Autonomous Agents for Complex Simulations: A Compute-Efficient Imitation Learning Approach

Julian Togelius, Raz Saremi

NYU

modl.ai

What am I doing here?

Constructing Human-like Agents for Complex Systems

- Many complex simulations (socio-technical systems) require human-like agents
  - Opponents, team-mates, instructors, etc.
- These could be constructed in various ways:
  - Elaborate manual construction (labor-intensive)
  - Reinforcement learning (not very human-like)
  - Imitation learning
- Many constraints
  - Data, training time, inference time

Tactical First-Person Shooters

The Mission

Demonstrate Human-like bots for use as teammates and opponents in a Tactical FPS game

Using ML to Create Agents

- Our agents learn from human players instead of relying solely on scripted behavior
- Minimize AI programmer upkeep costs at game updates
- Can train different agents for different skill-levels in the player base and challenge players at the right level

The Game: VAL-lite

- Built over Unreal's Lyra FPS Demo
- Intended as a "Valorant-lite"
  - **Not a Valorant clone**

Core VAL-Lite Features

- 2v2 team gameplay
- 2 characters based on the **Initiator** and **Controller** roles in Valorant
- Bomb-planting game mode
- Simplified version of **Ascent** map
  - 1 bomb site
  - Removed some areas from map
- 1 weapon (Pistol)

- **Initiator** character has:
  - A blinding grenade
  - An ability suppression grenade
  - Inspired by **KAY/O**
- **Controller** character has**:**
  - An incendiary grenade
  - A smokescreen/sphere grenade
  - Inspired by **Brimstone**

Some Constraints

- Inference needs to fit inside the core game update loop
  - A few milliseconds at most
- Needs to work on servers without a GPU
  - Inference on CPU
  - Ideally, should fit into CPU cache
  - Pixels? Fuhgeddaboutit
- Should work with a variety of different playing styles
- Ideally, architecture and sensors should transfer to other similar games

Using ML to Create Bots

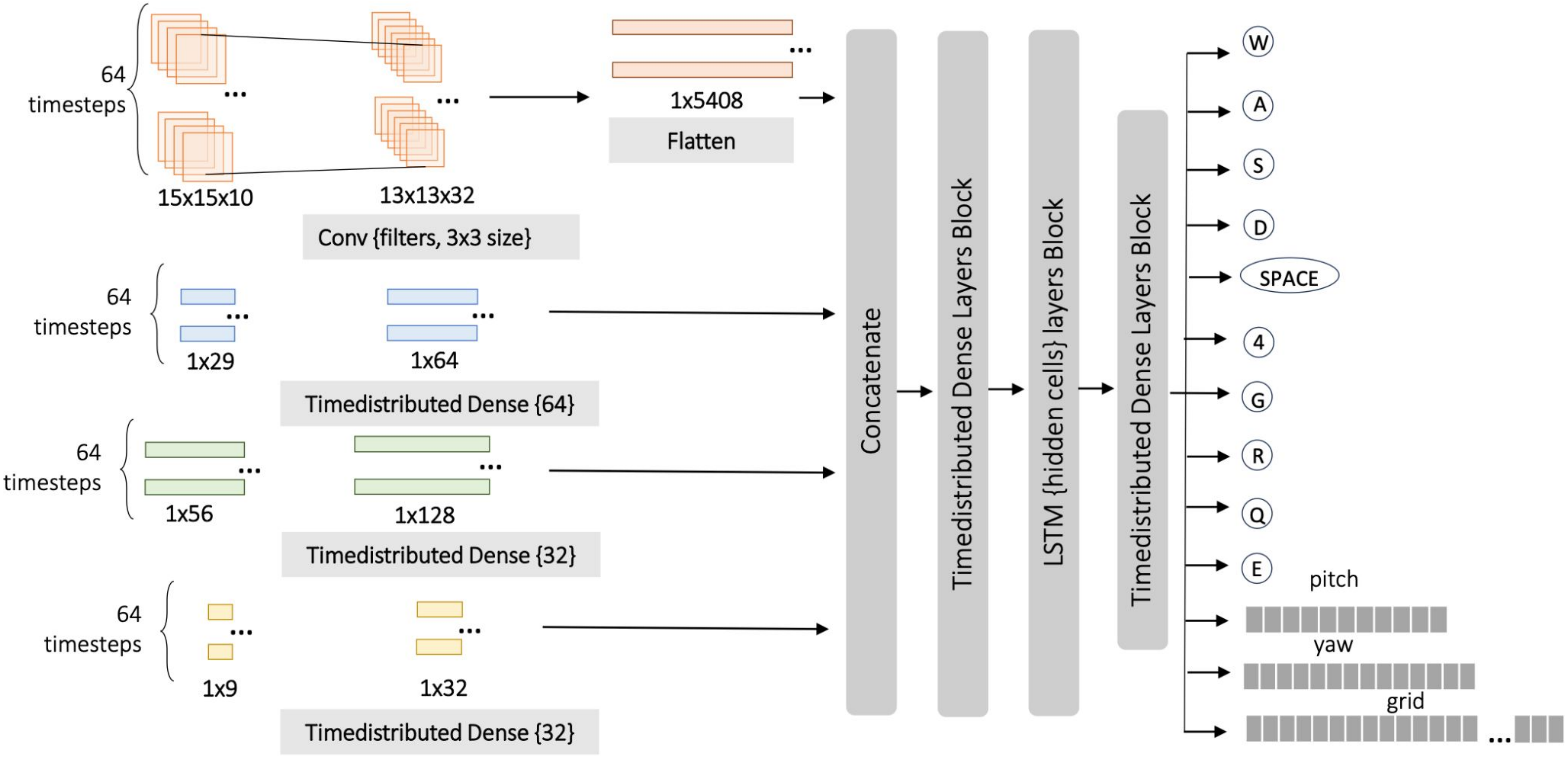
How we did it: Data

- Collected gameplay data from players
  - Relatively wide skill range
  - Allows us to tailor bot skill
- No pixels! Only sensors (see later slide)
  - Deployable cheaply in production
- Possible agent behaviors, strategies, skill levels, and adaptations scale with the available data
- More data = better and more diverse agents
  - Currently trained on <50 hours of play time

Agents in action

Model Architecture Details

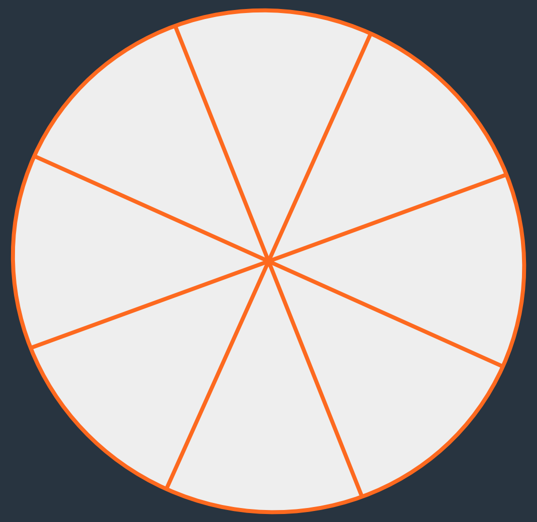
- Convolutional LSTMs
- 64 time-steps
- 4 input streams:
  - State features
  - Sensor grids
  - Audio sensors
  - Directional and distance features
- 2 LSTM layers -> 1 Dense layer
- ~24.5 million parameters
- Base model trained for ~300 epochs
- Expert and Novice models fine-tuned for ~250
  - On top and bottom 75% of data
- Full model trained for another ~1000
  - On full data set



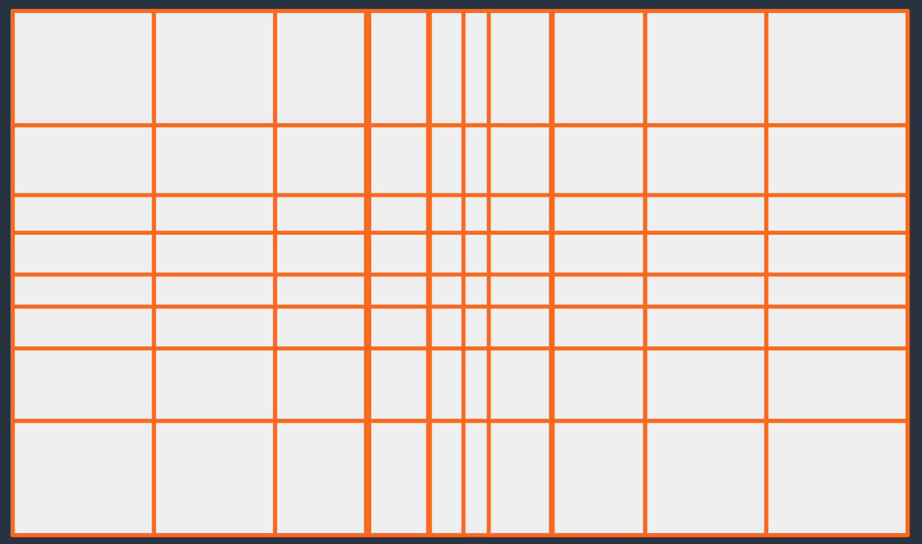
64
timesteps
15x15x10
13x13x32
Conv {filters, 3x3 size}
1x5408
Flatten
64
timesteps
1x29
1x64
Timedistributed Dense {64}
64
timesteps
1x56
1x128
Timedistributed Dense {32}
64
timesteps
1x9
1x32
Timedistributed Dense {32}
Concatenate
Timedistributed Dense Layers Block
LSTM {hidden cells} layers Block
Timedistributed Dense Layers Block
W
A
S
D
SPACE
4
G
R
Q
E
pitch
yaw
grid

Sensors

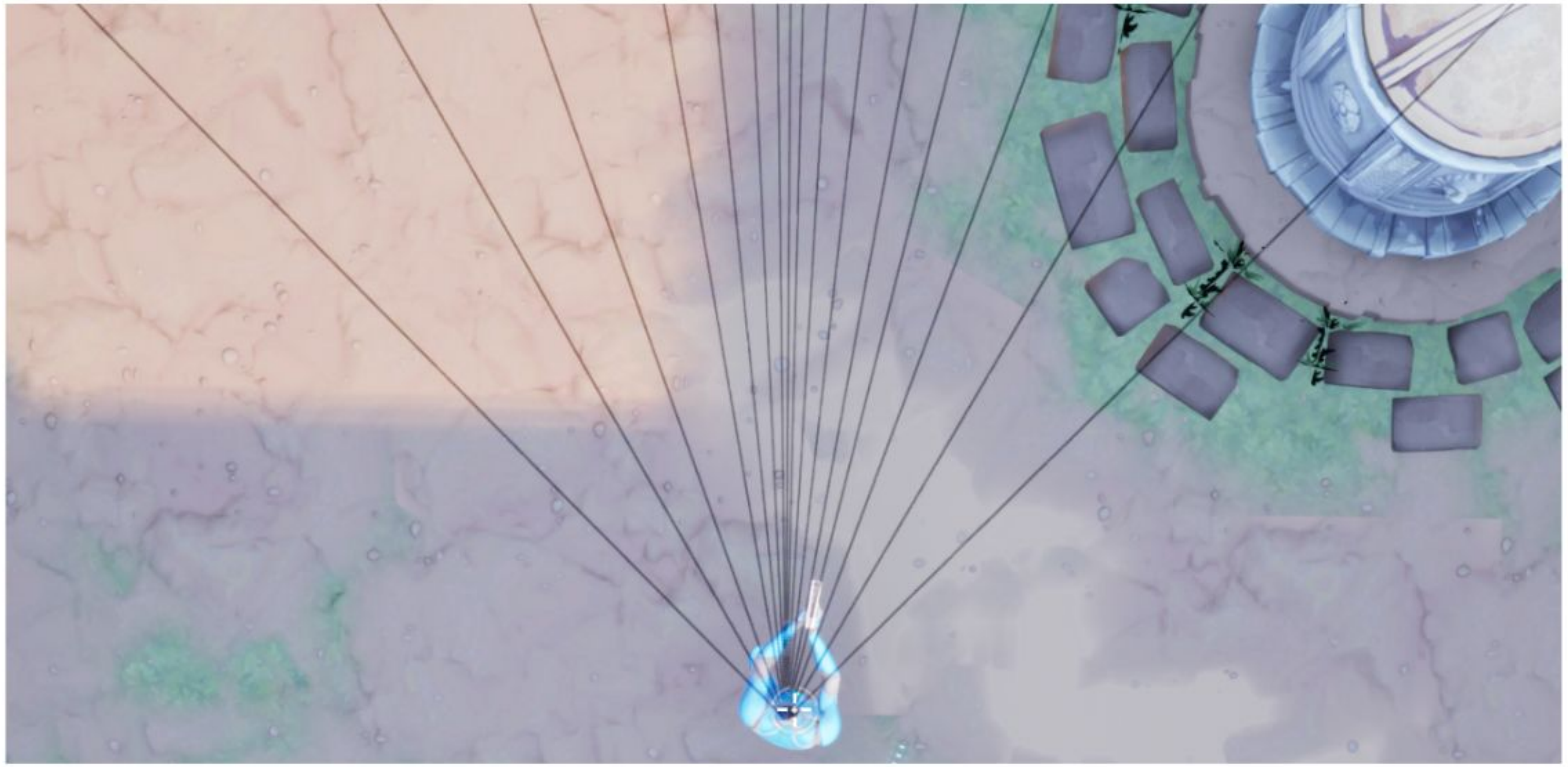
- Audio (directional)
  - Distance to source
  - **Shots fired**
  - **Footsteps**
  - **Grenade bouncing**
  - **Grenade exploding**
  - **Bomb beeping**



- Visual (15x15 sensors for each)
  - Concentrated around center
  - **Walls**
  - **Enemies**
  - **Teammate**
  - **Bomb (dropped, planted)**
  - **Grenade (team, enemy)**
  - **Bombsite**
  - **Smoke**
  - **Incendiary**



Sensors, visualized



(a) Horizontal rays.

(b) Vertical rays.

Look ma, no head!

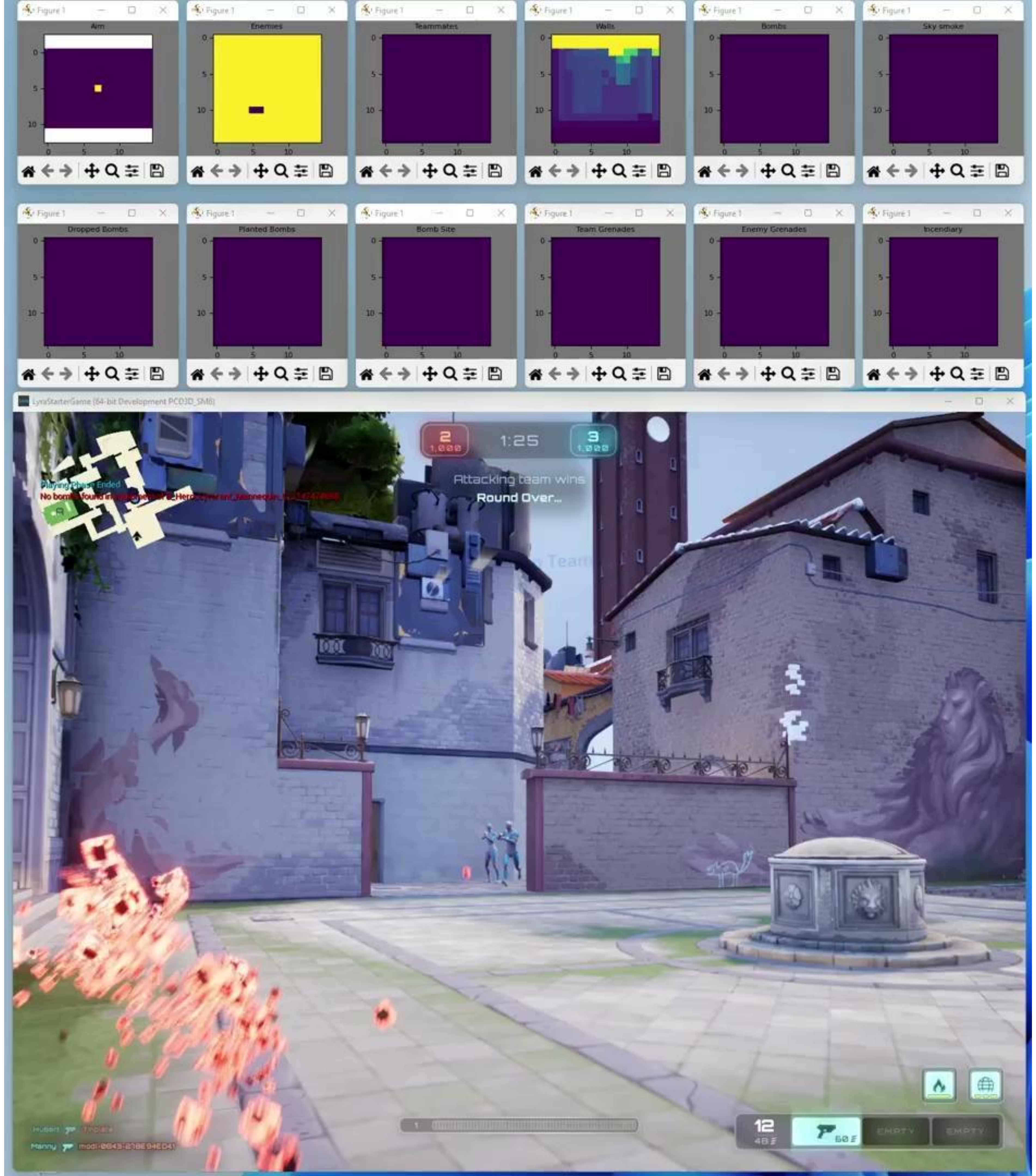
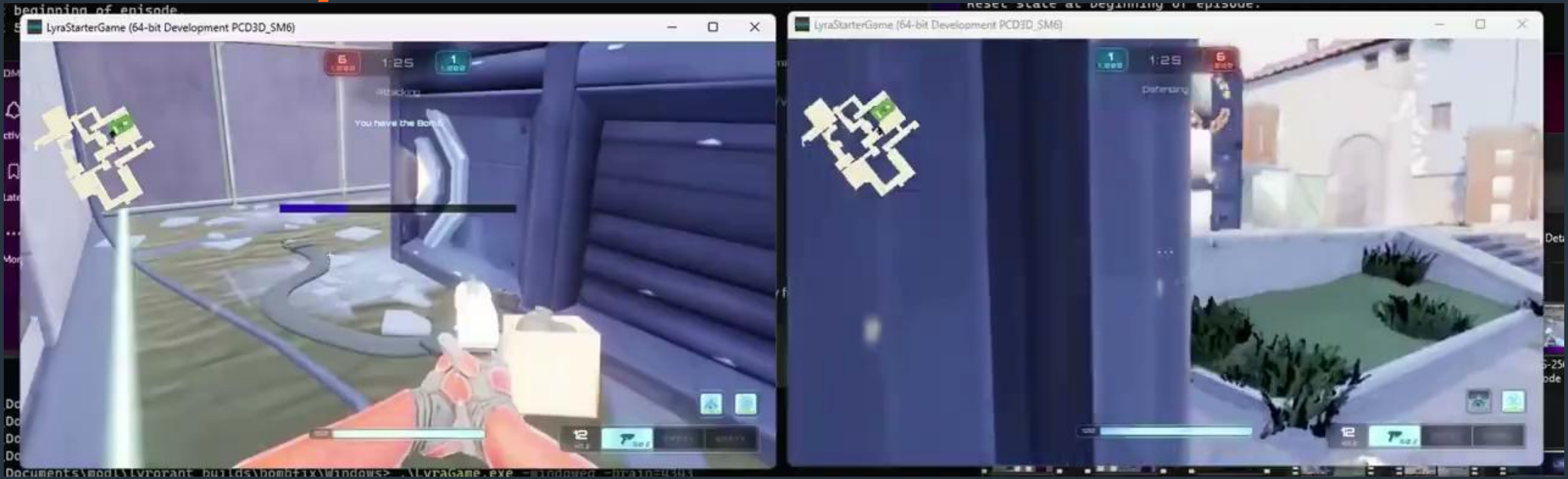


Figure 1
Aim
Enemies
Teammates
Walls
Bombs
Sky smoke
Dropped Bombs
Planted Bombs
Bomb Site
Team Grenades
Enemy Grenades
Incendiary
2
1:25
3
Attacking team wins
Round Over...
12
EMPTY
EMPTY

Some examples of agent behavior

Sneaky bomb defusal



Lava jumping

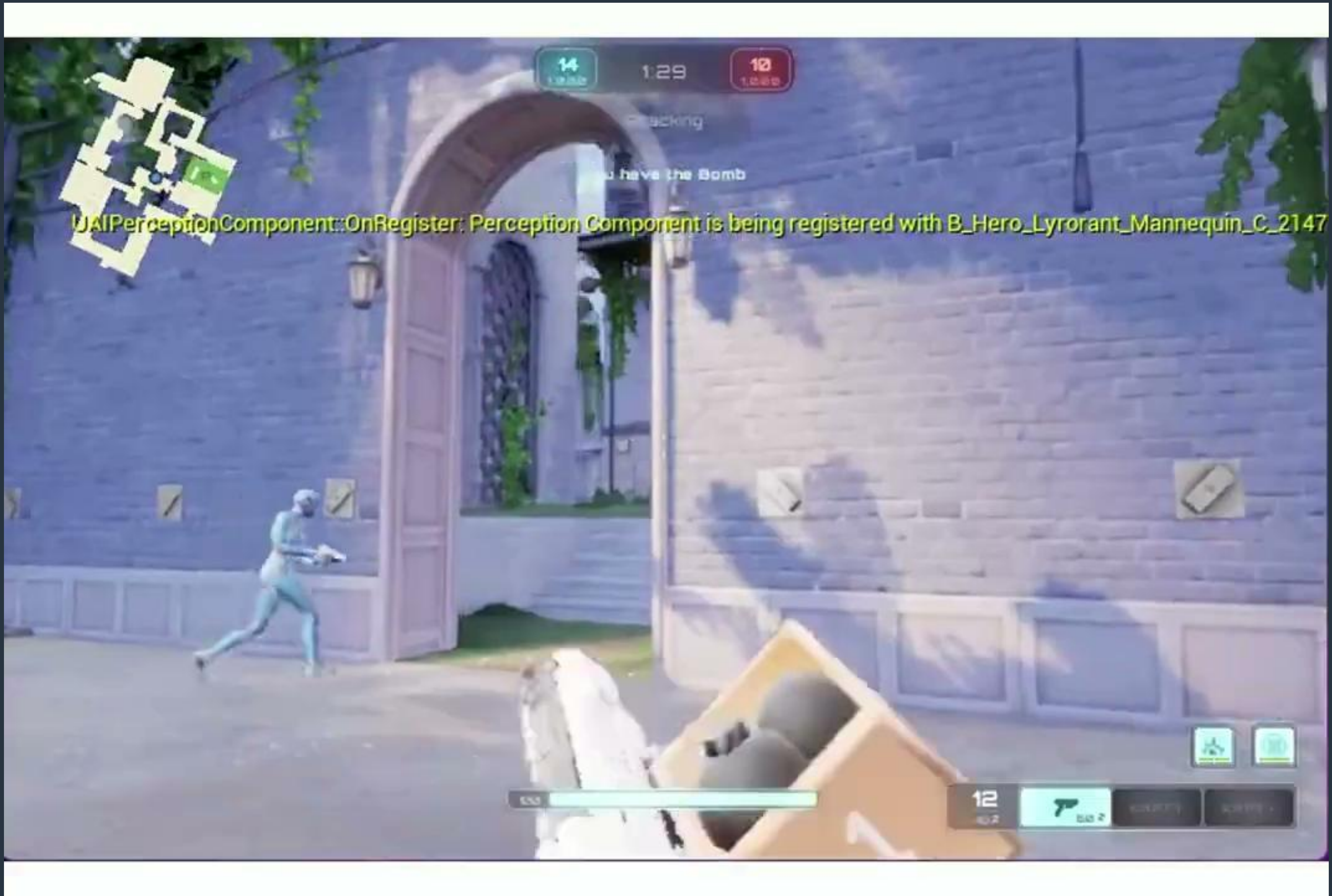
Waiting in smoke

Camping the smoke

Dodging an enemy

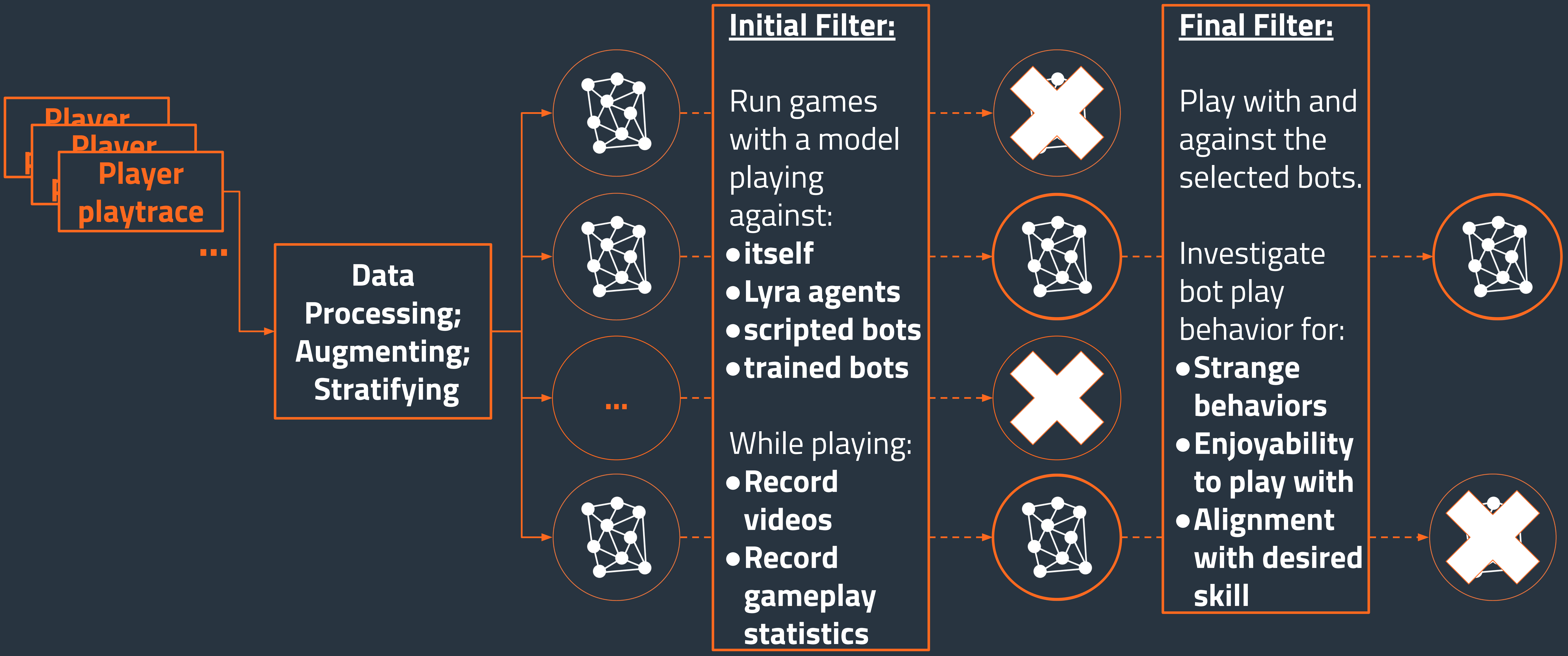
Planting the bomb and defending it

Chansing an enemy and avoiding damage



Surprising an enemy

Our Evaluation Framework and Pipeline



Figuring out which model to use

ATTACK

similarity

KL

Time Duration	E	D	F	B	C	A
Speed	B	C	F	E	A	D
Shot	C	A	B	F	D	E
Shots per Kill	C	B	A	E	D	F
Kill per Round	D	F	E	B	C	A
Plant Attempt per Round	E	B	D	C	A	F
Abilities per Round	F	B	C	E	D	A

JS

Time Duration	E	D	F	B	C	A
Speed	B	C	F	E	D	A
Shot	D	A	C	F	B	E
Shots per Kill	C	B	D	F	E	A
Kill per Round	D	F	E	B	C	A
Plant Attempt per Round	D	C	A	E	B	F
Abilities per Round	F	C	B	E	D	A

EMD

Time Duration	E	D	C	B	F	A
Speed	F	A	B	E	C	D
Shot	D	A	C	B	F	E
Shots per Kill	D	C	B	F	E	A
Kill per Round	D	C	B	F	E	A
Plant Attempt per Round	C	B	F	D	E	A
Abilities per Round	C	F	B	E	D	A

DEFENCE

similarity

KL

Time Duration	D	E	B	F	C	A
Speed	B	F	D	C	E	A
Shot	C	A	B	D	E	F
Shots per kill	C	E	D	A	B	F
Kill per Round	B	D	E	C	F	A
Defuse Attempt per Round	E	F	C	D	A	B
Abilites per Round	F	E	C	D	B	A

JS

Time Duration	D	E	F	B	C	A
Speed	B	F	C	E	D	A
Shot	C	D	E	B	A	F
Shots per kill	E	D	C	F	B	A
Kill per Round	D	B	E	C	F	A
Defuse Attempt per Round	E	F	D	C	A	B
Abilites per Round	F	C	E	D	B	A

EMD

Time Duration	E	D	B	C	F	A
Speed	B	E	D	F	C	A
Shot	A	C	B	D	E	F
Shots per kill	D	B	E	C	F	A
Kill per Round	D	B	E	C	F	A
Defuse Attempt per Round	D	E	F	A	B	C
Abilites per Round	F	E	C	D	A	B

Models	Similarity metric
A	1.67
B	3.87
C	4.10
D	**4.33**
E	3.67
F	3.43

Network sise

A Small size
B
C
D
E
F Large size

Thanks to everyone involved:
Sam Snodgress, Niels Justesen, Miruna Vozaru, Matthew Schlegel, Gabriella Barros, Tobias Mahlmann, Maria Kaselimi, Shyam Sudhukaran, Wesley Kerr, Albert Wang, Georgios Yannakakis, Sebastian Risi, Julian Togelius