GRANDMASTER LEVEL IN STARCRAFT II USING MULTI-AGENT REINFORCEMENT LEARNING

(2019,NATURE)

JuHyeong Kim

INTRODUCTION

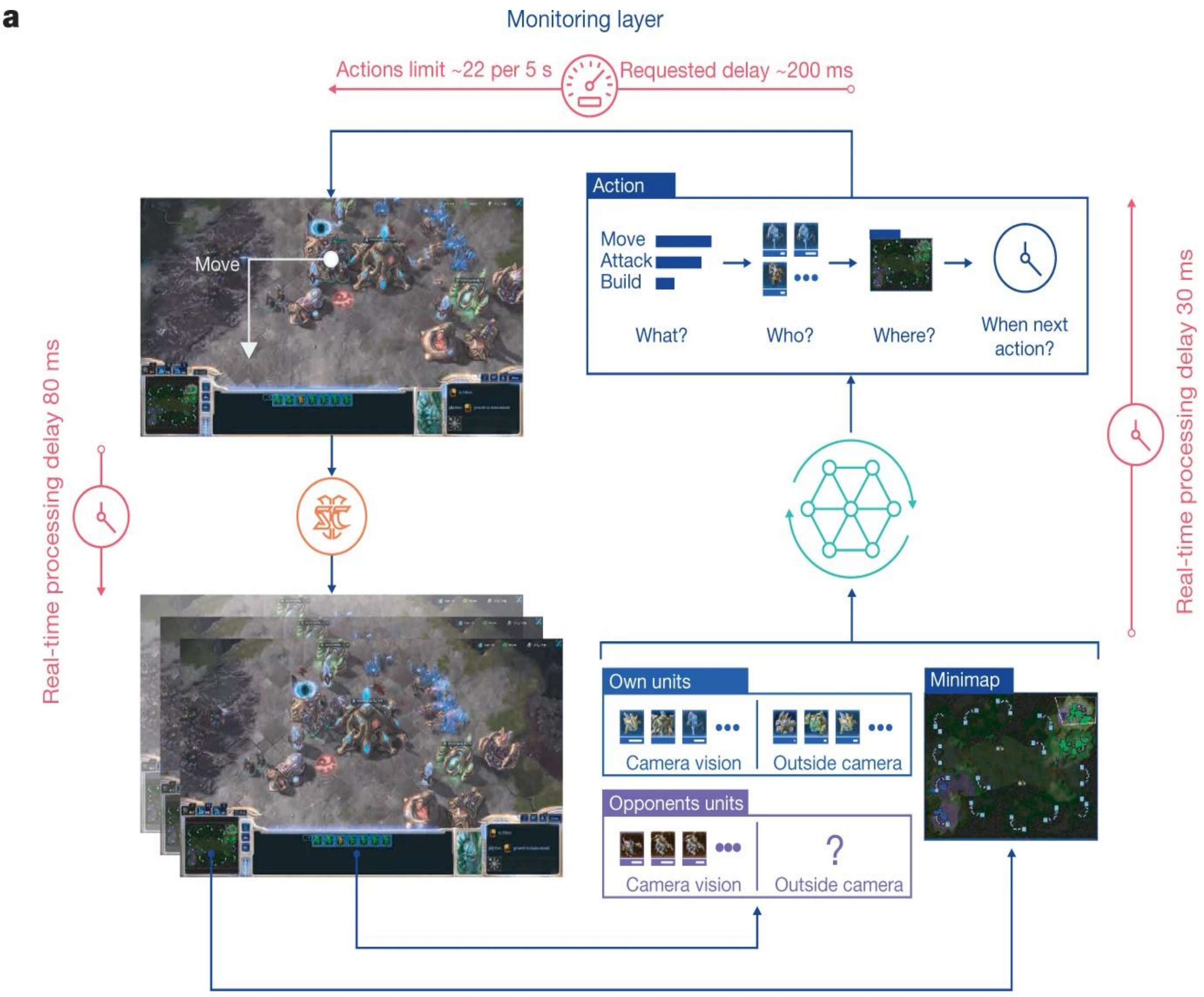
- Many real-world applications require artificial agents to compete and coordinate with other agents in complex environments.

- We chose to address the challenge of StarCraft using general-purpose learning methods that are in principle applicable to other complex domains:
  - a multi-agent reinforcement learning algorithm that uses data from both human and agent games within a diverse league of continually adapting strategies and counter-strategies, each represented by deep neural networks.

- We evaluated our agent, **AlphaStar**, in the full game of StarCraft II, through a series of online games against human players.

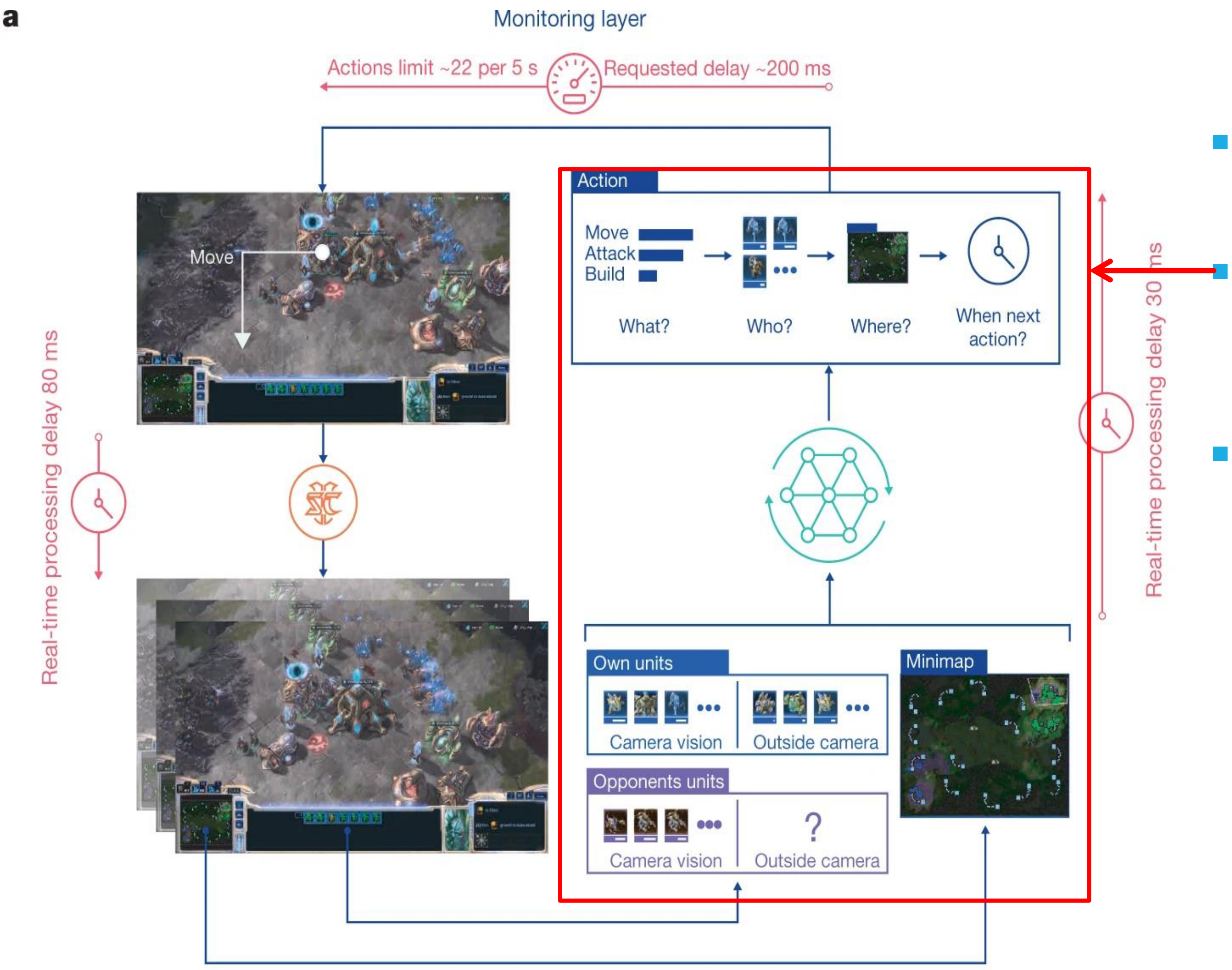
- this model is Supervised learning + Reinforcement learning

ARCHITECTURE OVERVIEW



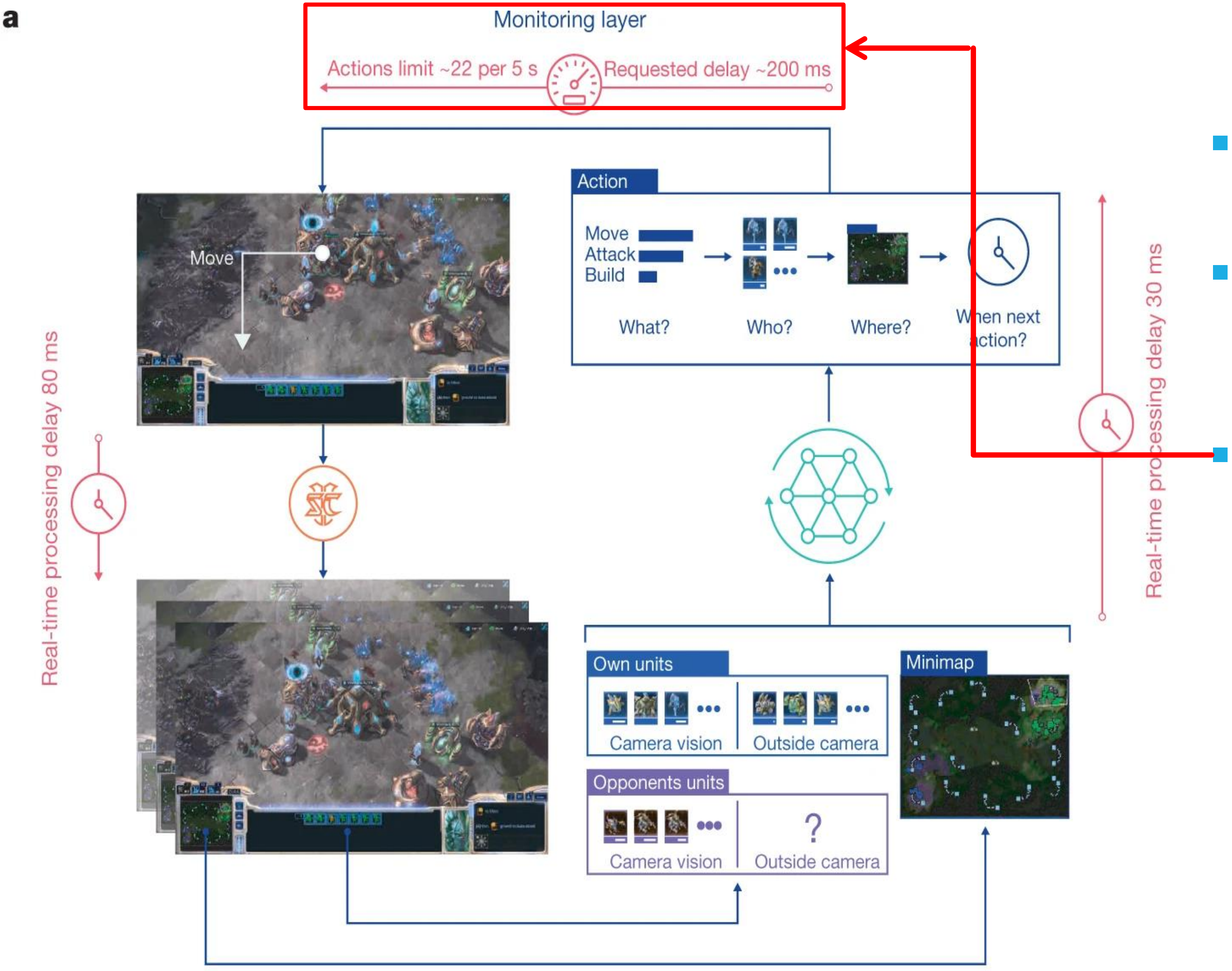
- **a**, AlphaStar observes the game through an overview map and list of units.
- 1) To act, the agent outputs what action type to issue (for example, build), who it is applied to, where it targets, and when the next action will be issued.
- 2) Actions are sent to the game through a monitoring layer that limits action rate. AlphaStar contends with delays from network latency and processing time.

ARCHITECTURE OVERVIEW



- **a**, AlphaStar observes the game through an overview map and list of units.
- 1) To act, the agent outputs what action type to issue (for example, build), who it is applied to, where it targets, and when the next action will be issued.
- 2) Actions are sent to the game through a monitoring layer that limits action rate. AlphaStar contends with delays from network latency and processing time.

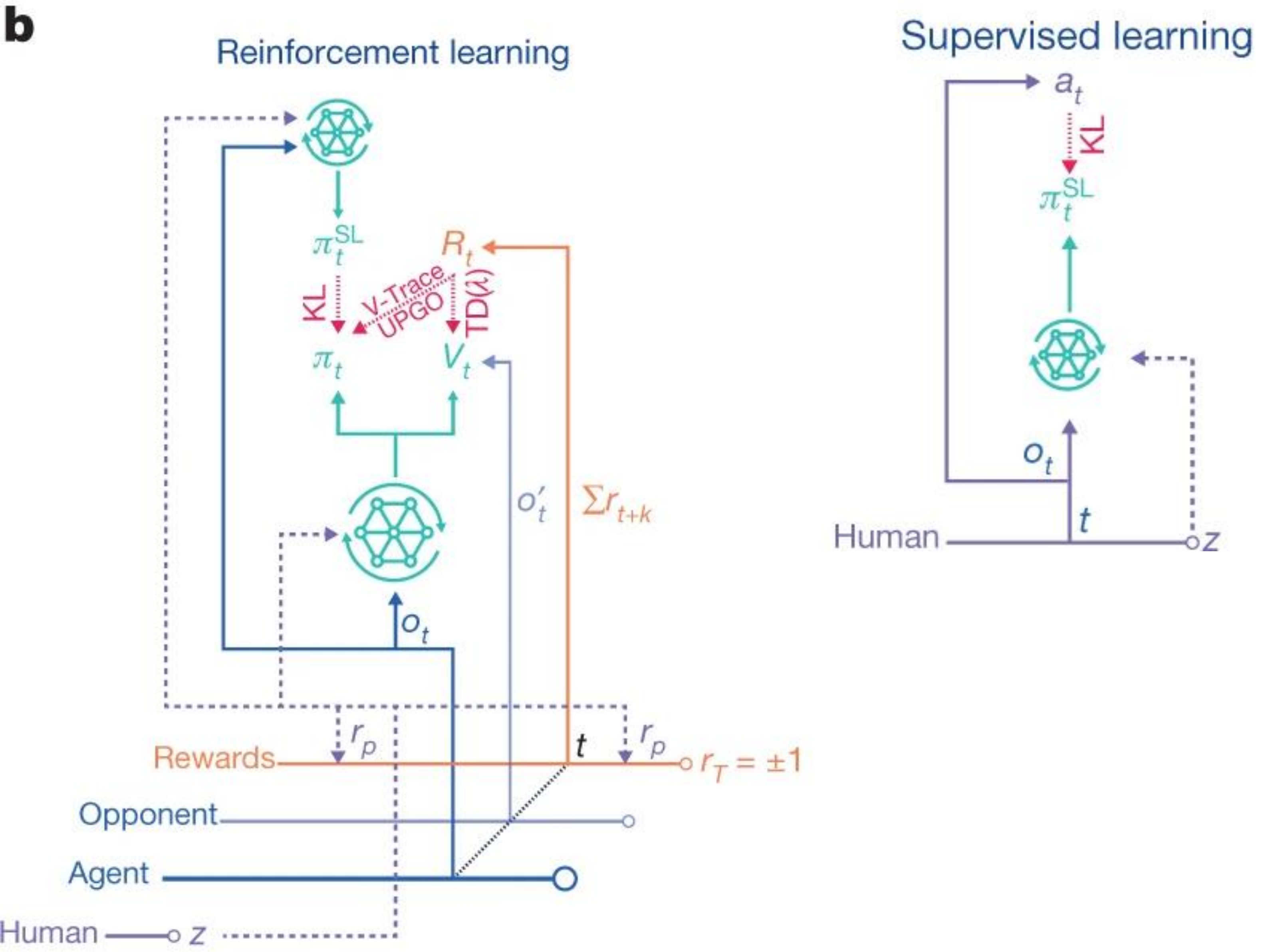
ARCHITECTURE OVERVIEW



- **a**, AlphaStar observes the game through an overview map and list of units.
- 1) To act, the agent outputs what action type to issue (for example, build), who it is applied to, where it targets, and when the next action will be issued.
- 2) Actions are sent to the game through a monitoring layer that limits action rate. AlphaStar contends with delays from network latency and processing time.

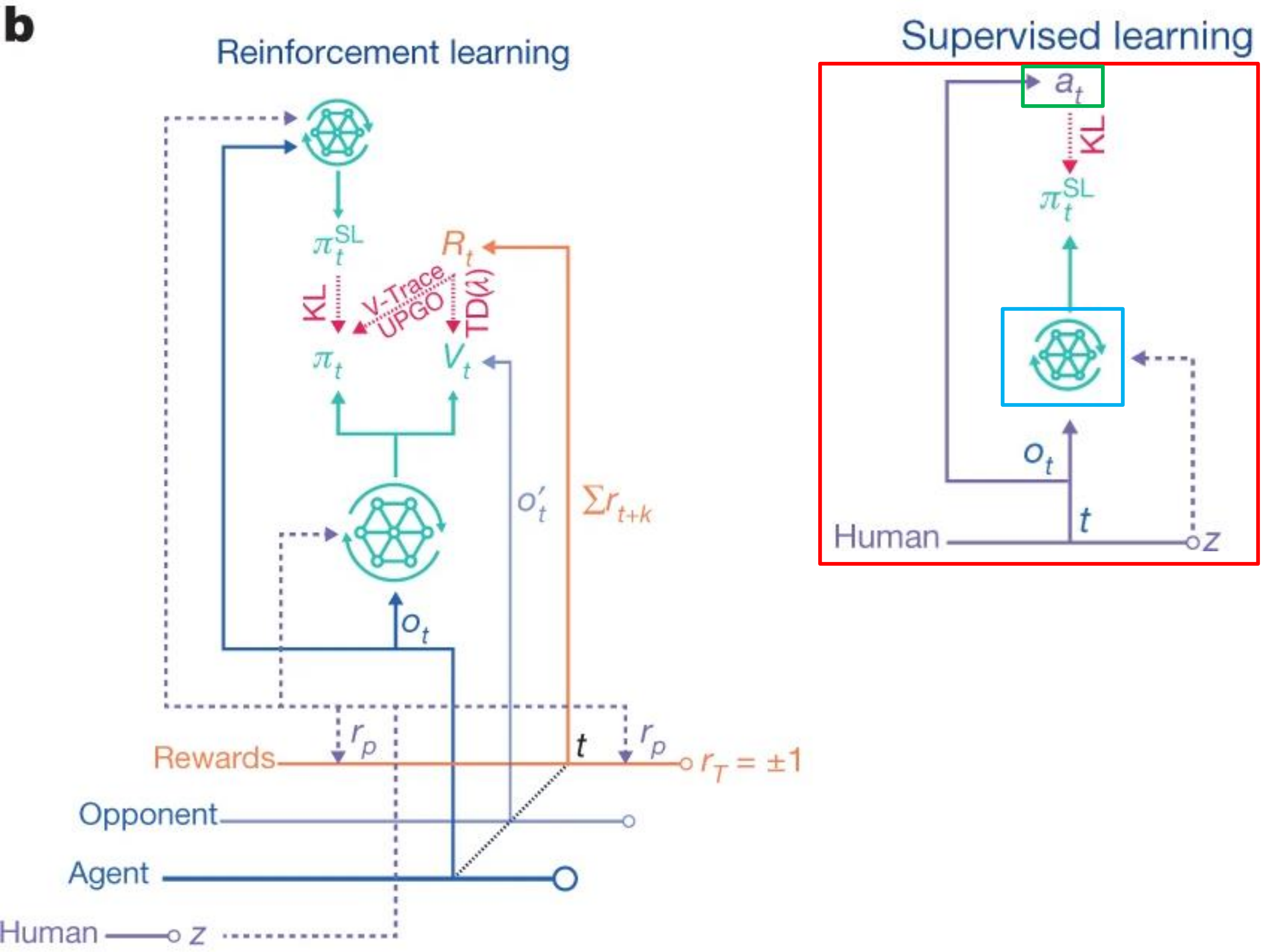
Humans play StarCraft under physical constraints that limit their reaction time and the rate of their actions.

ARCHITECTURE OVERVIEW



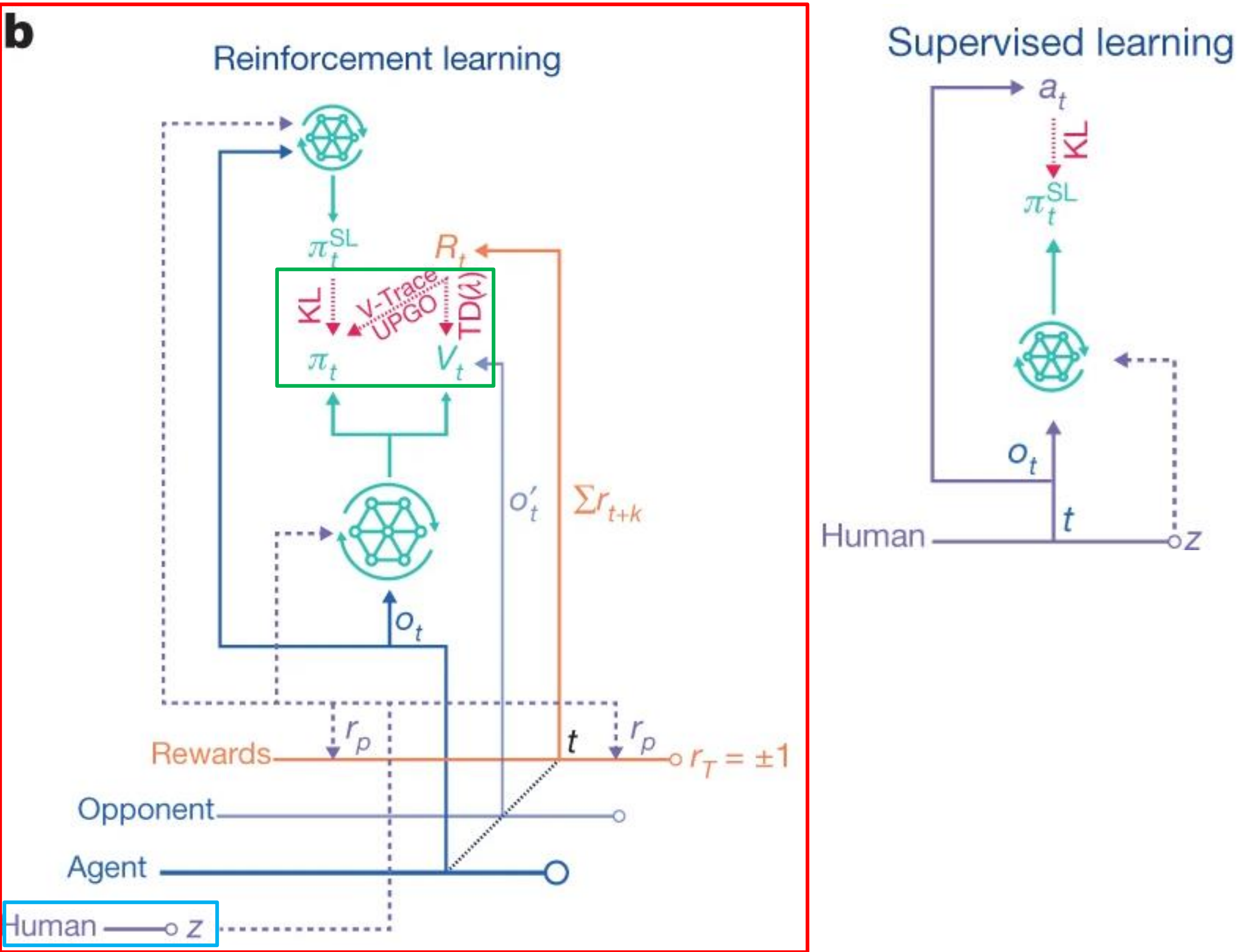
- **b**, AlphaStar is trained via both supervised learning and reinforcement learning.
- In supervised learning, the parameters are updated to optimize Kullback–Leibler (KL) divergence between its output and human actions sampled from a collection of replays.
- In reinforcement learning, human data are used to sample the statistic *z*, and agent experience is collected to update the policy and value outputs via reinforcement learning (TD(λ), V-trace, UPGO) combined with a KL loss towards the supervised agent.

ARCHITECTURE OVERVIEW



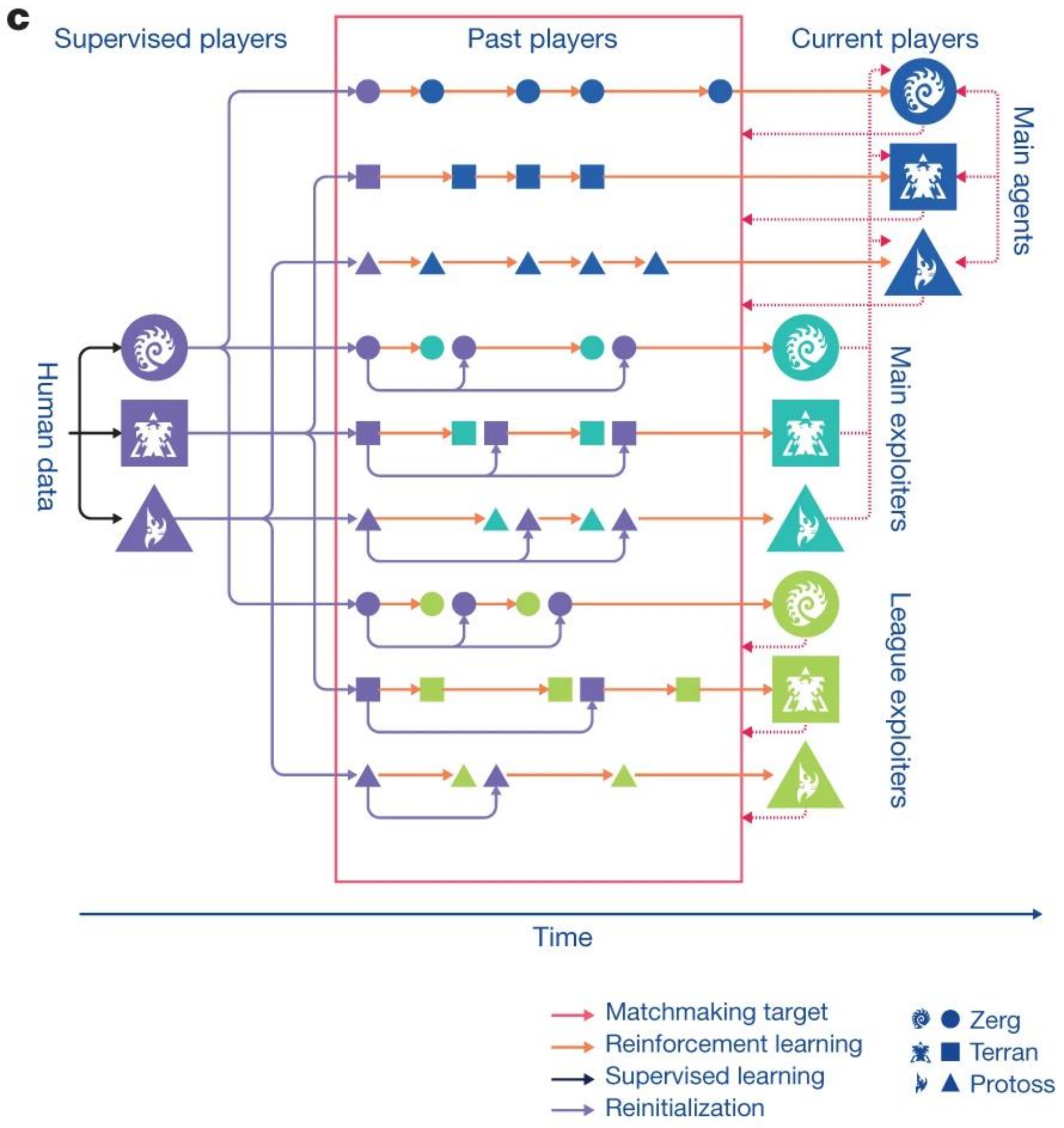
- **b**, AlphaStar is trained via both supervised learning and reinforcement learning.
- In supervised learning, the parameters are updated to optimize Kullback–Leibler (KL) divergence between its output and human actions sampled from a collection of replays.
- In reinforcement learning, human data are used to sample the statistic *z*, and agent experience is collected to update the policy and value outputs via reinforcement learning (TD(λ), V-trace, UPGO) combined with a KL loss towards the supervised agent.

ARCHITECTURE OVERVIEW



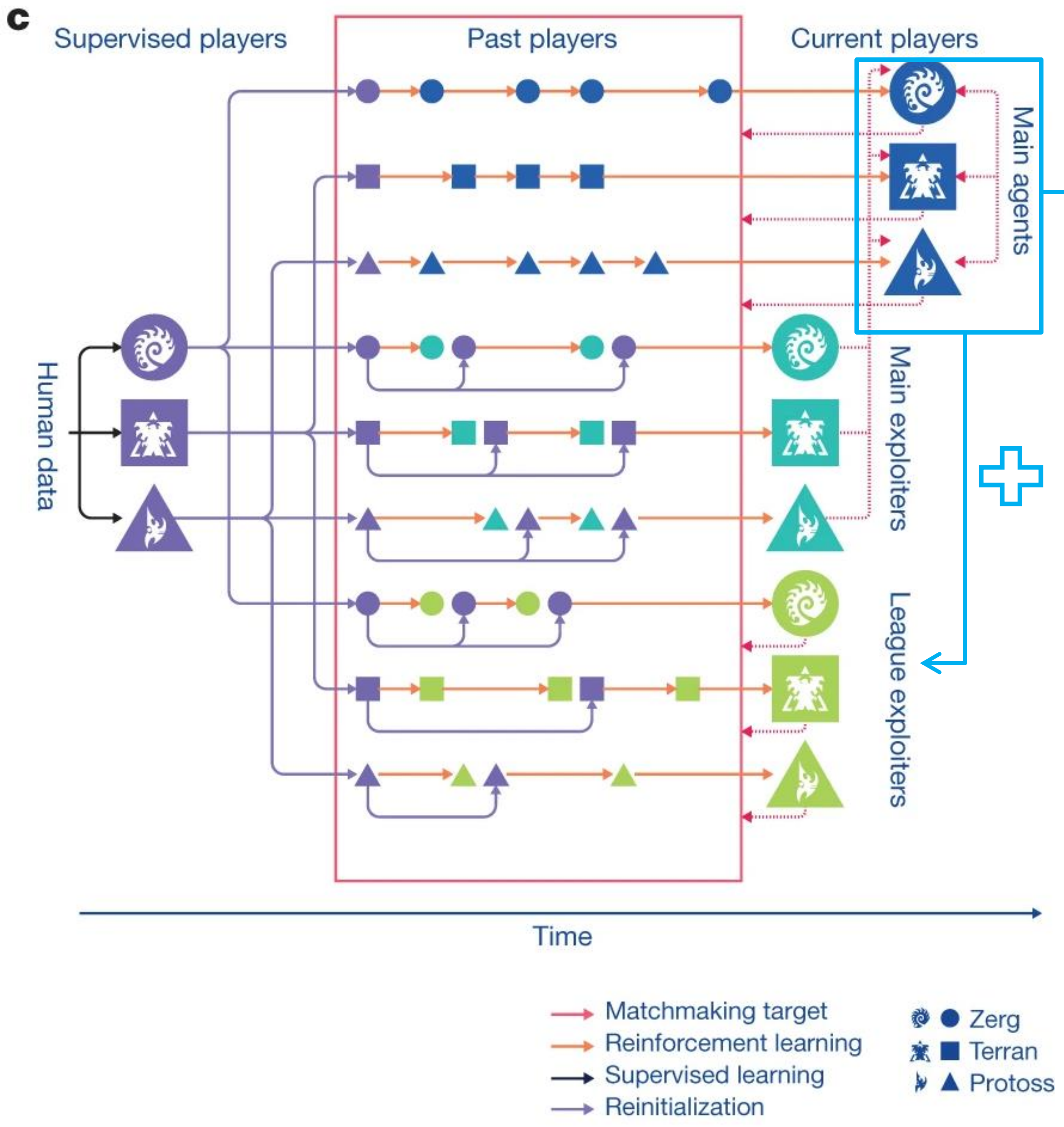
- **b**, AlphaStar is trained via both supervised learning and reinforcement learning.
- In supervised learning, the parameters are updated to optimize Kullback–Leibler (KL) divergence between its output and human actions sampled from a collection of replays.
- In reinforcement learning, human data are used to sample the statistic *z*, and agent experience is collected to update the **policy and value outputs** via reinforcement learning (TD(λ), V-trace, UPGO) combined with a KL loss towards the supervised agent.

ARCHITECTURE OVERVIEW



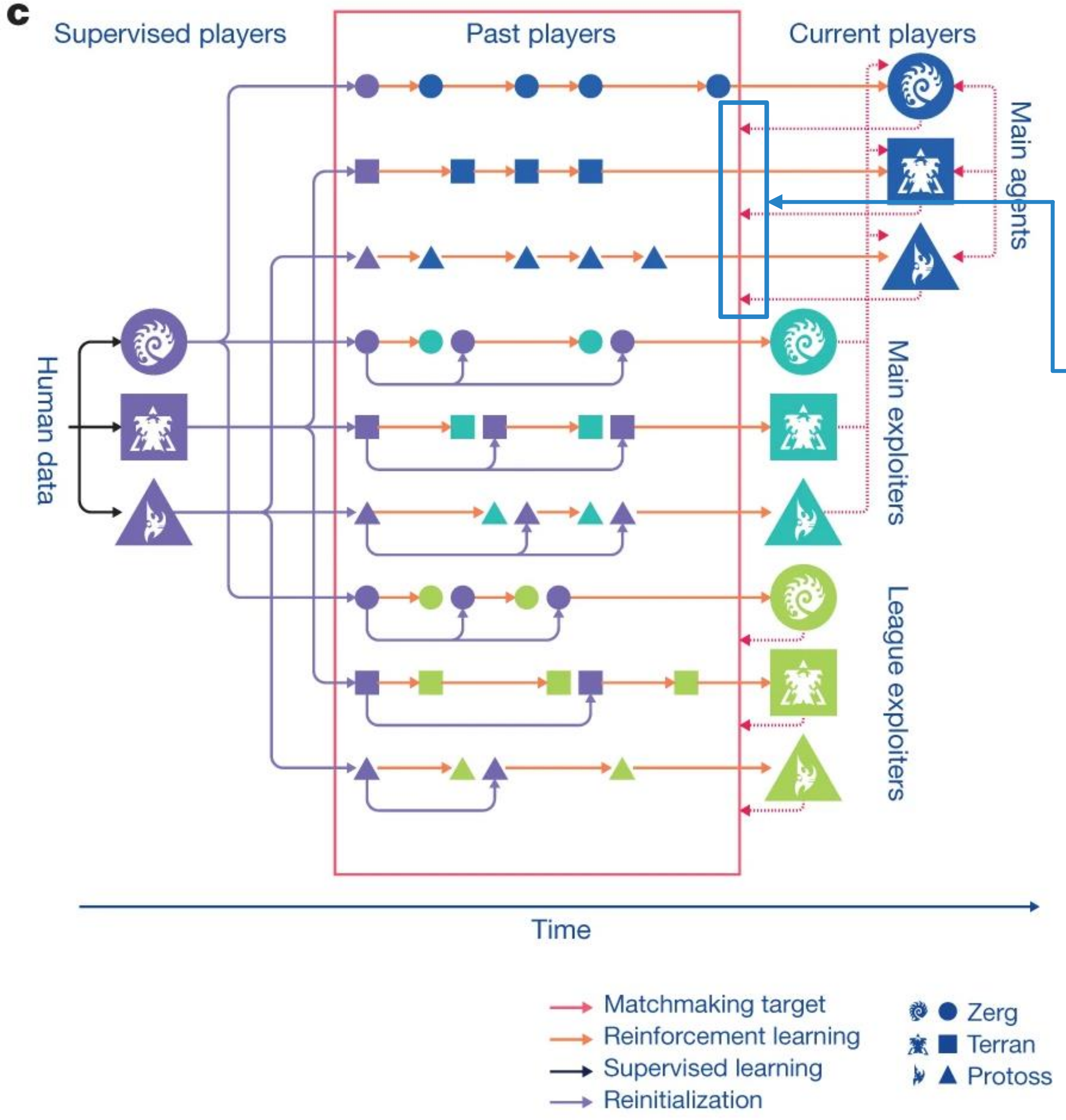
- **c**, Three pools of agents, each initialized by supervised learning, were subsequently trained with reinforcement learning.
- As they train, these agents intermittently add copies of themselves—'players' that are frozen at a specific point—to the league.
- The main agents train against all of these past players, as well as themselves.
- The main exploiters train against the main agents.
  - Complement the weaknesses of the main agent.
- The league exploiters train against all past players.
- Main exploiters and league exploiters can be reset to the supervised agent when they add a player to the league.

ARCHITECTURE OVERVIEW



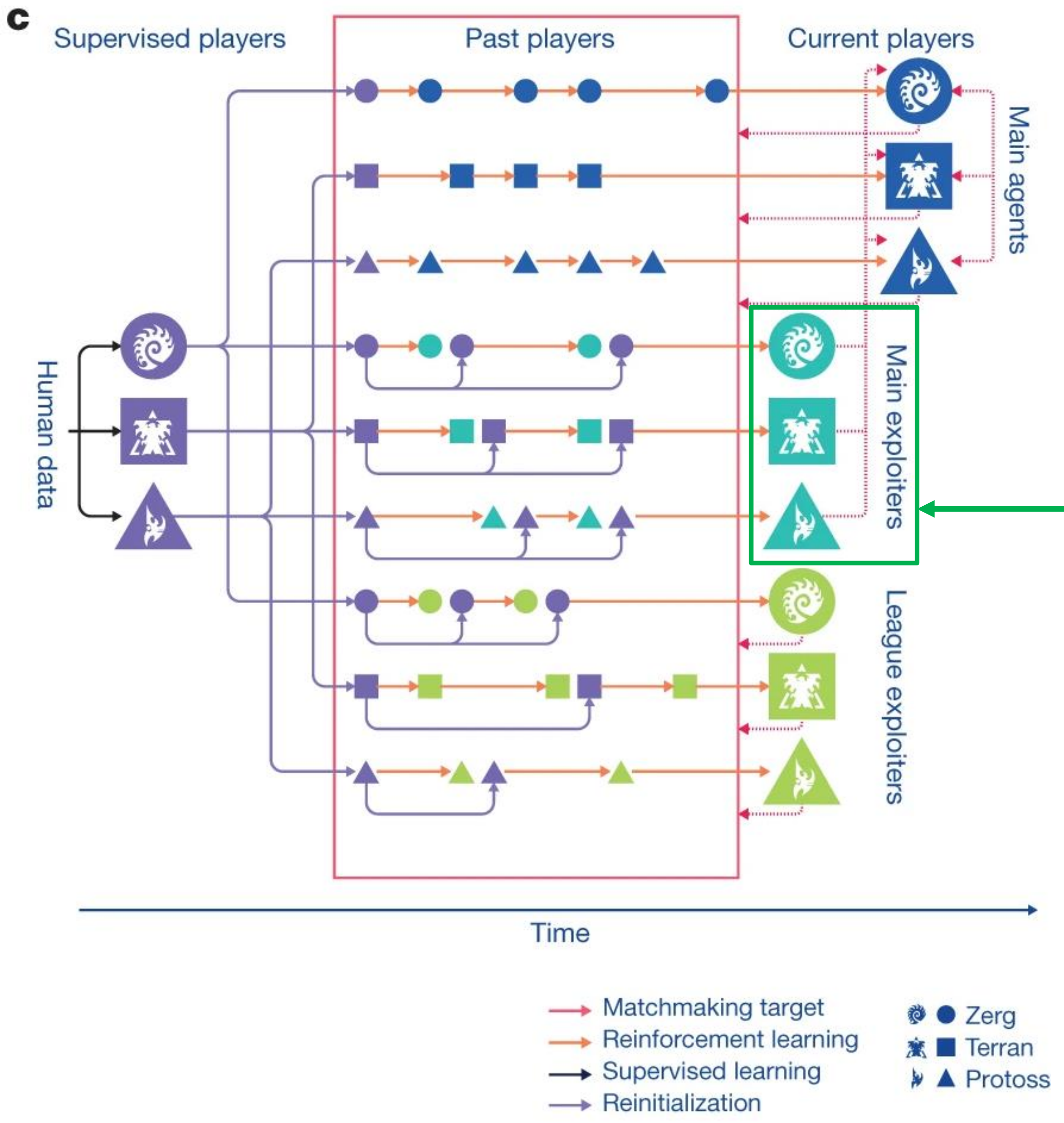
- **c, Three pools of agents,** each initialized by supervised learning, were subsequently trained with reinforcement learning.
- As they train, these agents intermittently add copies of themselves—'players' that are frozen at a specific point—to the league.
- The main agents train against all of these past players, as well as themselves.
- The main exploiters train against the main agents.
  - Complement the weaknesses of the main agent.
- The league exploiters train against all past players.
- Main exploiters and league exploiters can be reset to the supervised agent when they add a player to the league.

ARCHITECTURE OVERVIEW



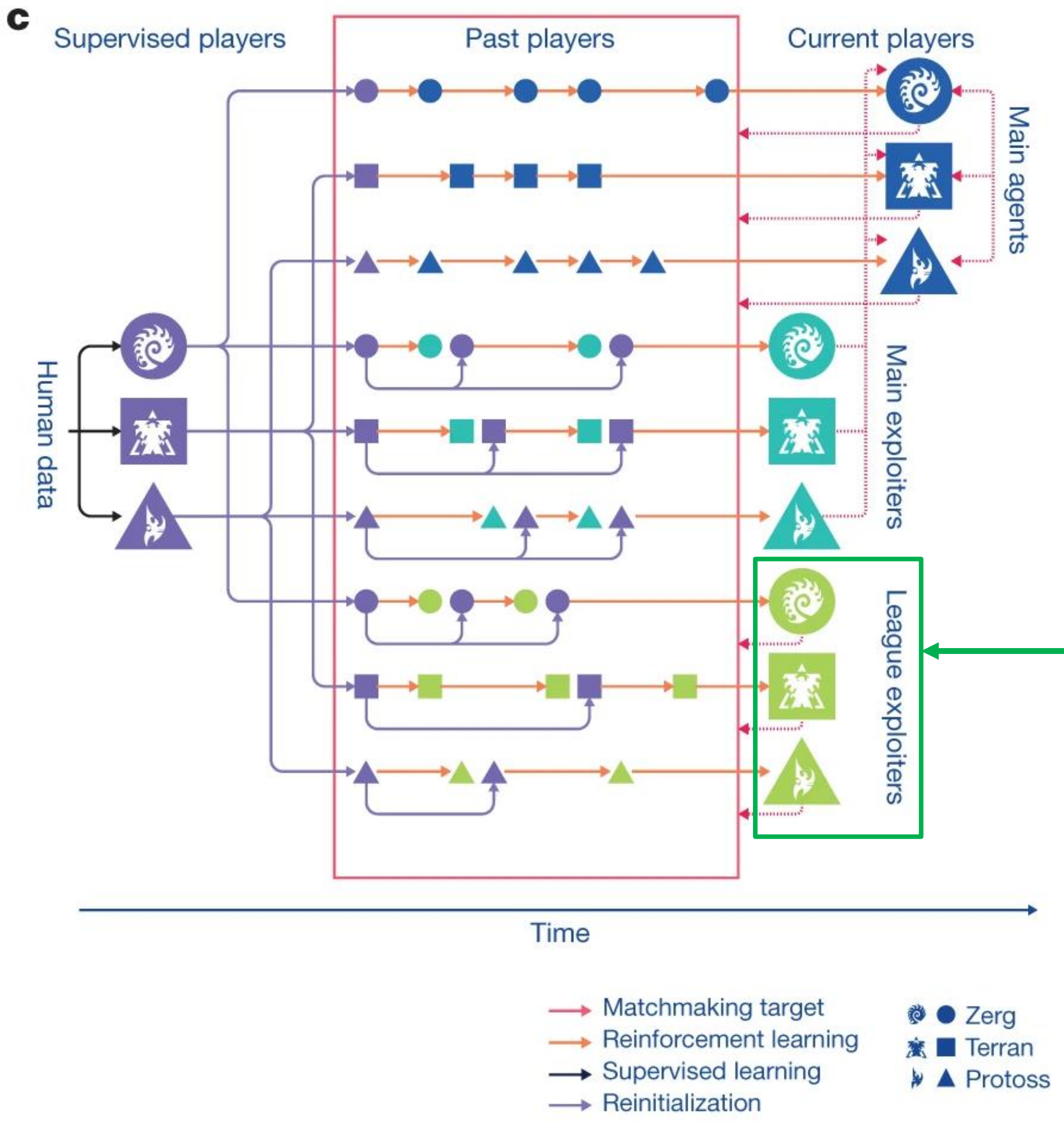
- **c, Three pools of agents**, each initialized by supervised learning, were subsequently trained with reinforcement learning.
- As they train, these agents intermittently add copies of themselves—'players' that are frozen at a specific point—to the league.
- The main agents train against all of these past players, as well as themselves.
- The main exploiters train against the main agents.
  - Complement the weaknesses of the main agent.
- The league exploiters train against all past players.
- Main exploiters and league exploiters can be reset to the supervised agent when they add a player to the league.

ARCHITECTURE OVERVIEW



- **c**, Three pools of agents, each initialized by supervised learning, were subsequently trained with reinforcement learning.
- As they train, these agents intermittently add copies of themselves—'players' that are frozen at a specific point—to the league.
- The main agents train against all of these past players, as well as themselves.
- The main exploiters train against the main agents.
  - Complement the weaknesses of the main agent.
- The league exploiters train against all past players.
- Main exploiters and league exploiters can be reset to the supervised agent when they add a player to the league.

ARCHITECTURE OVERVIEW



- **c**, Three pools of agents, each initialized by supervised learning, were subsequently trained with reinforcement learning.
- As they train, these agents intermittently add copies of themselves—'players' that are frozen at a specific point—to the league.
- The main agents train against all of these past players, as well as themselves.
- The main exploiters train against the main agents.
  - Complement the weaknesses of the main agent.
- The league exploiters train against all past players.
- Main exploiters and league exploiters can be reset to the supervised agent when they add a player to the league.

SUPERVISED LEARNING

- Each agent is initially trained through supervised learning on **replays to imitate human actions**.
- Supervised learning is used both to initialize the agent and to maintain diverse exploration.
  - Because of this, the primary goal is to produce a diverse policy that captures StarCraft's complexities.
- From each replay, we extract **a statistic *z*** that encodes each player's build order, defined as the first 20 constructed buildings and units, and cumulative statistics, defined as the units, buildings, effects, and upgrades that were present during a game.

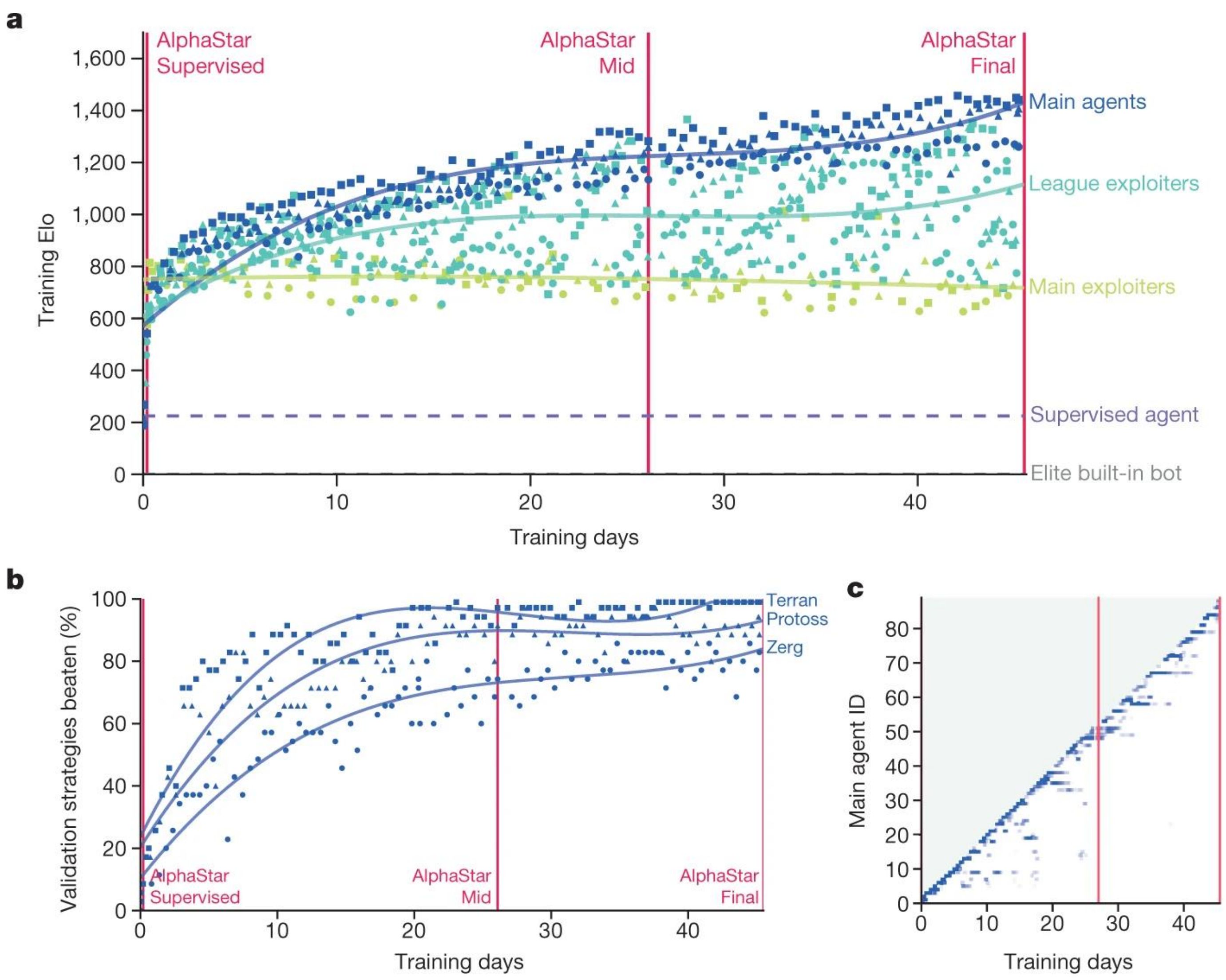
LEARNING ALGORITHM

- Agent parameters were initially trained by supervised learning.
- Games were sampled from a publicly available **dataset of anonymized human replays**.
- The policy was then trained to **predict each action a_t, conditioned either solely on s_t, or also on z**.
- This results in a diverse set of strategies that reflects the modes of human play.

LEARNING ALGORITHM

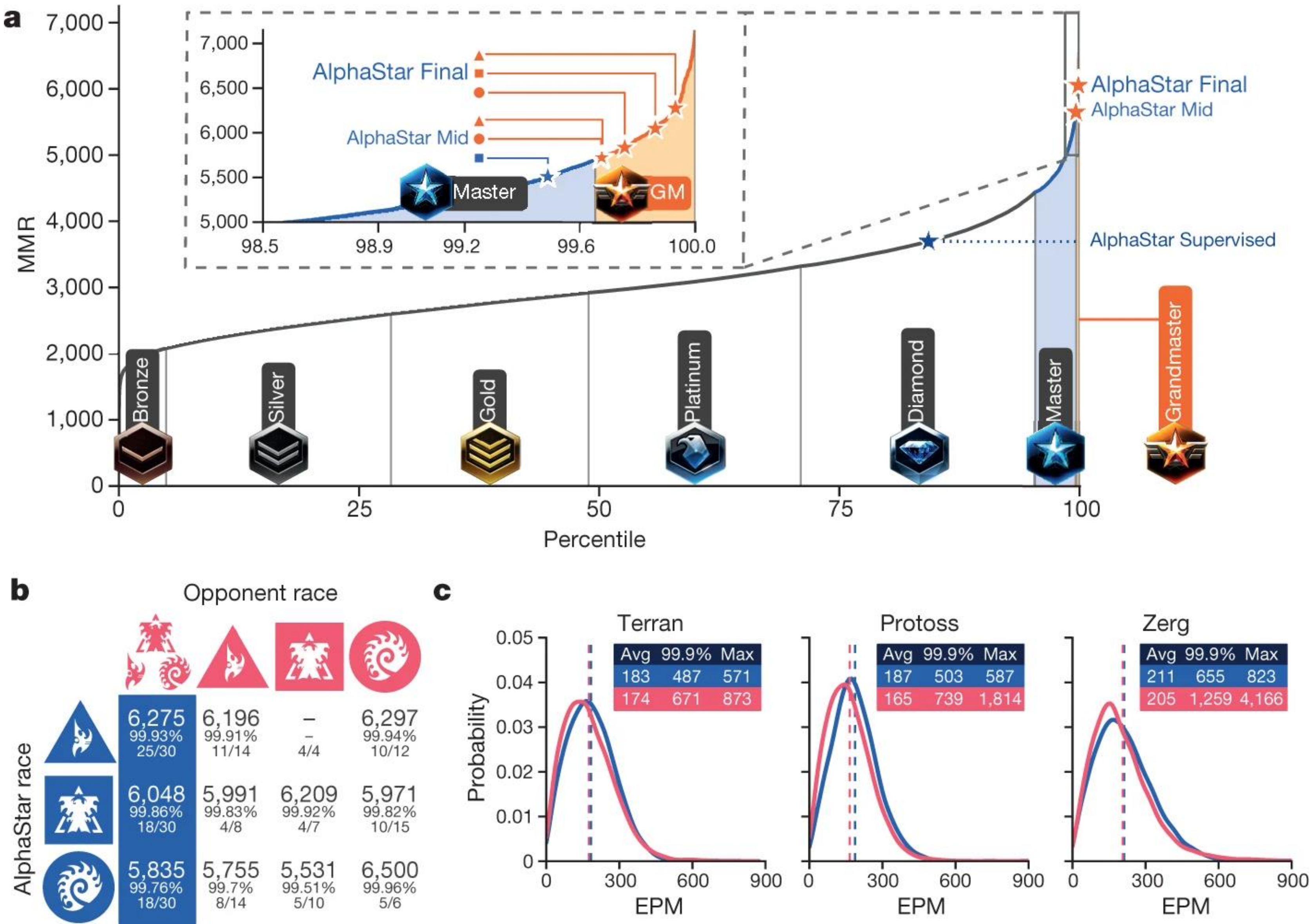
- The agent parameters were subsequently trained by a reinforcement learning algorithm that is designed to maximize the win rate (that is, compute a best response) against a mixture of opponents.
- The choice of opponent is determined by a multi-agent procedure, described below.
- AlphaStar's reinforcement learning algorithm is **based on a policy gradient algorithm similar to advantage actor–critic.** Updates were applied asynchronously on replayed experiences.
- This requires an approach known as off-policy learning, that is, updating the current policy from experience generated by a previous policy.
- **We therefore use a combination of techniques** that can learn effectively despite the mismatch: temporal difference learning **(TD(λ)),** clipped importance sampling **(V-trace),** and a new self-imitation algorithm **(UPGO)** that moves the policy towards trajectories with better-than-average reward.
- To reduce variance, during training only, the value function is estimated using information from both the player's and the opponent's perspectives.

ALPHASTAR TRAINING PROGRESSION



- **a,** Training Elo scores of agents in the league during the 44 days of training
  - Each point represents a past player, evaluated a gainst the entire league and the elite built-in b ot (whose Elo is set to 0)
- Each agent was assessed at three different s napshots during training:
- after supervised training only
  - AlphaStar Supervised
- after 27 days of league training
  - AlphaStar Mid
- after 44 days of league training
  - AlphaStar Final

RESULTS


a
MMR
Percentile
AlphaStar Final
AlphaStar Mid
AlphaStar Supervised
Master
GM
Bronze
Silver
Gold
Platinum
Diamond
Master
Grandmaster
b
Opponent race
AlphaStar race
6,275 99.93% 25/30
6,196 99.91% 11/14
– – 4/4
6,297 99.94% 10/12
6,048 99.86% 18/30
5,991 99.83% 4/8
6,209 99.92% 4/7
5,971 99.82% 10/15
5,835 99.76% 18/30
5,755 99.7% 8/14
5,531 99.51% 5/10
6,500 99.96% 5/6
c
Terran
Protoss
Zerg
Probability
EPM
Avg 99.9% Max
183 487 571
174 671 873
187 503 587
165 739 1,814
211 655 823
205 1,259 4,166

REFERENCE

- https://www.nature.com/articles/s41586-019-1724-z#MOESM2
- https://www.youtube.com/watch?v=6Thu5vlDc6Y&feature=emb_logo