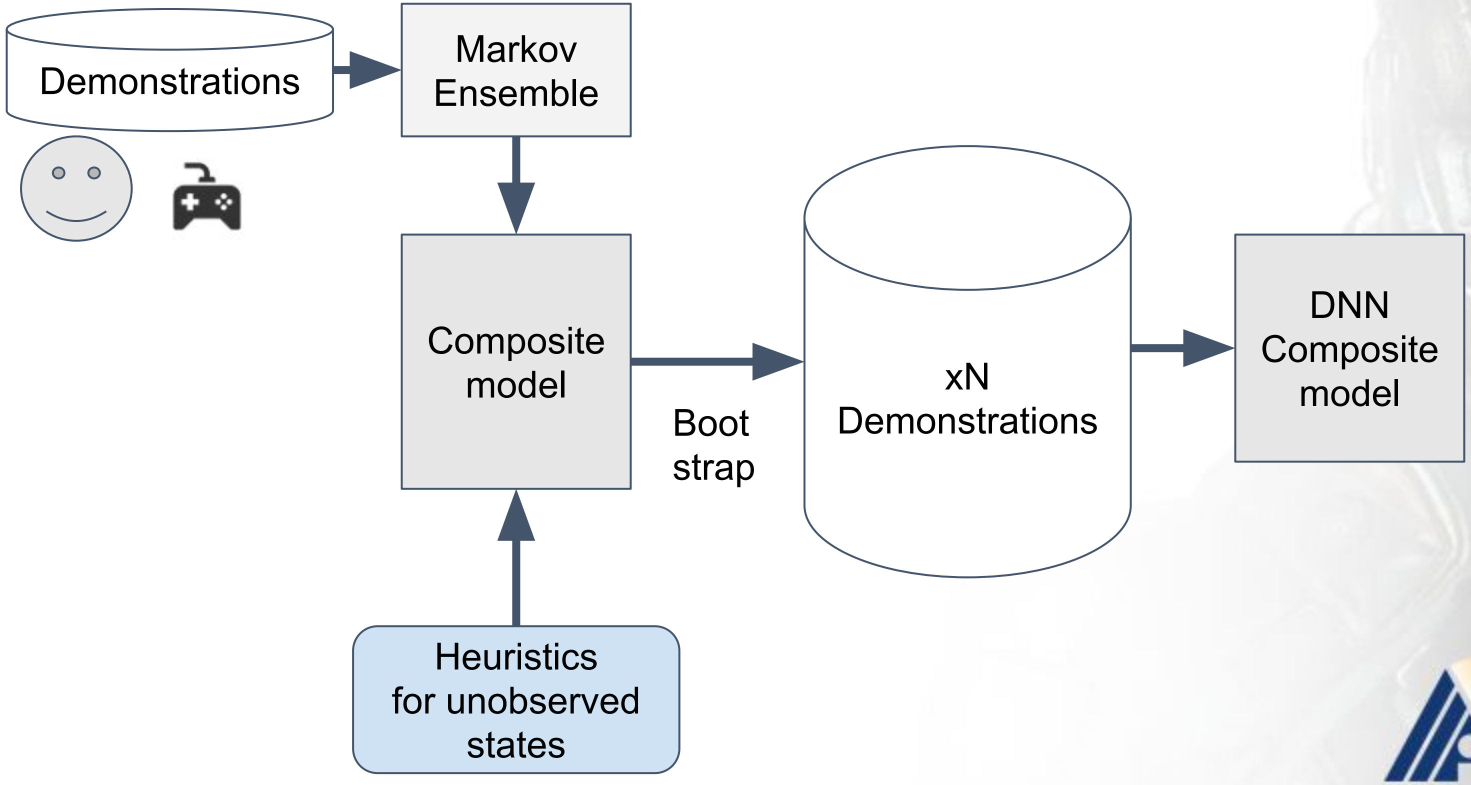


DNN (sketch)

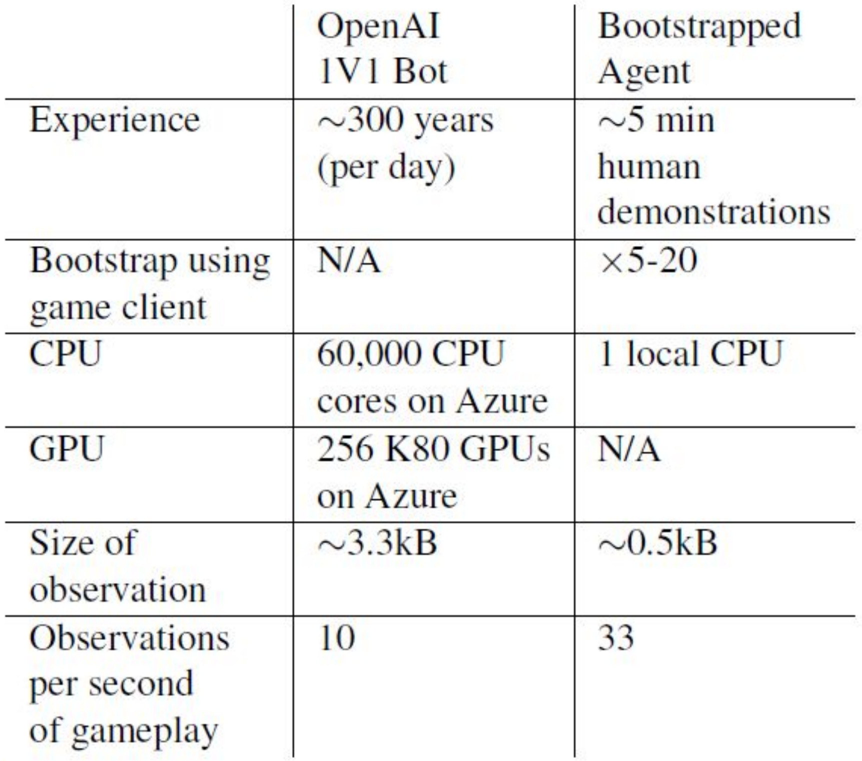

Game State
(~30-40 variables)
(~x4-8 input state dimensionality) ReLUs
ReLUs
Continuous Controls (4)
ReLUs
ReLUs
Softmax over one-hot encoded discrete actions (~15-20)

Behavior remains similar in a wide range of DNN configurations

Summary of the Approach



Comparing to end-to-end RL



	OpenAI 1V1 Bot	Bootstrapped Agent
Experience	∼300 years (per day)	∼5 min human demonstrations
Bootstrap using game client	N/A	×5-20
CPU	60,000 CPU cores on Azure	1 local CPU
GPU	256 K80 GPUs on Azure	N/A
Size of observation	∼3.3kB	∼0.5kB
Observations per second of gameplay	10	33

Comparison between OpenAI 1V1 Dota 2 Bot (OpenAI Five 2018) training metrics and training an agent from human demonstrations, programmed rules, and bootstrap in a proprietary open-world first-person shooter. While the objectives of training are different, the environments are somewhat comparable. The metrics illustrate the practical advantages of the proposed technique.

Note: This is not apples to apples!

Future Work

- Style measure?
- Improve performance of the resulting model
- Apply to genres different from FPS

Selected Q&A from the live discussion

Q: How much knowledge engineering went into designing quantization schemes?

A: We found experimentally that the choice of quantization in a wide range of parameters was not critical to the overall performance of the system. As such, we could get away with being off by a factor of x2 for the thresholds values. Meaning, our quantization - both threshold values and the number of the thresholds - could be "eyeballed." Hence, the answer: only a very modest amount of efforts went into knowledge engineering for quantization and the domain "knowledge" itself was not very deep. However, a proper answer warrants more scientific approach, e.g., we need to understand what are the optimal quantization schemas for the proposed method. That will be part of our future work.

Selected Q&A from the live discussion

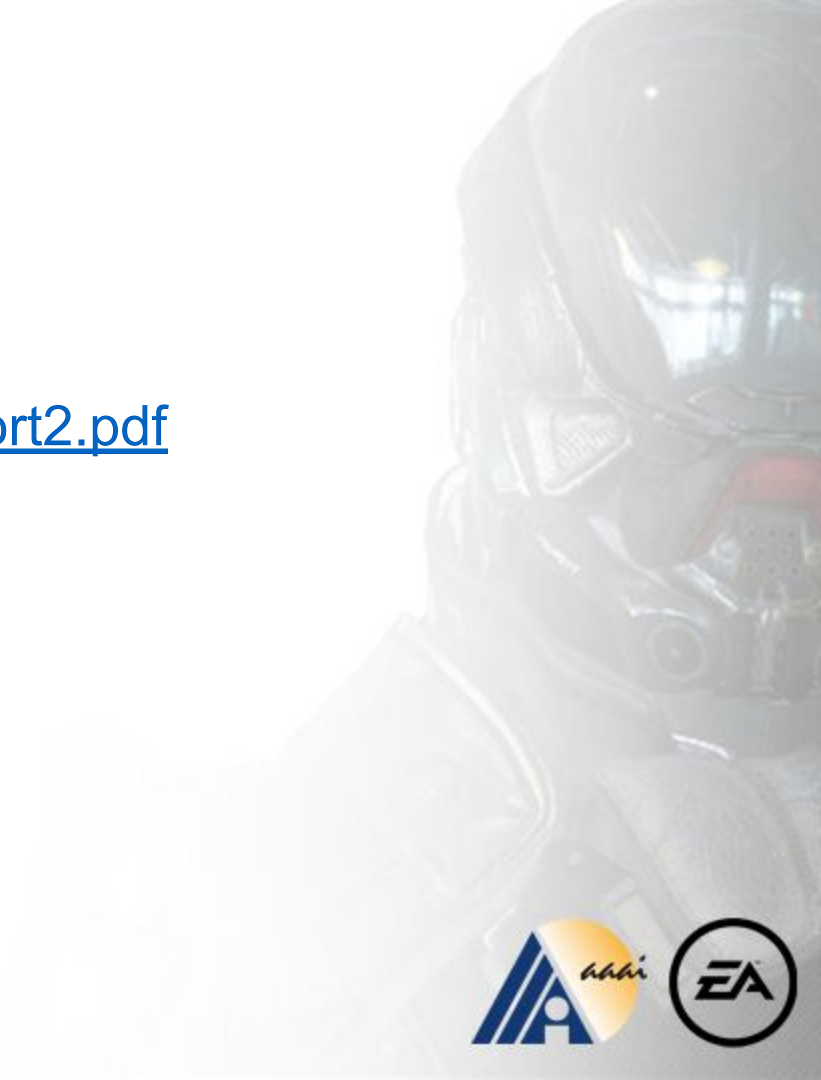
Q: Can the trained agent improve its performance in the game by interacting with it?

A: A short answer is - "no," at least not by itself. That is a well-known limitation of plain vanilla behavior cloning and baseline imitation learning. However, since we can utilize player telemetry data while (s)he improves game-playing skills, we can iterate on the entire training cycle, as described in the talk, and include latest data with improved performance of the teacher. That new data will improve the performance of the agent. Training as the whole process is fast and inexpensive, which partially compensates for the lack of the agent being able to learn by itself (to the extent that we implemented interactive learning, see “Winning isn’t everything” from the references list).

References

Available from the paper

https://proceedings.aaai-make.info/short2.pdf



Thank you!

Igor Borovikov, Ahmad Beirami
Electronic Arts, Digital Platform - Data and AI
{iborovikov, abeirami}@ea.com