DeepMind

AlphaStar

Wojciech Czarnecki
on behalf of AlphaStar Team

Invited talk as part of
Computational Game Solving

Carnegie Mellon University
School of Computer Science


1
Collect resources
2
Build a base
3
Build units
4
Defeat the opponent

1

Complex Combinatorial Action Space

- → automatic theorem proving
- → drug design
- → industrial control problems
- → AGI
- → ...

2

Multi-modal Observation Space

- → robotics
- → self-driving cars
- → AGI
- → …

STARCRAFT II

3

Information "Poverty" and Hard Exploration

- → natural sciences
- → weather forecasting
- → robotics
- → AGI
- → …

4

Human "alignment"

- → self driving cars
- → home assistants
- → human enhancing AIs
- → AGI
- → …

5

Multiple Interacting Agents

- → self driving cars
- → any "real life" deployment
- → AGI
- → ...

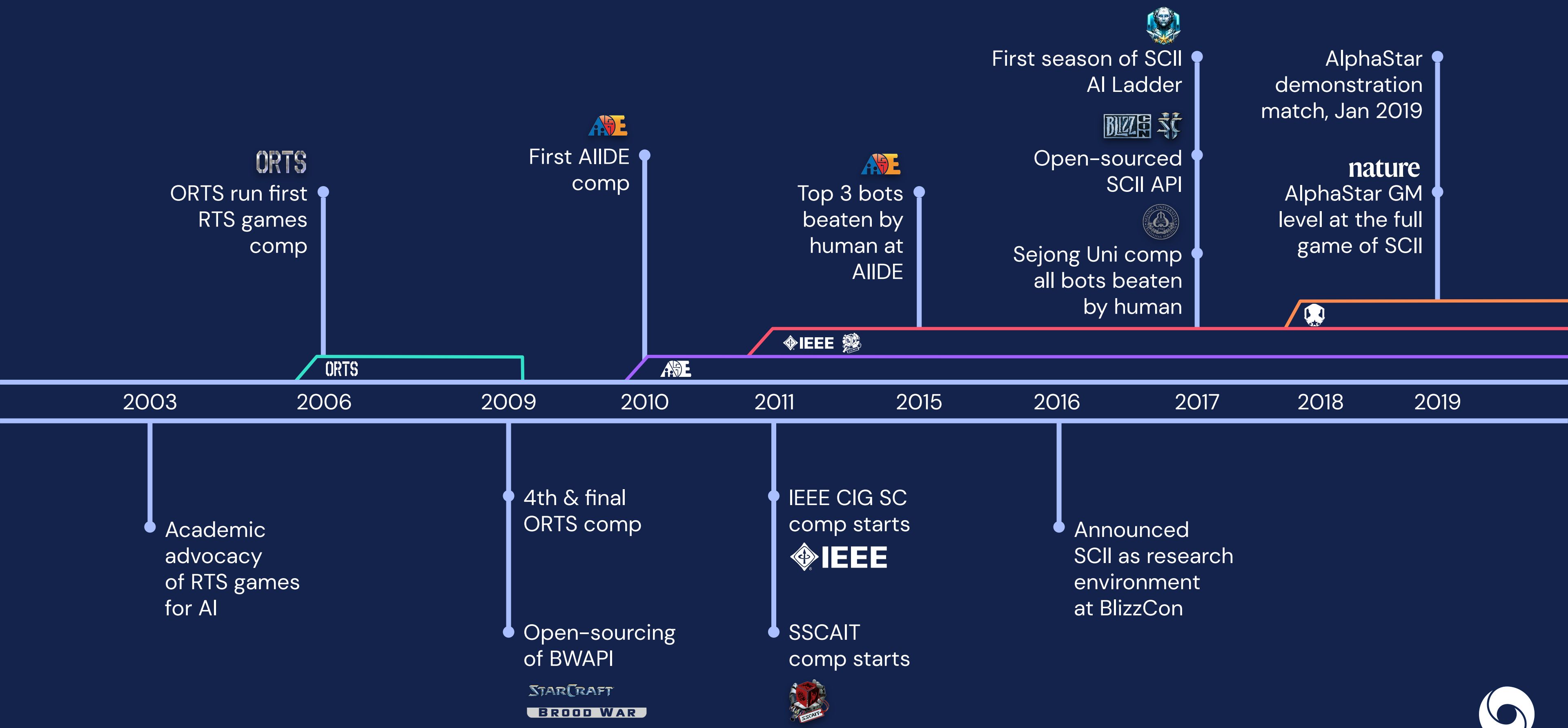
YOU'VE BEEN PROMOTED!

1V1 GRANDMASTER

Your performance has qualified you for placement in a new league

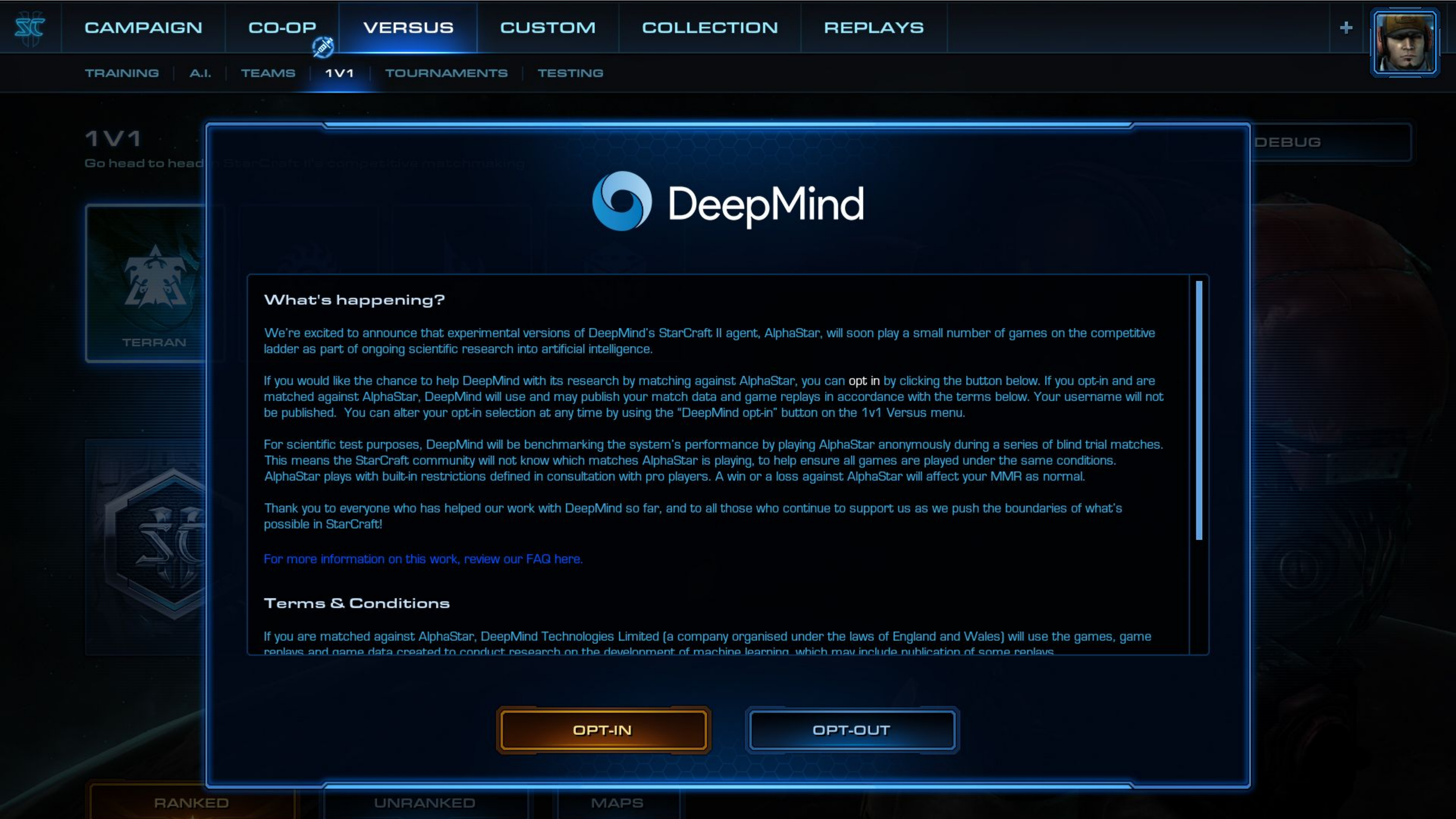
OK

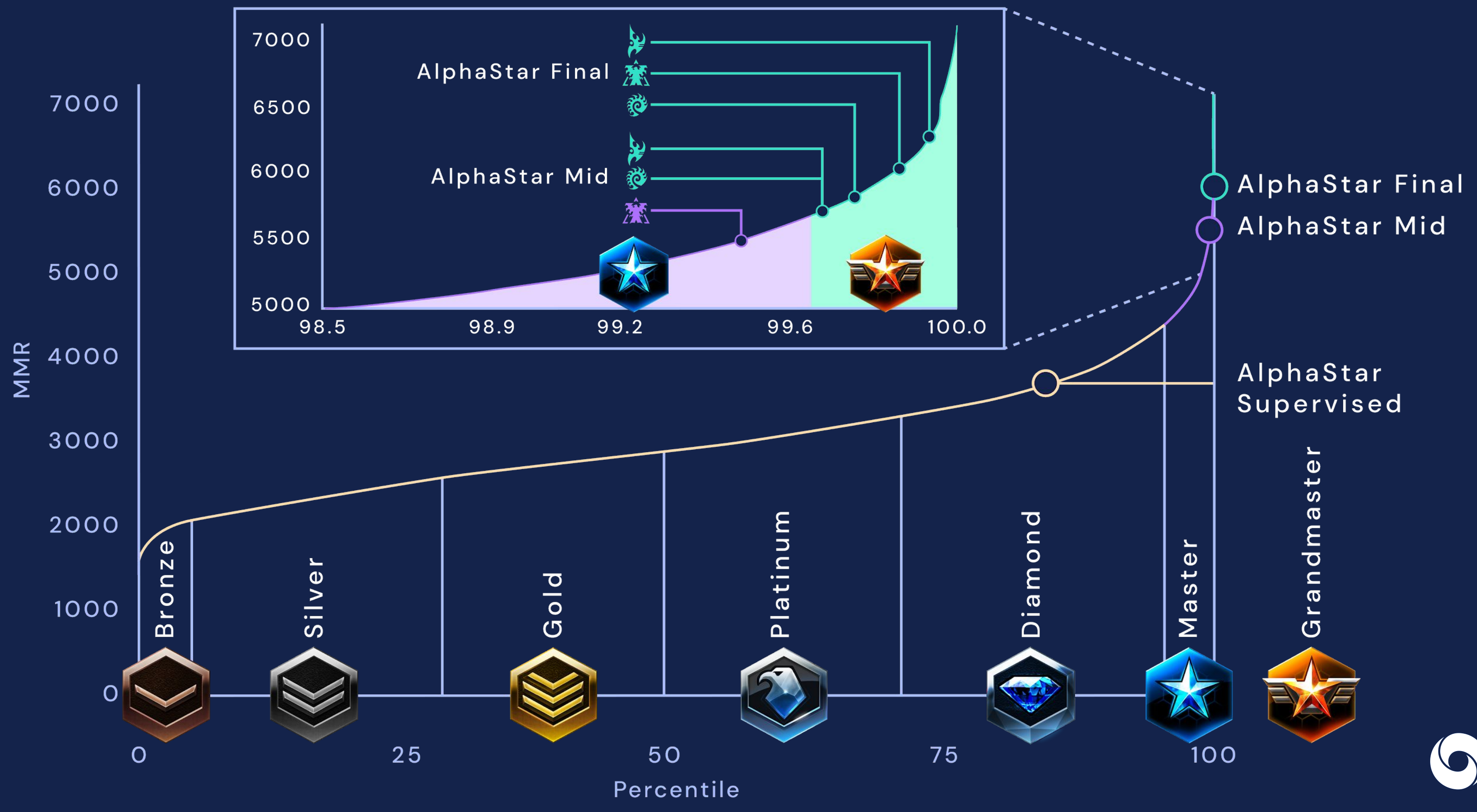
The international journal of science / 14 November 2019
nature
GAME PLAN
AI program learns to play *StarCraft II* to Grandmaster level
Pharmaceuticals
How to fit a drug factory inside a briefcase
3D printing
Nozzle extrudes multimaterial objects in a single run
Cancer imaging
Tracer reveals metabolic nature of live lung tumours



2003
Academic advocacy of RTS games for AI
2006
ORTS
ORTS run first RTS games comp
2009
4th & final ORTS comp
Open-sourcing of BWAPI
StarCraft Brood War
2010
AIIDE
First AIIDE comp
2011
IEEE
IEEE CIG SC comp starts
SSCAIT comp starts
SSCAIT
2015
Top 3 bots beaten by human at AIIDE
2016
Announced SCII as research environment at BlizzCon
2017
First season of SCII AI Ladder
BlizzCon
Open-sourced SCII API
Sejong Uni comp all bots beaten by human
2018
2019
AlphaStar demonstration match, Jan 2019
nature
AlphaStar GM level at the full game of SCII


GOAL
Observations
Agent
Environment
Actions

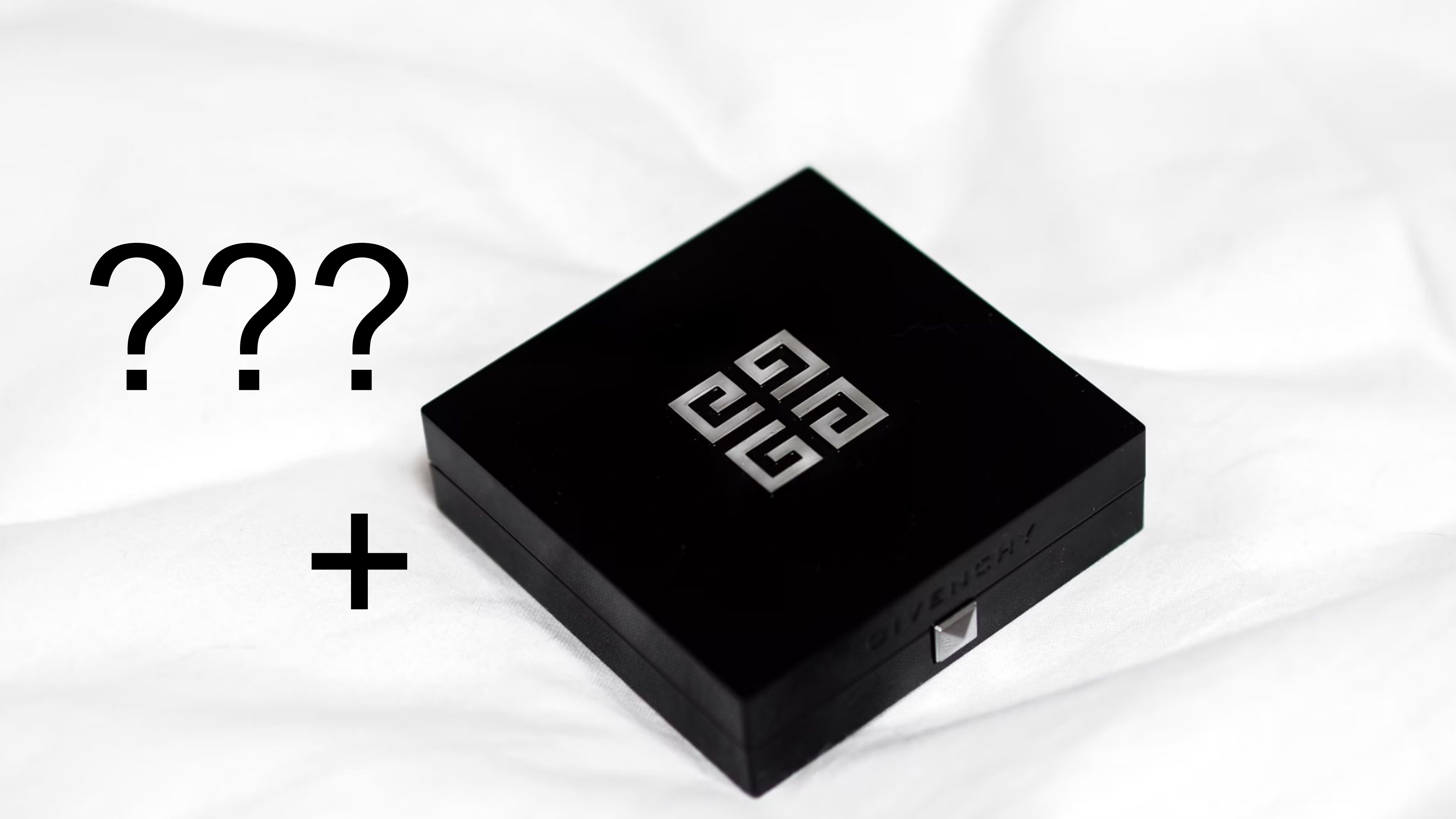

CAMPAIGN
CO-OP
VERSUS
CUSTOM
COLLECTION
REPLAYS
TRAINING
A.I.
TEAMS
1V1
TOURNAMENTS
TESTING
1V1
Go head to head
DEBUG
TERRAN
DeepMind
What's happening?
We're excited to announce that experimental versions of DeepMind's StarCraft II agent, AlphaStar, will soon play a small number of games on the competitive ladder as part of ongoing scientific research into artificial intelligence.
If you would like the chance to help DeepMind with its research by matching against AlphaStar, you can opt in by clicking the button below. If you opt-in and are matched against AlphaStar, DeepMind will use and may publish your match data and game replays in accordance with the terms below. Your username will not be published. You can alter your opt-in selection at any time by using the "DeepMind opt-in" button on the 1v1 Versus menu.
For scientific test purposes, DeepMind will be benchmarking the system's performance by playing AlphaStar anonymously during a series of blind trial matches. This means the StarCraft community will not know which matches AlphaStar is playing, to help ensure all games are played under the same conditions. AlphaStar plays with built-in restrictions defined in consultation with pro players. A win or a loss against AlphaStar will affect your MMR as normal.
Thank you to everyone who has helped our work with DeepMind so far, and to all those who continue to support us as we push the boundaries of what's possible in StarCraft!
For more information on this work, review our FAQ here.
Terms & Conditions
If you are matched against AlphaStar, DeepMind Technologies Limited (a company organised under the laws of England and Wales) will use the games, game replays and game data created to conduct research on the development of machine learning, which may include publication of some replays
OPT-IN
OPT-OUT
RANKED
UNRANKED
MAPS



7000
6000
5000
4000
3000
2000
1000
0
MMR
0
25
50
75
100
Percentile
Bronze
Silver
Gold
Platinum
Diamond
Master
Grandmaster
AlphaStar Final
AlphaStar Mid
AlphaStar Supervised
7000
6500
6000
5500
5000
98.5
98.9
99.2
99.6
100.0
AlphaStar Final
AlphaStar Mid


Serral
AlphaStar Final
DeepMind
ArtosisTV
Zergling

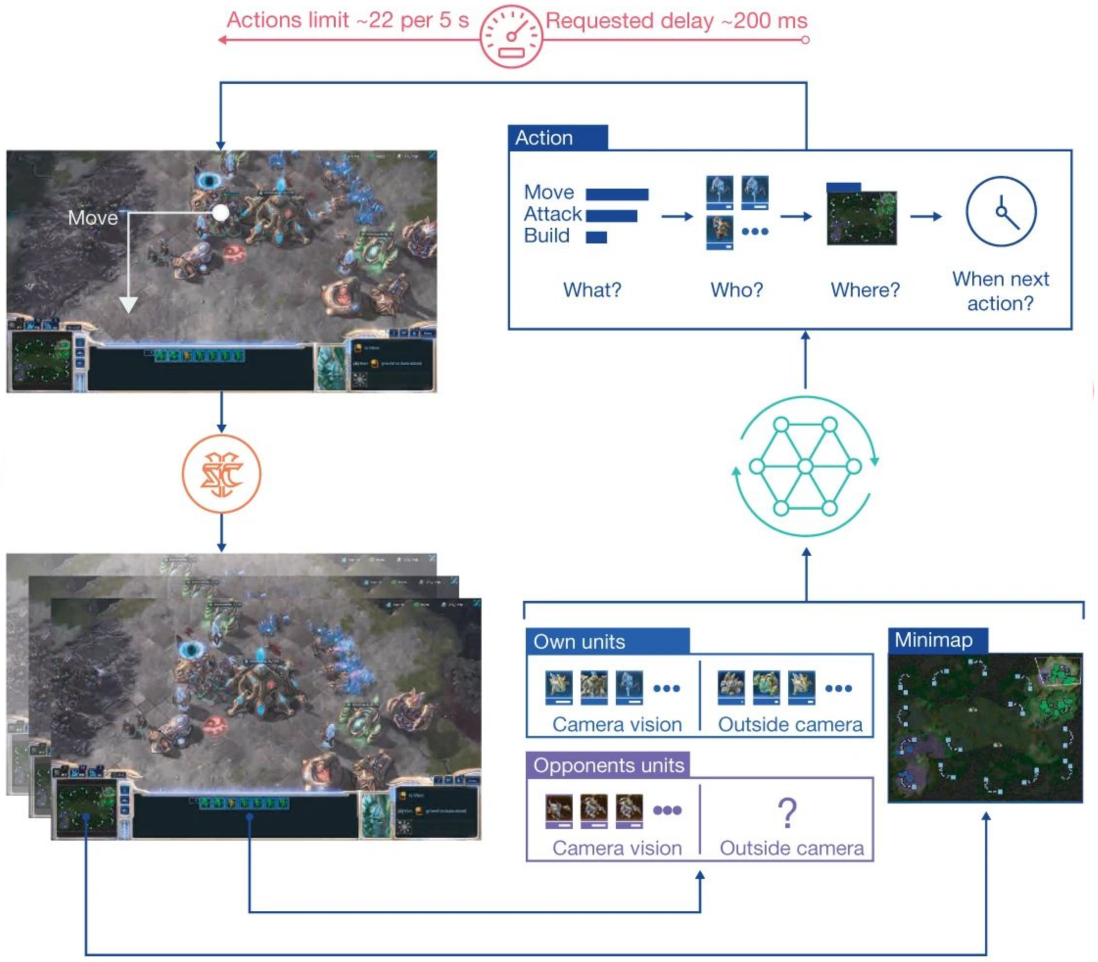

STARCRAFT
WORLD CHAMPIONSHIP SERIES


???
+
GIVENCHY

DeepMind

Interface

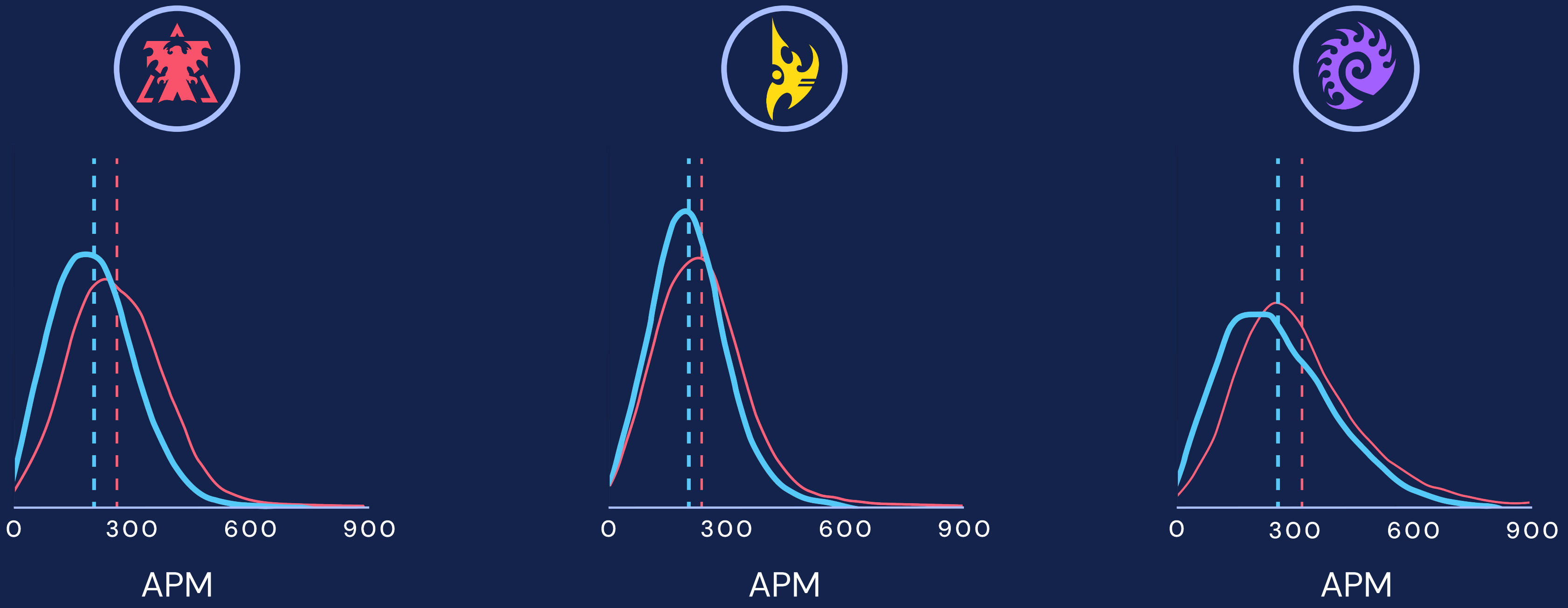
Human alignment



Actions limit ~22 per 5 s
Requested delay ~200 ms
Move
Action
Move
Attack
Build
What?
Who?
Where?
When next action?
Own units
Camera vision
Outside camera
Opponents units
?
Camera vision
Outside camera
Minimap

Reduced APM - Pro tested and approved

AlphaStar   Battle.net opponents

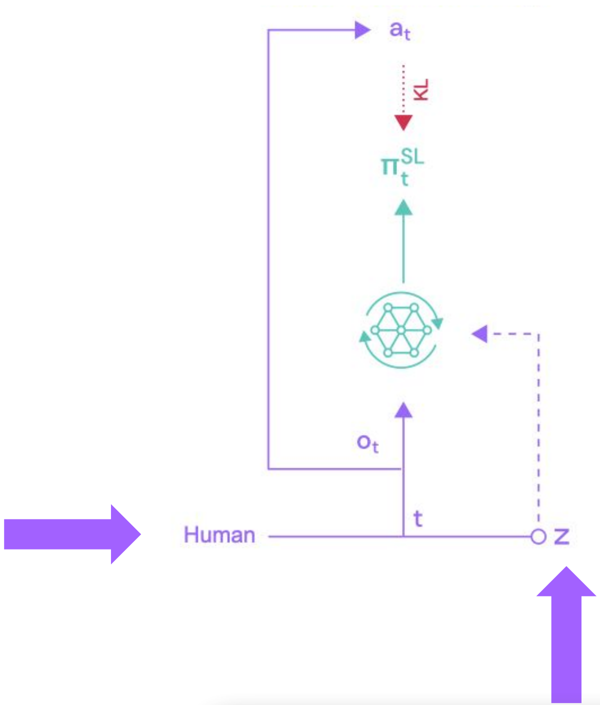


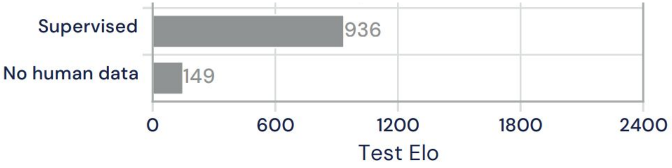
Zerg has higher APMs due to repeat actions, such as morphing & spawning

DeepMind

Supervised learning

Hard exploration
Information poverty


a_t
KL
π_t^{SL}
o_t
t
Human
z


Supervised
936
No human data
149
0
600
1200
1800
2400
Test Elo


13/14
17/22
18/22
17/22
19/22
19/22
21/30

Even AlphaStar Supervised is not a single "strategy". It is a (controllable!) collection of dozens of thousands of strategies

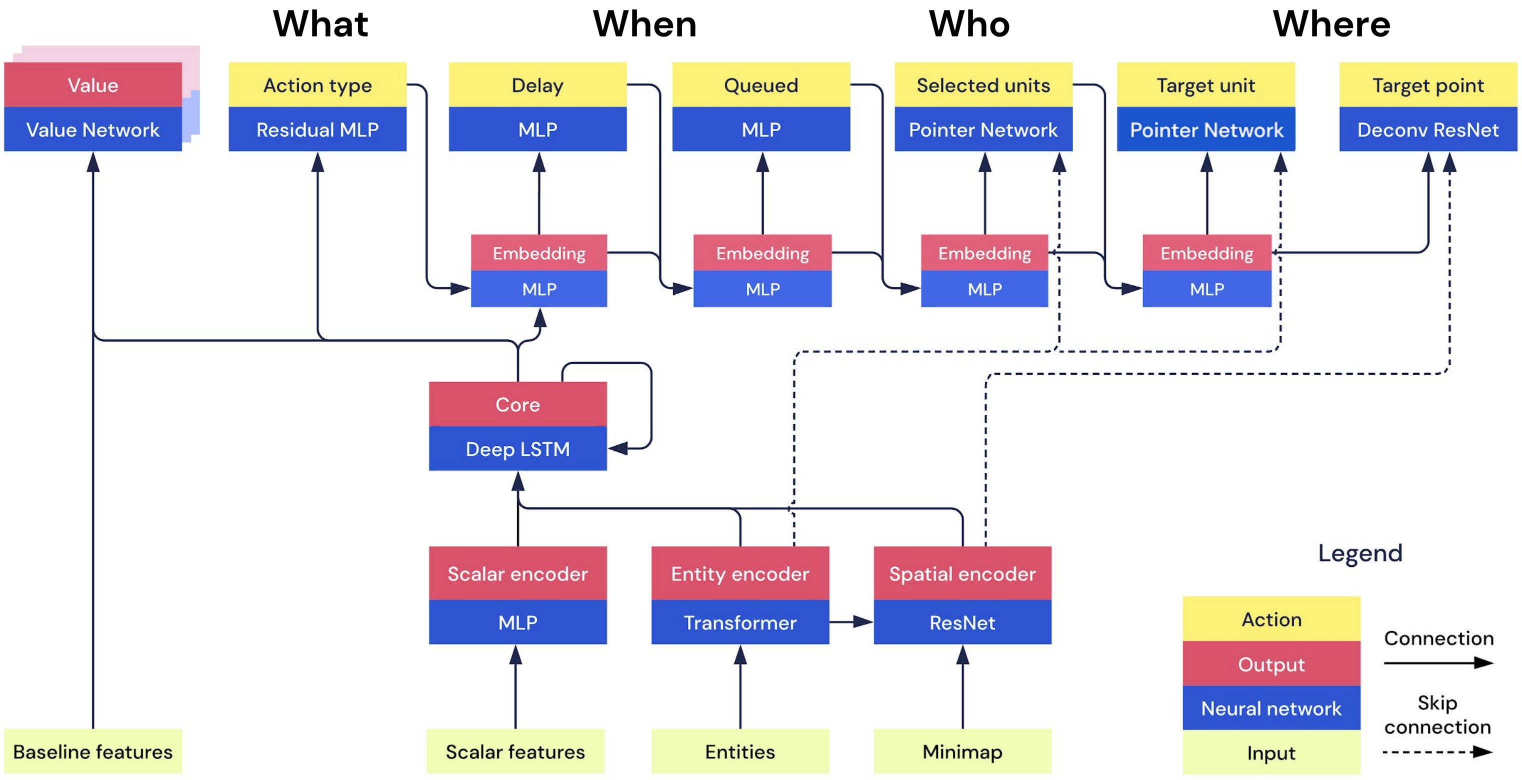

ALPHASTAR'S
MASS
MUTALISK!
DiamondPlayer 613 331 161/192 57 104
AlphaStar 4420 1547 187/198 69 118
Spire


AlphaStar
+ Main exploiters
PFSP + SP
PFSP
Master league
FSP
Diamond league
AlphaStar supervised
70
80
90
100

DeepMind

Architecture

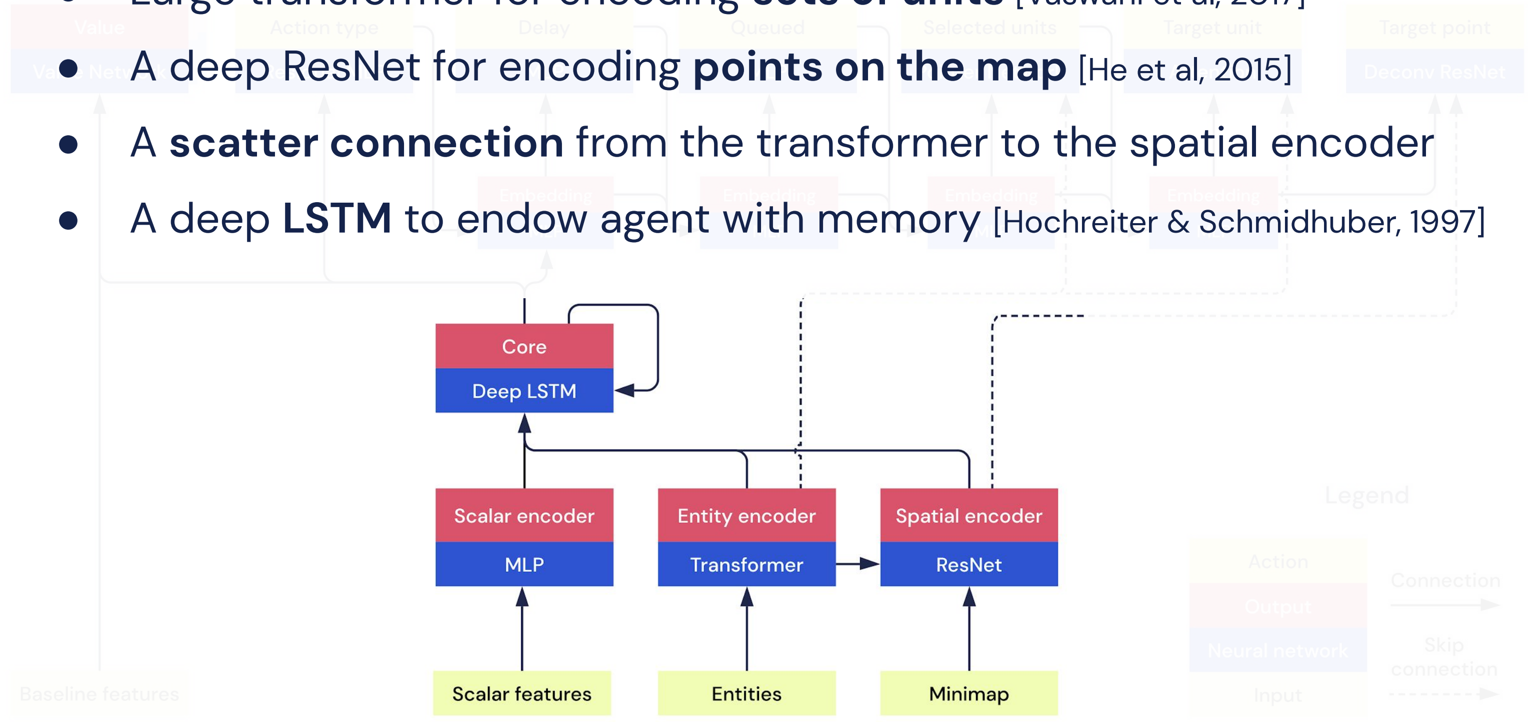
Combinatorial Action Space
Multi-modal Observations



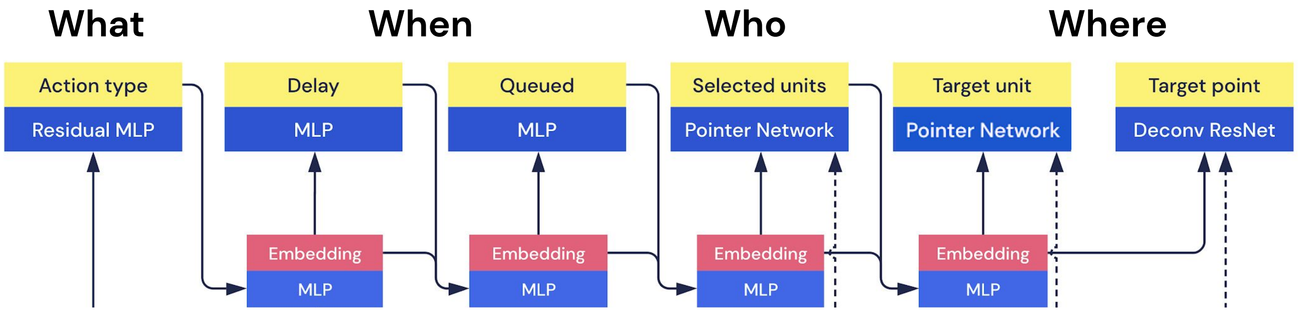
What
When
Who
Where
Value
Value Network
Action type
Residual MLP
Delay
MLP
Queued
MLP
Selected units
Pointer Network
Target unit
Pointer Network
Target point
Deconv ResNet
Embedding
MLP
Embedding
MLP
Embedding
MLP
Embedding
MLP
Core
Deep LSTM
Scalar encoder
MLP
Entity encoder
Transformer
Spatial encoder
ResNet
Baseline features
Scalar features
Entities
Minimap
Legend
Action
Output
Neural network
Input
Connection
Skip connection

Observation encoders and LSTM

- Large transformer for encoding **sets of units** [Vaswani et al, 2017]
- A deep ResNet for encoding **points on the map** [He et al, 2015]
- A **scatter connection** from the transformer to the spatial encoder
- A deep **LSTM** to endow agent with memory [Hochreiter & Schmidhuber, 1997]



Autoregressive action head



- Fully autoregressive action head with 7 sub-heads: $p(x) = \prod_{i=1}^{n} p(x_i | x_1, \ldots\ldots, x_{i-1})$
- Four scalar heads: **action type**, **action delay**, **action repeat, modifier key**
- A recurrent pointer network to select a **set of units** [Vinyals, Fortunato & Jaitly, 2015]
- A simple pointer network to select **single units**
- A ResNet decoder to select **points on the map**

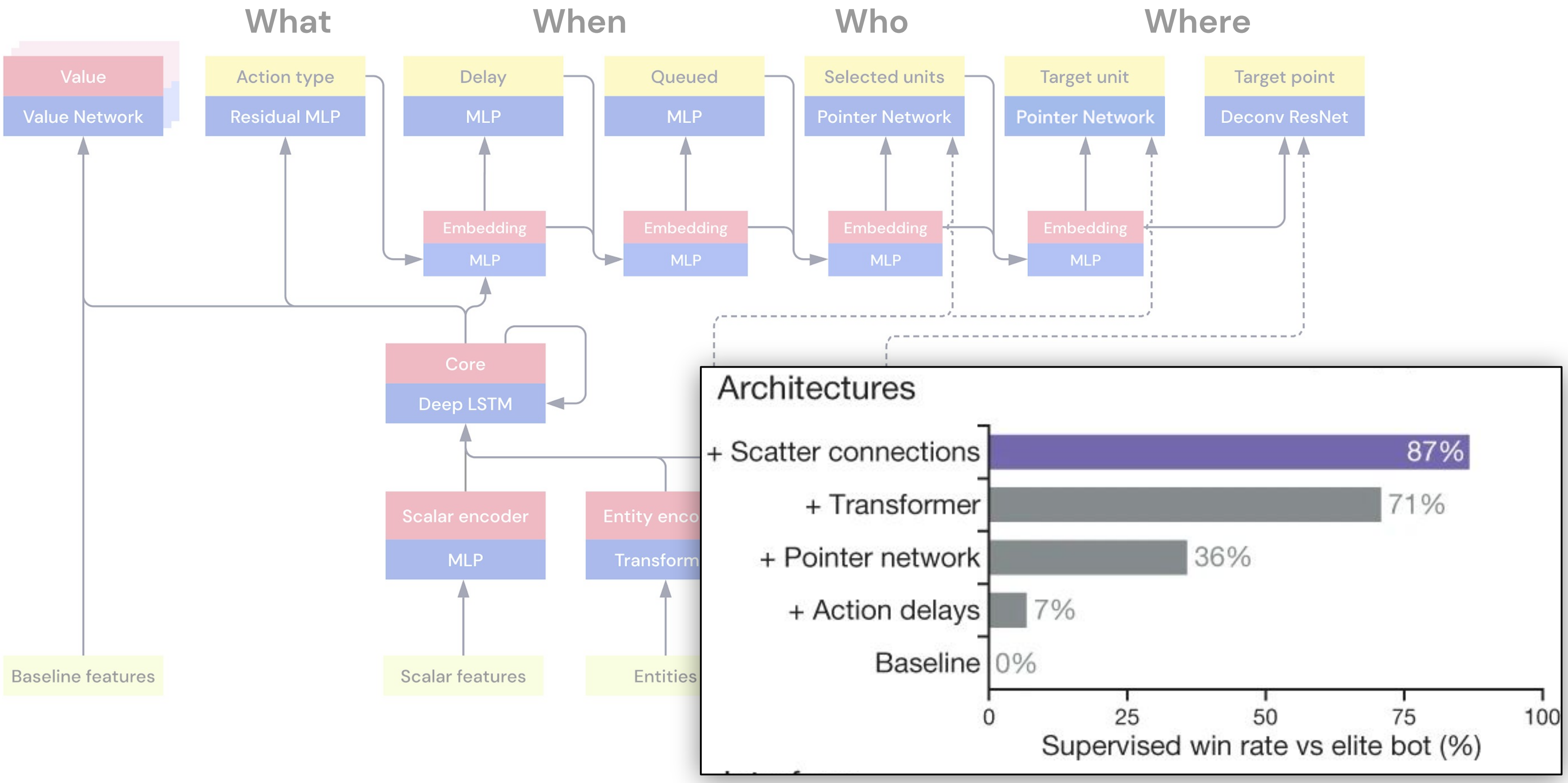
Story time:
Why is our Zerg agent so bad compared to other races?

- Other races are just better initially, they dominate the Zerg and it cannot flourish
- We are missing some of the Zerg-specific observations/parts of environment

Story time:
Why is our Zerg agent so bad compared to other races?

- Other races are just better initially, they dominate the Zerg and it cannot flourish
- We are missing some of the Zerg-specific observations/parts of environment
- It's because our Protoss agent is not good enough, and there is an architectural trick missing!

Story time:
Why is our Zerg agent so bad compared to other races?

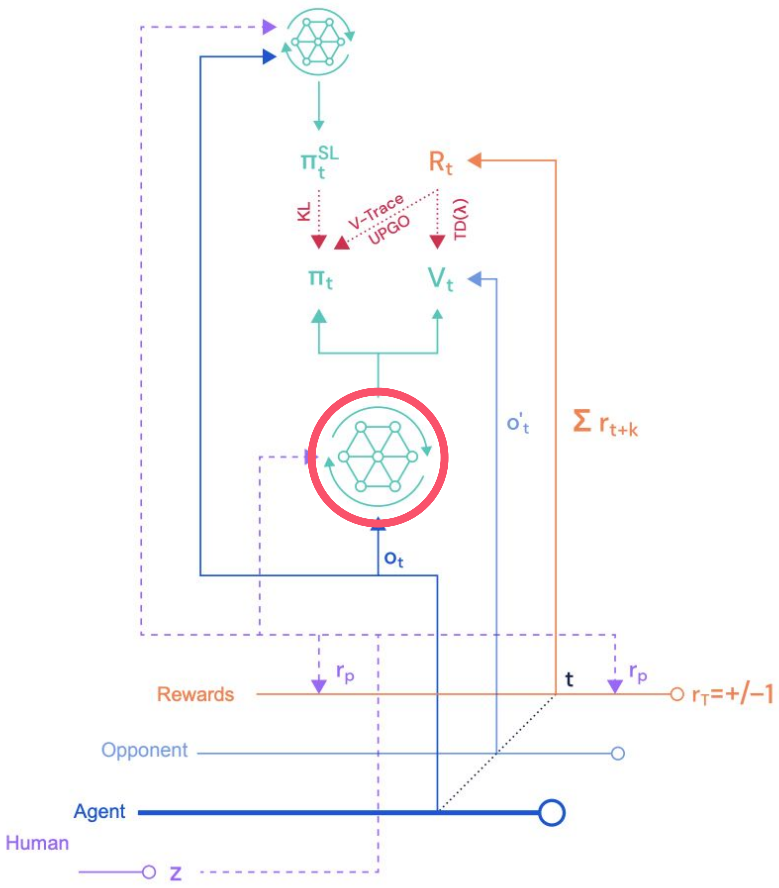

What
When
Who
Where
Value
Value Network
Action type
Residual MLP
Delay
MLP
Queued
MLP
Selected units
Pointer Network
Target unit
Pointer Network
Target point
Deconv ResNet
Embedding
MLP
Embedding
MLP
Embedding
MLP
Embedding
MLP
Core
Deep LSTM
Scalar encoder
MLP
Entity enco
Transform
Baseline features
Scalar features
Entities
Architectures
+ Scatter connections
87%
+ Transformer
71%
+ Pointer network
36%
+ Action delays
7%
Baseline
0%
0
25
50
75
100
Supervised win rate vs elite bot (%)

DeepMind

Reinforcement Learning

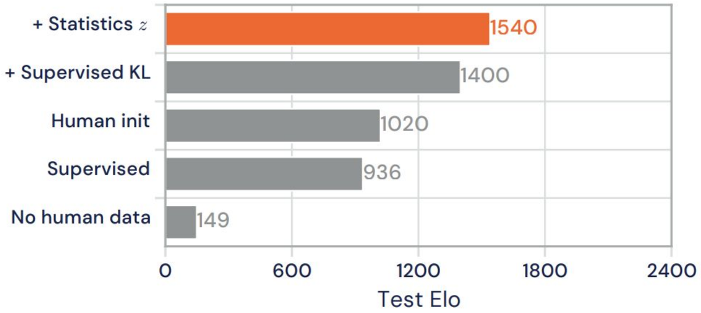
Hard exploration
Information poverty

Reinforcement Learning

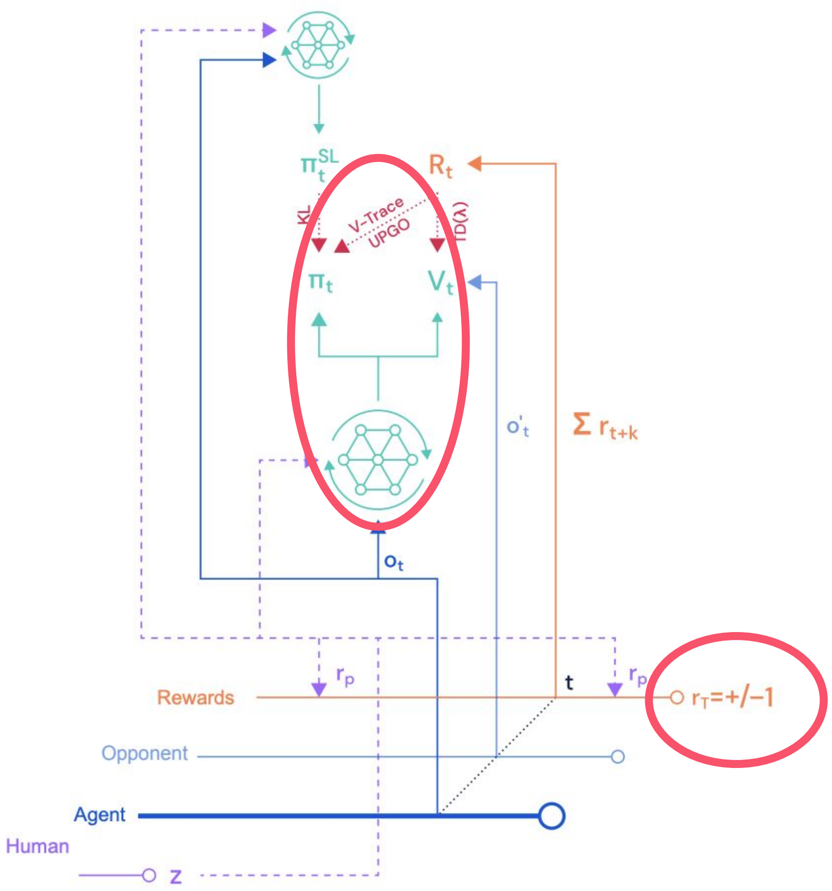


$$\pi^{SL}$$

E Human data usage

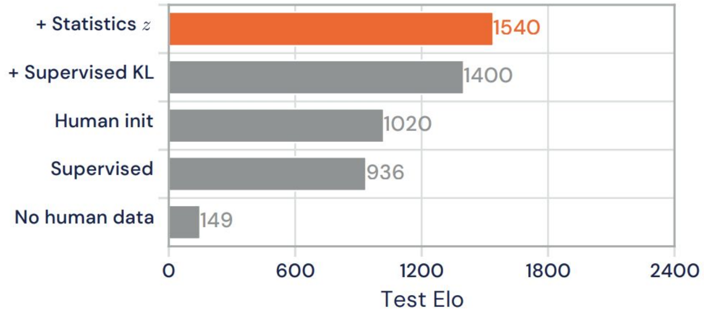


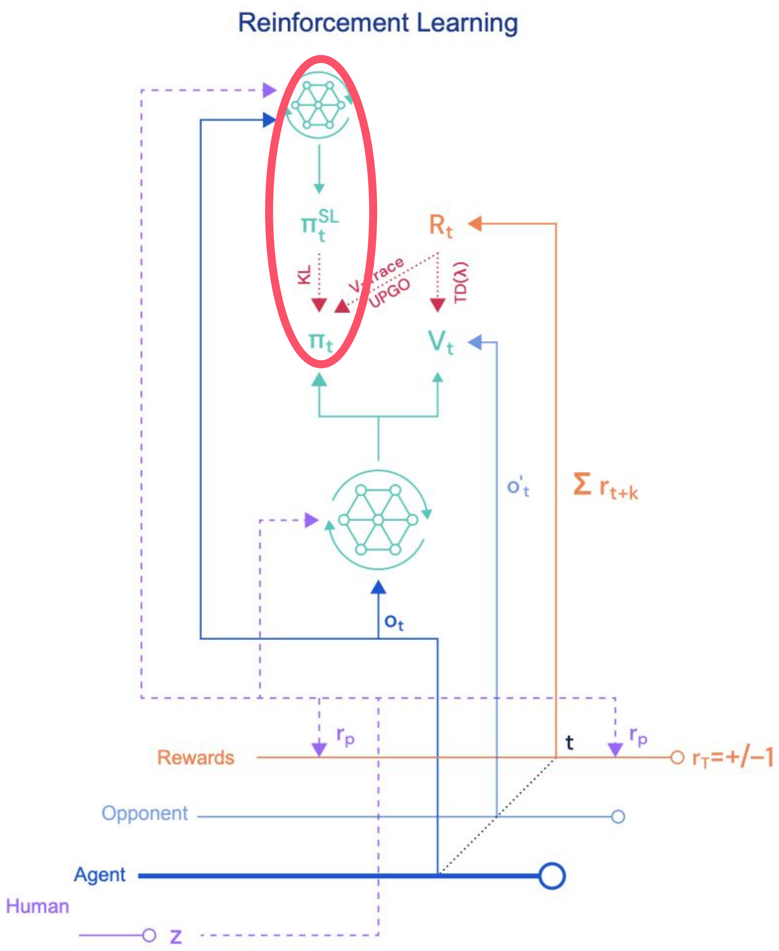
Reinforcement Learning



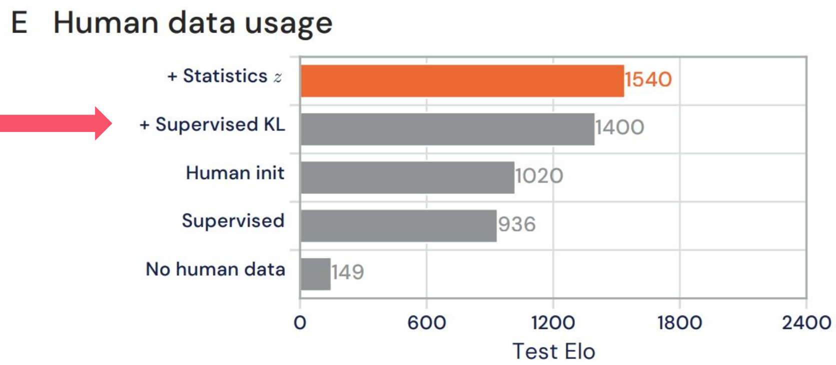
$$\pi_0 = \pi^{SL}$$

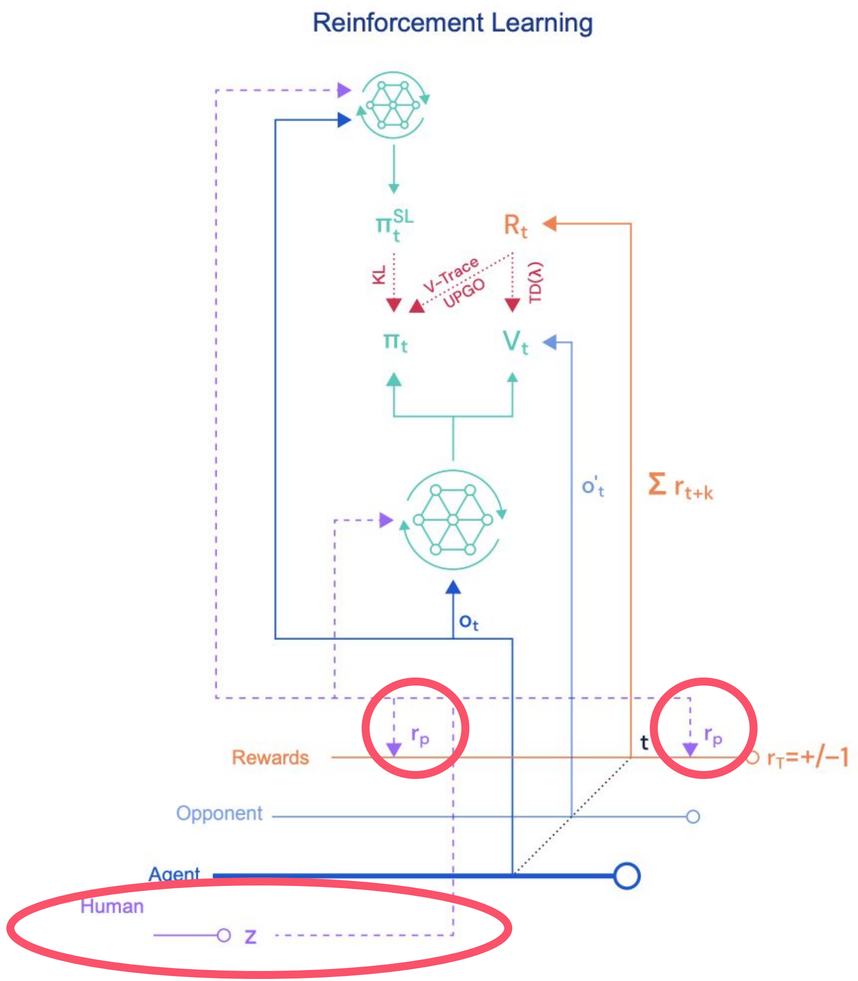
E Human data usage



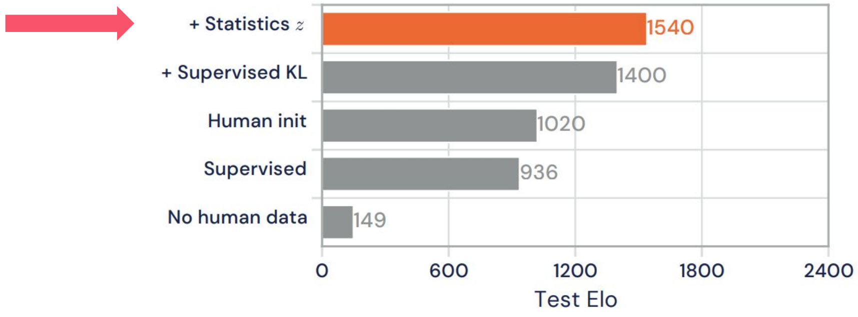


$$\pi_0 = \pi^{SL}$$
$$KL(\pi, \pi^{SL})$$





E Human data usage



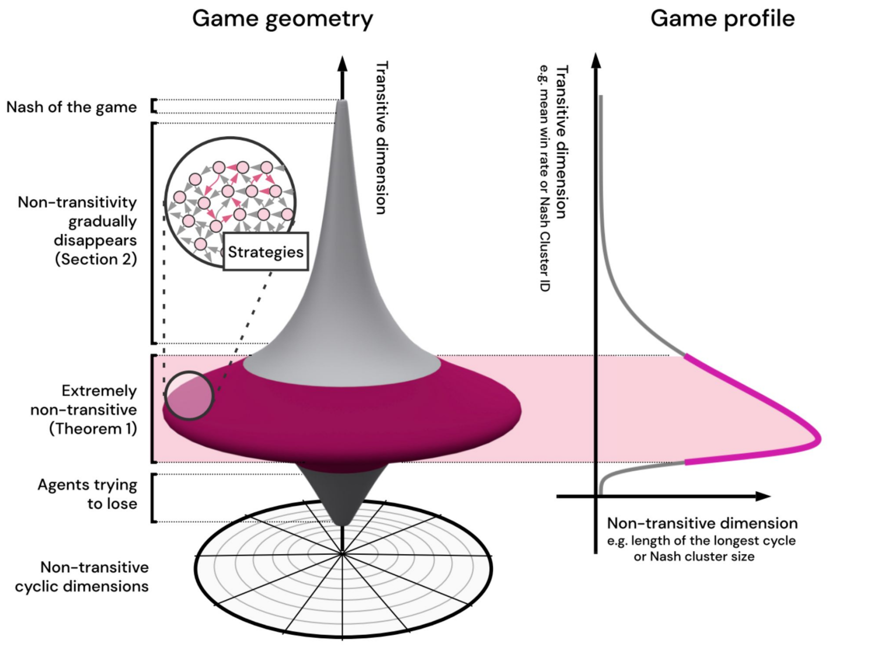
Do not start thinking about multi-agent dynamics research until you have a fully working, robust “best response” setup.

DeepMind

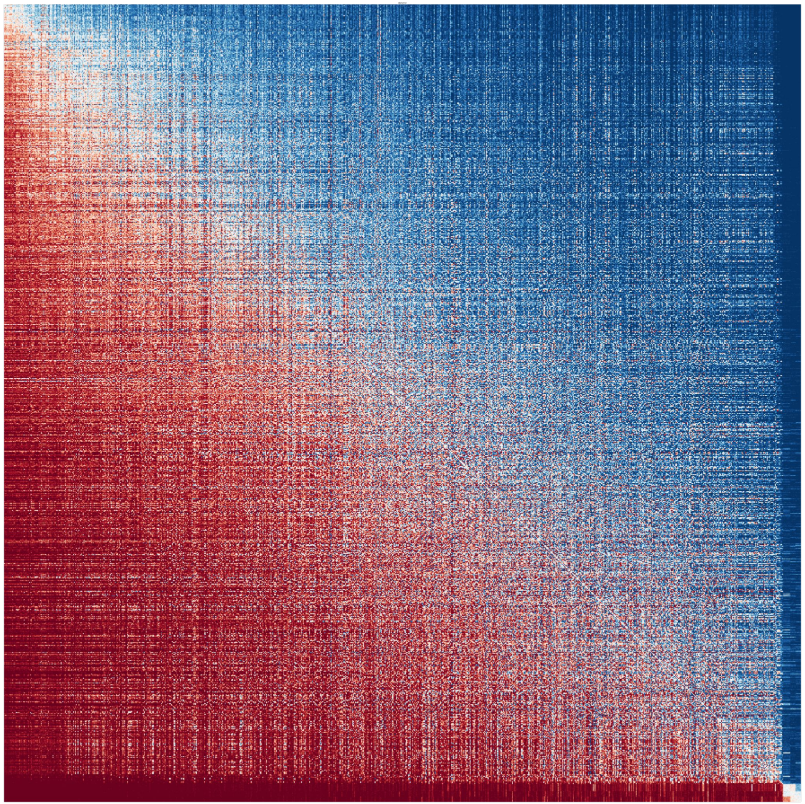
Multi-agent Learning

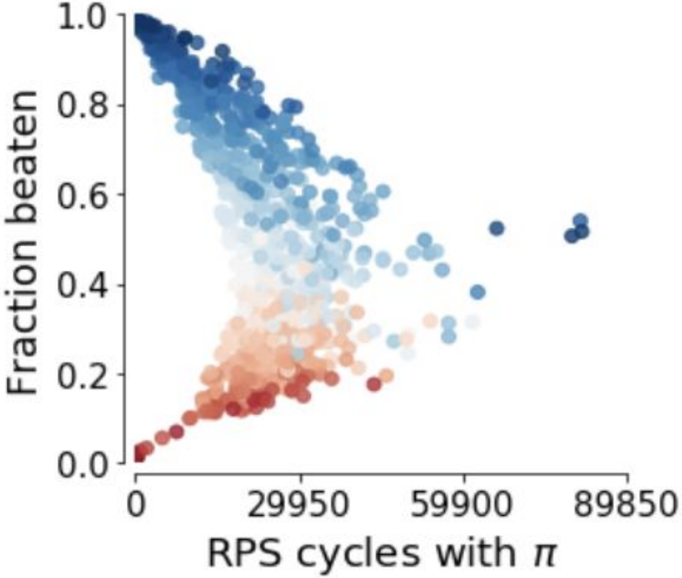
Multiple Interacting Agents
Hard exploration
Information poverty

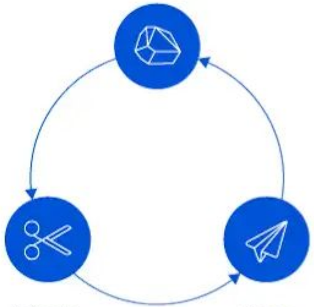
Real world games look like spinning tops



Payoff matrix analysis



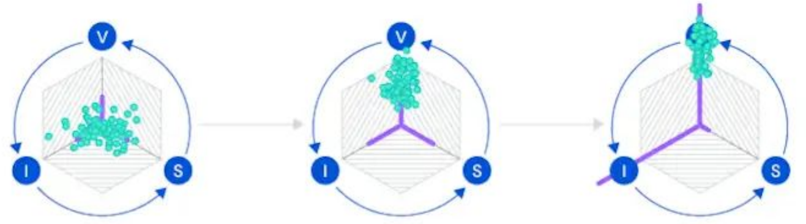




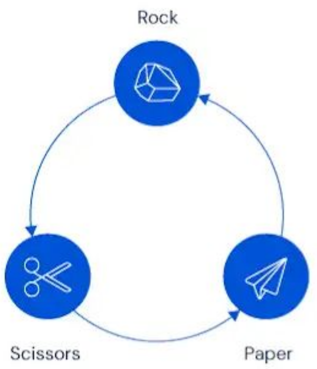
Rock, paper, scissors

StarCraft II players can create a variety of 'units', which have balanced strengths and weaknesses, similar to the game rock, paper, scissors

Self-play

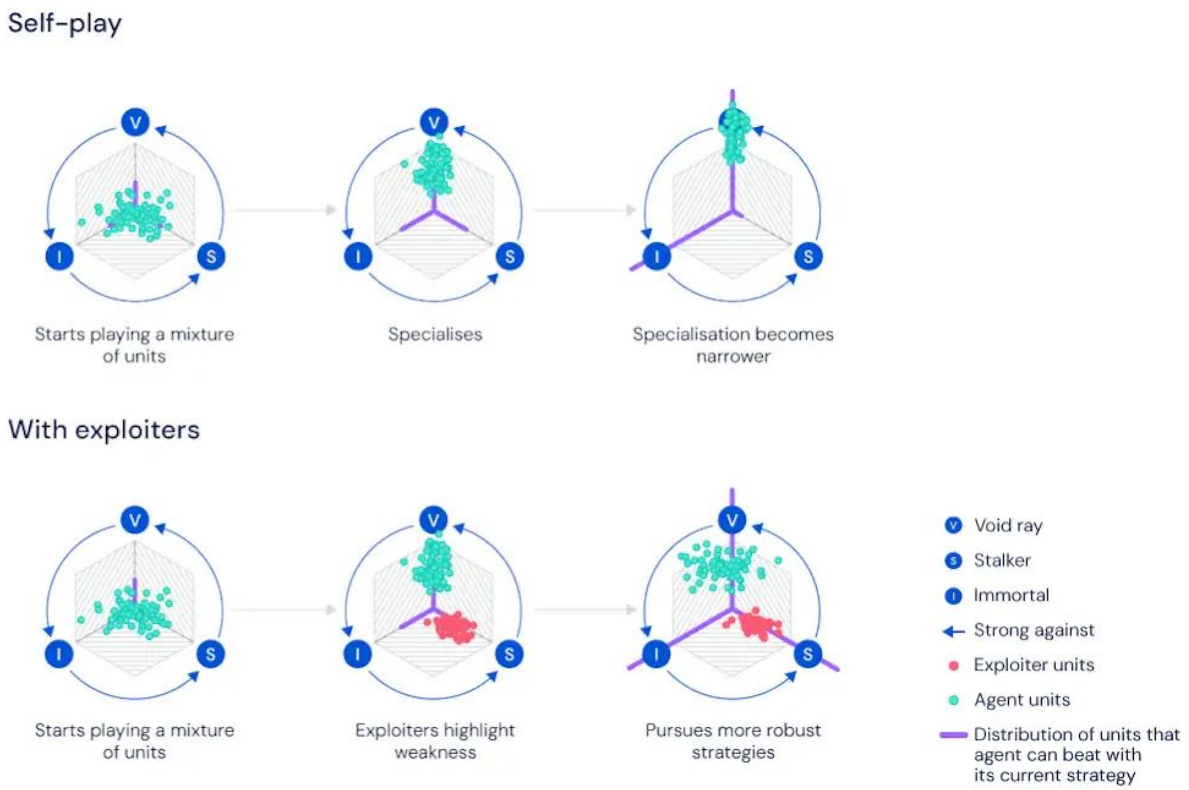


Void ray
Stalker
Immortal
Strong against
Exploiter units
Agent units
Distribution of units that agent can beat with its current strategy



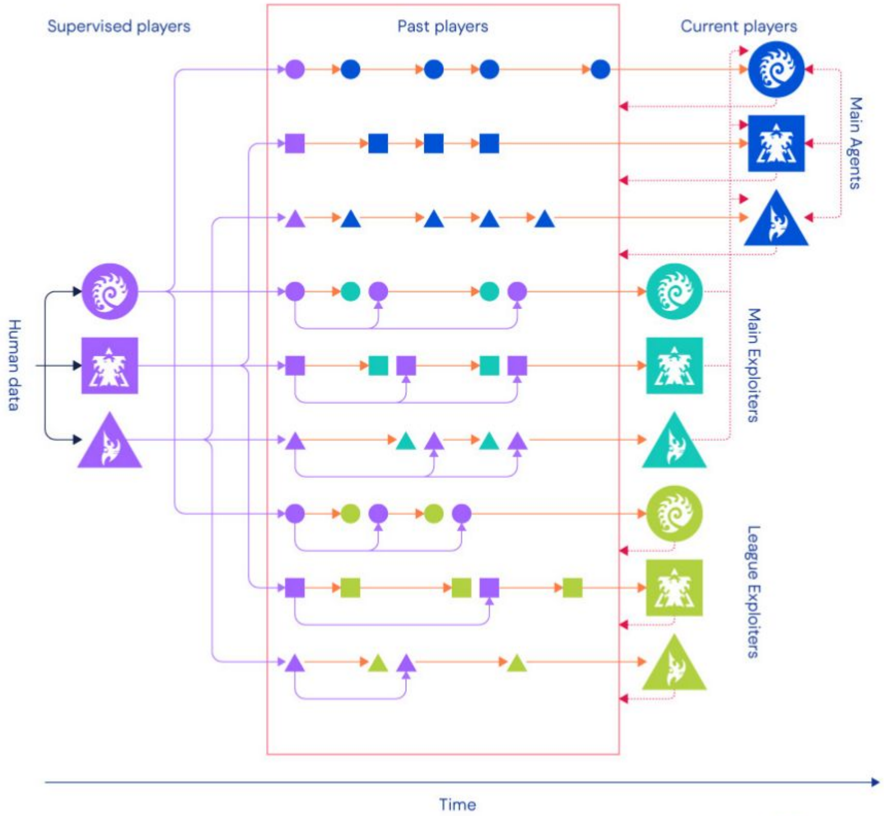
Rock, paper, scissors

StarCraft II players can create a variety of 'units', which have balanced strengths and weaknesses, similar to the game rock, paper, scissors



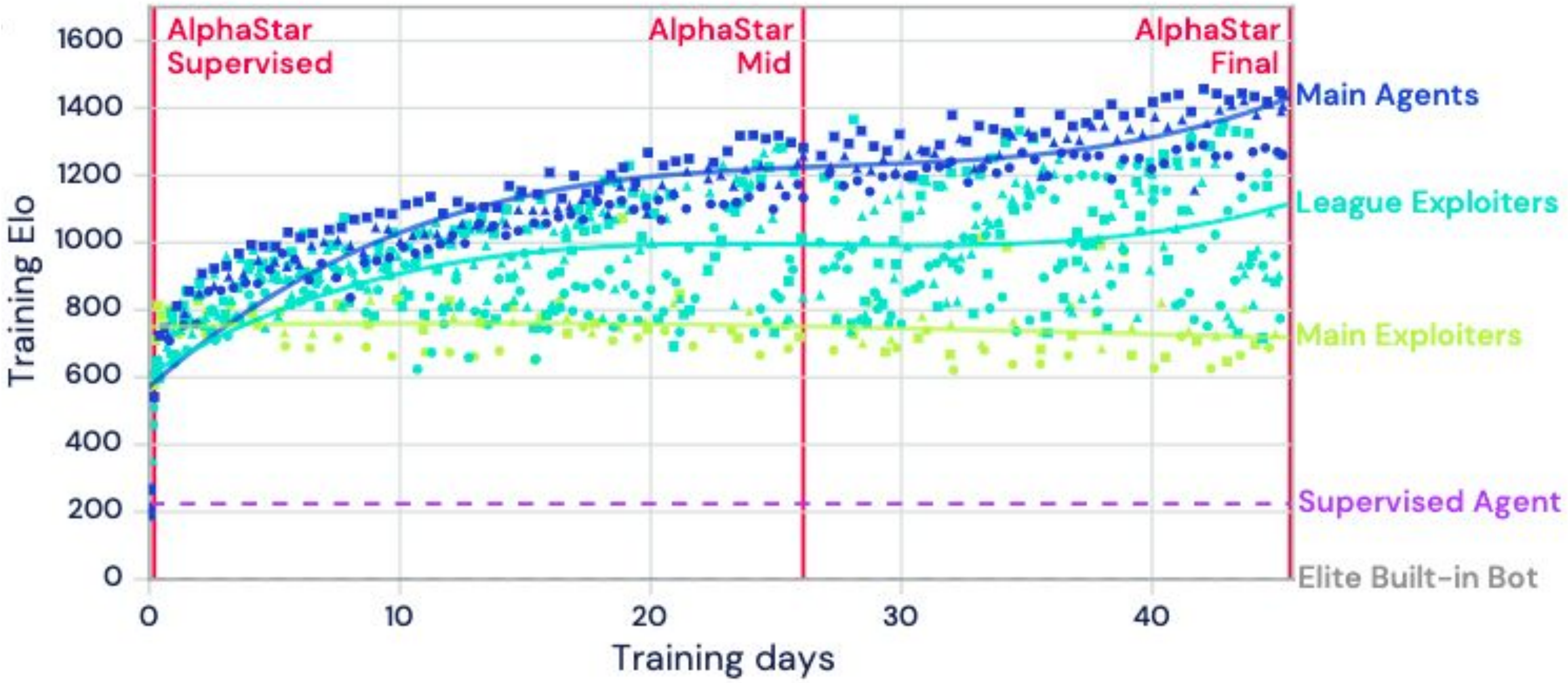

The main agent fails to defend itself.

The League Training

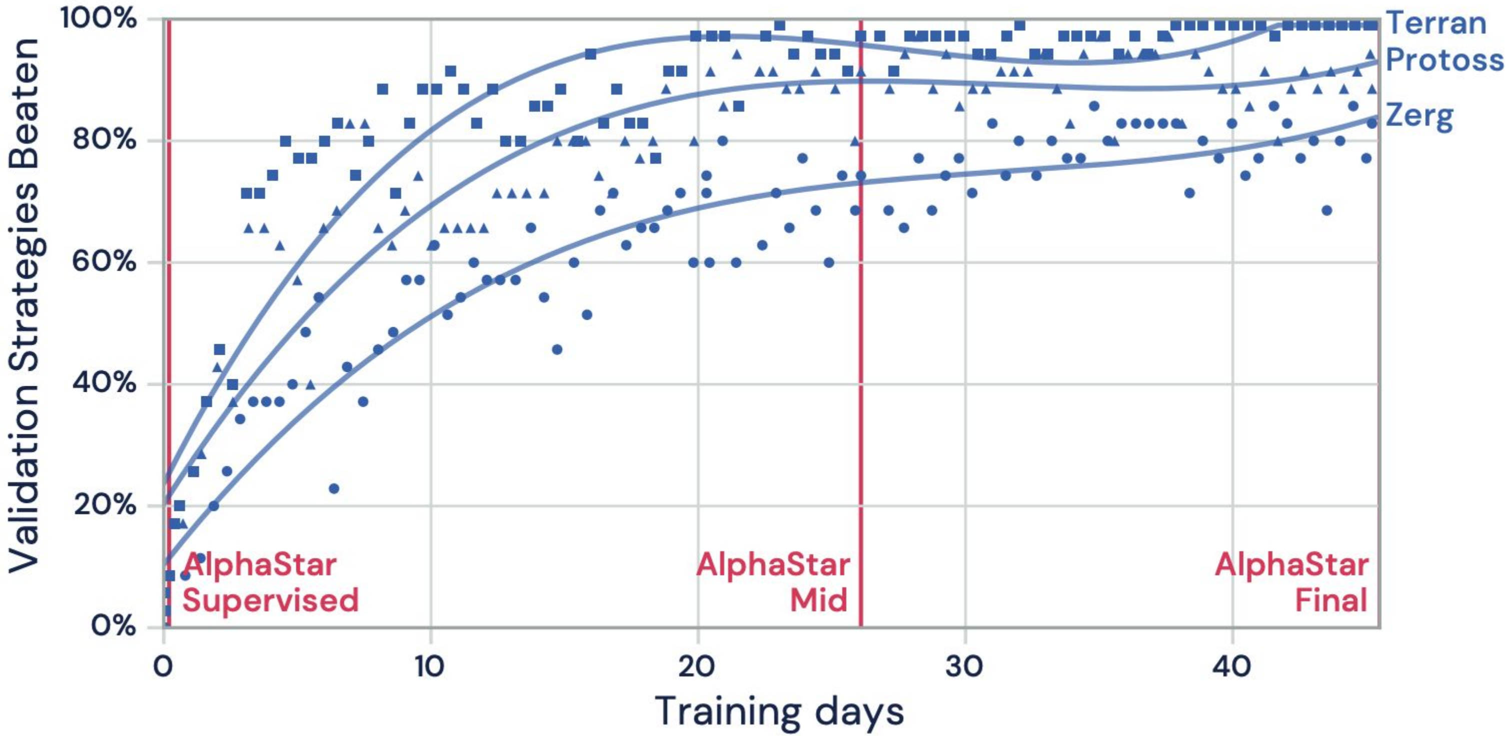


- Build strong, robust agents
- Expose main agents weaknesses
- Expose weaknesses of entire League

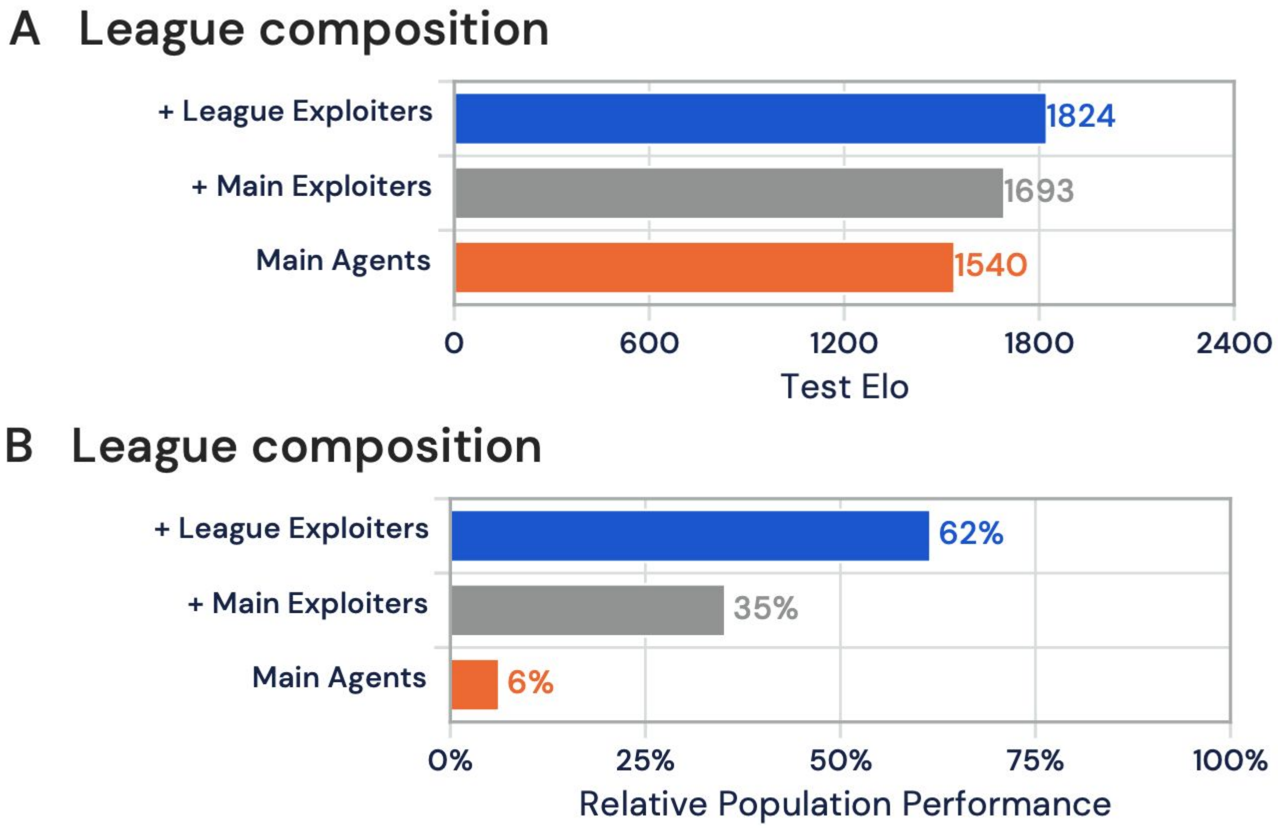
The League Training



The League Training



The League Training



Who to train against?

$$\forall_i \;\; \mathrm{P}(\pi_\theta \text{ winning against } \pi_i) > 0.5$$

Who to train against?

$$\forall_i \;\; \mathrm{P}(\pi_\theta \text{ winning against } \pi_i) > 0.5$$

- FSP $\quad \mathrm{U}(\{\pi_i\}_{i=1}^{N})$

Who to train against?

$$\forall_i \;\; \mathrm{P}(\pi_\theta \text{ winning against } \pi_i) > 0.5$$

- FSP $\mathrm{U}(\{\pi_i\}_{i=1}^{N})$
- PFSP $\mathrm{P}(\text{playing against } \pi_j) = \dfrac{f(\mathrm{P}(\pi_\theta \text{ winning } \pi_j))}{\sum_i f(\mathrm{P}(\pi_\theta \text{ winning } \pi_i))}$ $\quad f : [0,1] \to \mathbb{R}_+$

Who to train against?

$$\forall_i \ \ \mathrm{P}(\pi_\theta \text{ winning against } \pi_i) > 0.5$$

- FSP $\mathrm{U}(\{\pi_i\}_{i=1}^N)$
- **P**FSP $\mathrm{P}(\text{playing against } \pi_j) = \dfrac{f(\mathrm{P}(\pi_\theta \text{ winning } \pi_j))}{\sum_i f(\mathrm{P}(\pi_\theta \text{ winning } \pi_i))}$ $f : [0,1] \to \mathbb{R}_+$

$$f_{\text{hard}}(x) = (1-x)^p$$

You **never** play against opponents that you **dominate**.

You **focus** on **beating everyone** rather than average win rate.

When a **rare**, but **strong**, opponent appears – it is being **focused** on.

Who to train against?

$$\forall_i \ \ \mathrm{P}(\pi_\theta \text{ winning against } \pi_i) > 0.5$$

- FSP $\quad \mathrm{U}(\{\pi_i\}_{i=1}^{N})$
- **P**FSP $\quad \mathrm{P}(\text{playing against } \pi_j) = \dfrac{f(\mathrm{P}(\pi_\theta \text{ winning } \pi_j))}{\sum_i f(\mathrm{P}(\pi_\theta \text{ winning } \pi_i))} \qquad f : [0,1] \to \mathbb{R}_+$

$$f_{\text{hard}}(x) = (1-x)^p$$

You **never** play against opponents that you **dominate**.

You **focus** on **beating everyone** rather than average win rate.

When a **rare**, but **strong**, opponent appears – it is being **focused** on.

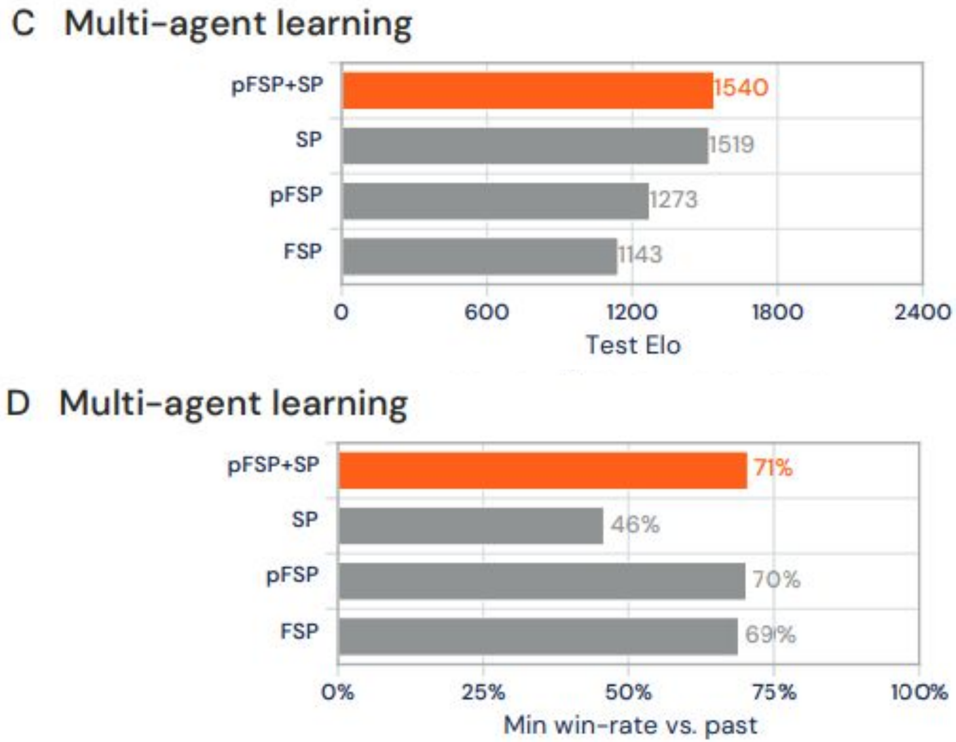
$$f_{\text{var}}(x) = (1-x)x$$

You always pick **opponents** at your **own level**.

Creates a natural auto **curriculum**.

A **black-box** version of **TD-error prioritisation**.

Who to train against?



Path matters

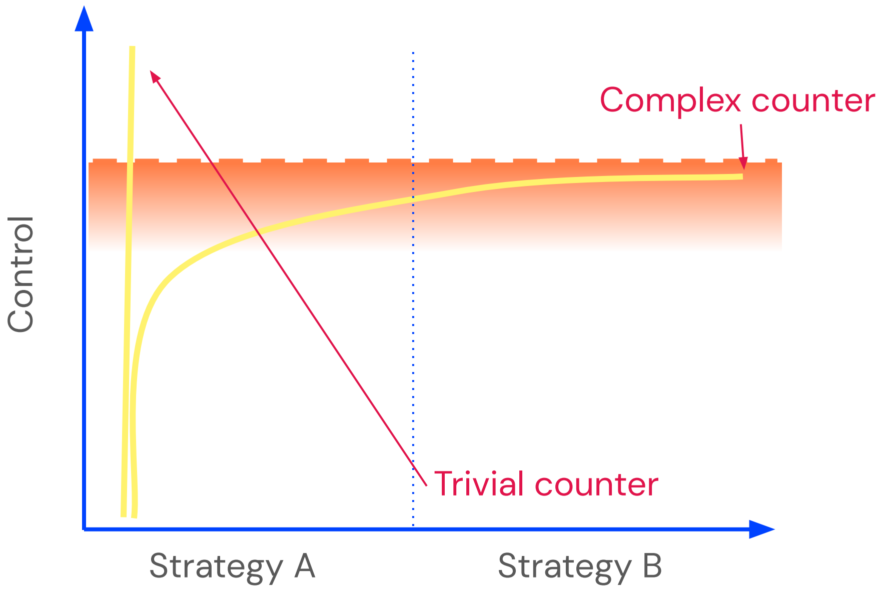
- There are often infinitely many solutions for “best response to a fixed set of opponents” problem
- “Greedy” decisions on the way identify which one we will end up with
- They might differ dramatically with respect to their transitive strength and properties.
- “Hard opponents” can prefer policies of low transitive strength, but converges fast and to diverse policies.

$$f_{\text{hard}}(x) = (1-x)^p$$

- “Variance” produces more “standard” strategies, but converges much slower (and somewhat deterministically).

$$f_{\text{var}}(x) = (1-x)x$$

Path matters



Mutli-agent Deep Reinforcemenet Learning

!=

Multi-agent +
Reinforcement learning +
Deep learning

It is a "new field"

- Choice of correct opponents is not just guided towards convergence to the Nash, but also takes into consideration dynamics of Deep RL
- RL needs to be curated towards specifics of Multi-agent, e.g. rapid changes of targets, non-stationarity
- Exploration is not just an RL issue, with multi-agent algorithms we can guide the weak exploration strategy to shine in a complex problem
- Architectures can create entire new levels of multi-agency
- Architectures, and improvements that are the best in simplified setups are not the ones that shine in the long term - speed of convergence is a wrong thing to optimise!
- Even the game interface shapes the dynamics of RL, and multi-agent!

1. Complex Combinatorial Action Space
2. Multi-modal Observation Space
3. Information poverty and Hard Exploration
4. Human “Alignment”
5. Multiple Interacting Agents

- ➔ human-like constraints
- ➔ architecture
- ➔ new RL objectives
- ➔ “human exploration”
- ➔ AlphaStar League
- ➔ A lot of hard teamwork!

Questions?