Adaptive Q-Network: On-the-fly Target Selection for Deep Reinforcement Learning

Théo Vincent

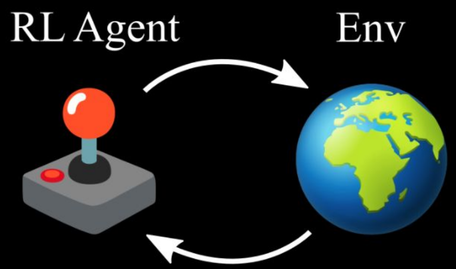
Fabian Wahren

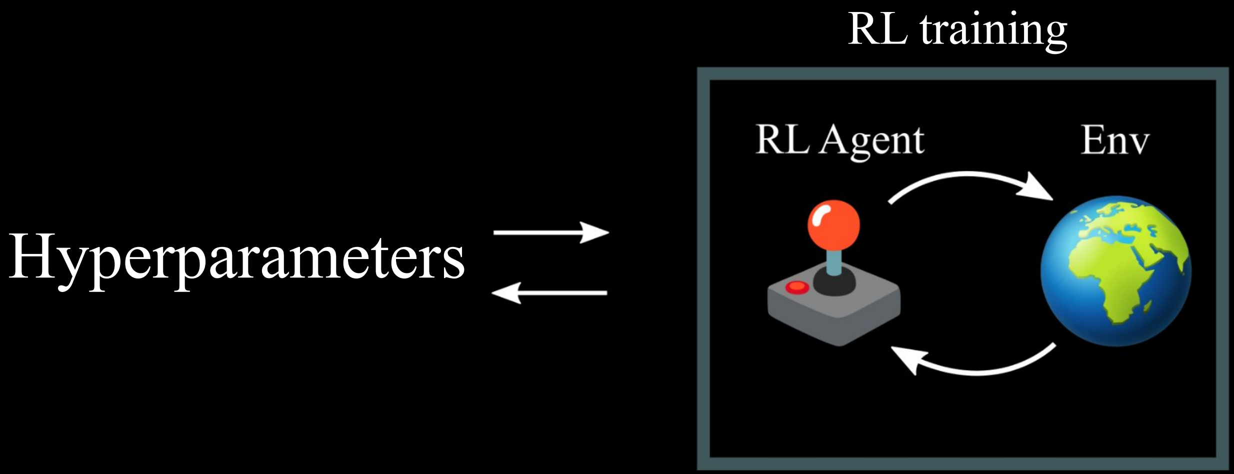
Jan Peters

Boris Belousov

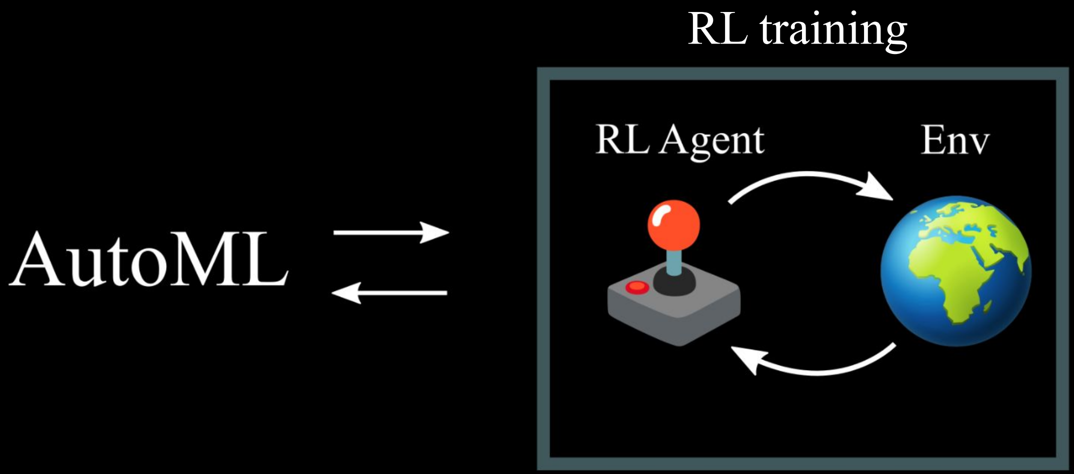
Carlo D'Eramo


An AutoML algorithm tailored for RL


RL Agent
Env

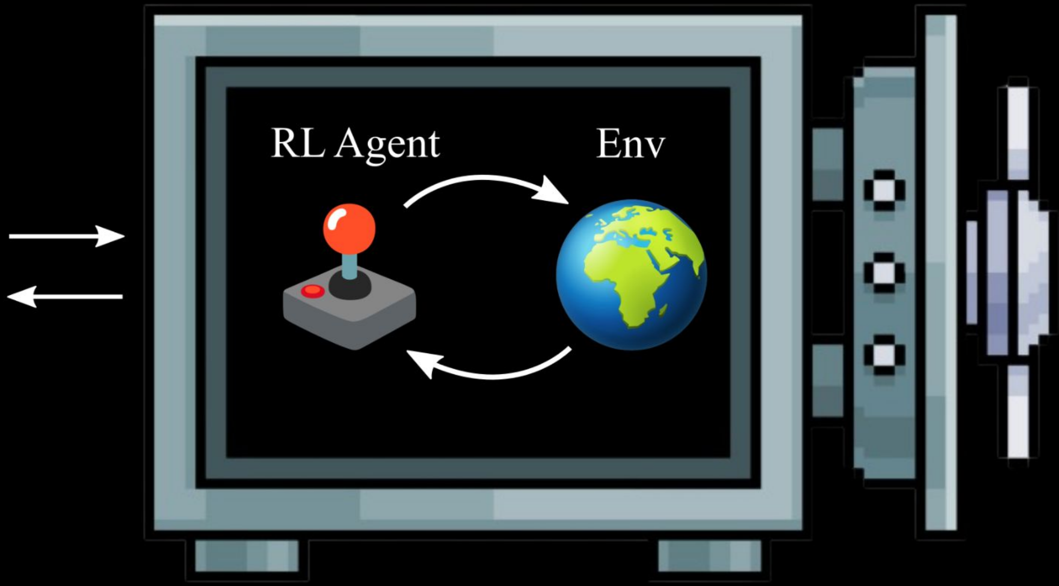

RL training
RL Agent
Env
Hyperparameters

Automated Machine Learning for RL

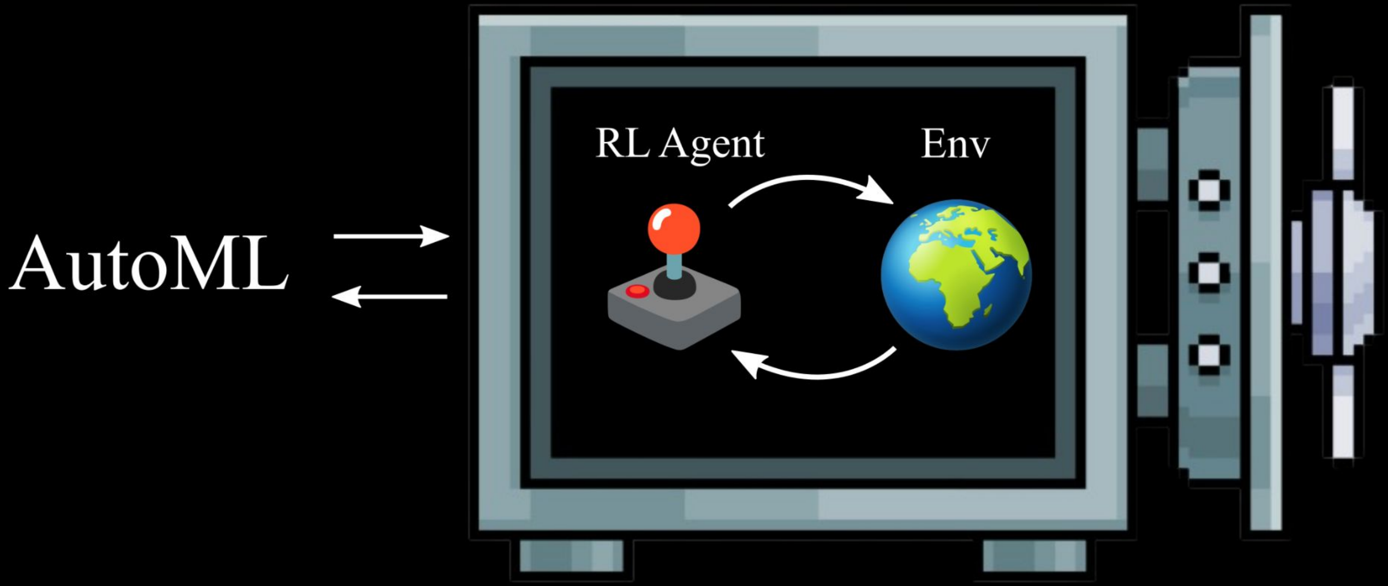


Automated Machine Learning for RL

AutoML



Automated Machine Learning for RL



Ignores RL nonstationarities

Q-learning

Q-learning

$\bar{Q}_0$

Target network

Q-learning

$$\Gamma \bar{Q}_0$$

Target

Q-learning

$$\Gamma\bar{Q}_0 \longrightarrow Q_1$$

Online network

$$\mathcal{L}(\theta \mid \bar{\theta}) = ||\Gamma\bar{Q}_0 - Q_1||$$

Q-learning

$$\Gamma\bar{Q}_0 \longrightarrow Q_1 | \Gamma\bar{Q}_1$$

Target update $\quad \bar{\theta} \leftarrow \theta$

$$\mathcal{L}(\theta \mid \bar{\theta}) = ||\Gamma\bar{Q}_1 - Q_1||$$

Q-learning

$$\Gamma\bar{Q}_0 \longrightarrow Q_1 | \Gamma\bar{Q}_1$$

Target update $\bar{\theta} \leftarrow \theta$

$$\mathcal{L}(\theta \mid \bar{\theta}) = ||\Gamma\bar{Q}_1 - Q_1||$$

Brakes stationarity

Q-learning

$$\Gamma\bar{Q}_0 \longrightarrow Q_1 \,|\, \Gamma\bar{Q}_1 \longrightarrow Q_2$$

$$\mathcal{L}(\theta \mid \bar{\theta}) = ||\Gamma\bar{Q}_1 - Q_2||$$

Q-learning

$$\Gamma\bar{Q}_0 \longrightarrow Q_1 \,|\, \Gamma\bar{Q}_1 \longrightarrow Q_2 \,|\, \Gamma\bar{Q}_2$$

$$\bar{\theta} \leftarrow \theta$$

$$\mathcal{L}(\theta \mid \bar{\theta}) = ||\Gamma\bar{Q}_2 - Q_2||$$

Brakes stationarity

Q-learning

$$\Gamma\bar{Q}_0 \longrightarrow Q_1 \,|\, \Gamma\bar{Q}_1 \longrightarrow Q_2 \,|\, \Gamma\bar{Q}_2 \longrightarrow \cdots$$

Naive approach using AutoML for Q-learning

$$\Gamma\bar{Q}_0 \longrightarrow Q_1 \,|\, \Gamma\bar{Q}_1 \longrightarrow Q_2 \,|\, \Gamma\bar{Q}_2 \longrightarrow \cdots$$

$$\Gamma\bar{Q}_0 \longrightarrow Q_1 \,|\, \Gamma\bar{Q}_1 \longrightarrow Q_2 \,|\, \Gamma\bar{Q}_2 \longrightarrow \cdots$$

$$\Gamma\bar{Q}_0 \longrightarrow Q_1 \,|\, \Gamma\bar{Q}_1 \longrightarrow Q_2 \,|\, \Gamma\bar{Q}_2 \longrightarrow \cdots$$

Grid search

Naive approach using AutoML for Q-learning

$$\Gamma\bar{Q}_0 \longrightarrow Q_1 \,|\, \Gamma\bar{Q}_1 \longrightarrow Q_2 \,|\, \Gamma\bar{Q}_2 \longrightarrow \cdots$$

$$\Gamma\bar{Q}_0 \longrightarrow Q_1 \,|\, \Gamma\bar{Q}_1 \longrightarrow Q_2 \,|\, \Gamma\bar{Q}_2 \longrightarrow \cdots$$

$$\Gamma\bar{Q}_0 \longrightarrow Q_1 \,|\, \Gamma\bar{Q}_1 \longrightarrow Q_2 \,|\, \Gamma\bar{Q}_2 \longrightarrow \cdots$$

Grid search

Ignores nonstationarities

Multiple trainings

Adaptive Q-Network (ours)

Only one training

Adaptive Q-Network

$\bar{Q}_0$

Target network

Adaptive Q-Network

$\Gamma \bar{Q}_0$

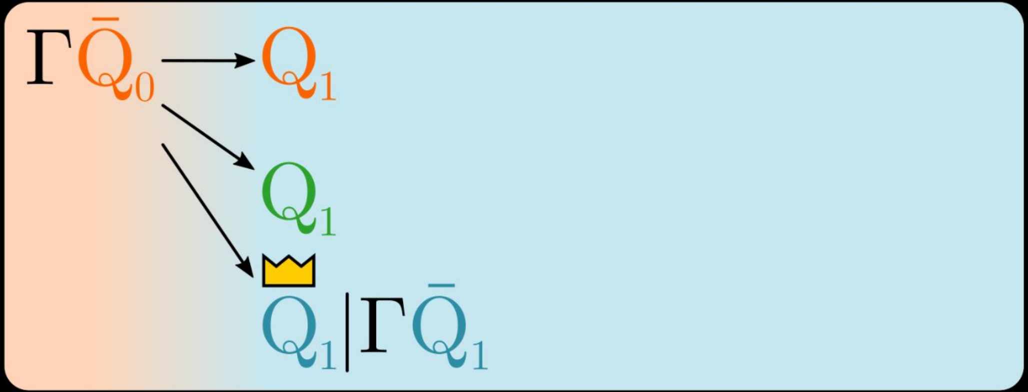
Target

Adaptive Q-Network

$$\begin{tikzcd} \Gamma\bar{Q}_0 \arrow[r] \arrow[dr] \arrow[ddr] & Q_1 \\ & Q_1 \\ & Q_1 \end{tikzcd}$$

Online networks
with different hyperparameters

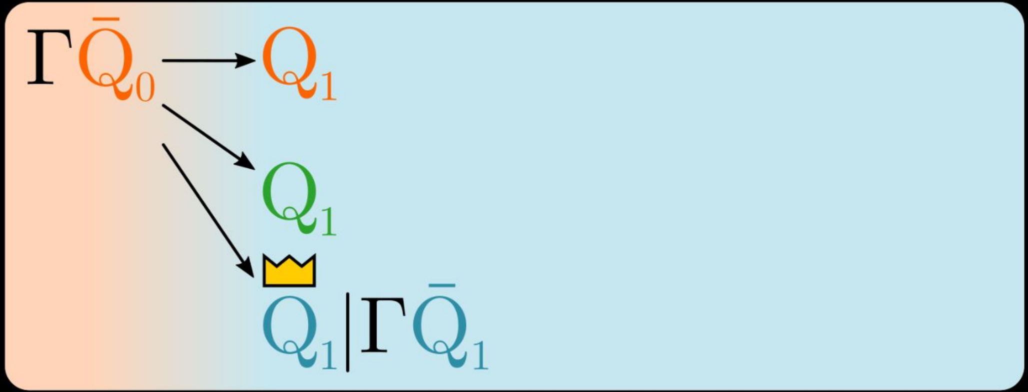
Adaptive Q-Network



$$\bar{\theta} \leftarrow \operatorname*{argmin}_{\theta^k,\ k\in\{1,\ \ldots,\ K\}} \text{TD-error}(\theta^k,\ \bar{\theta})$$

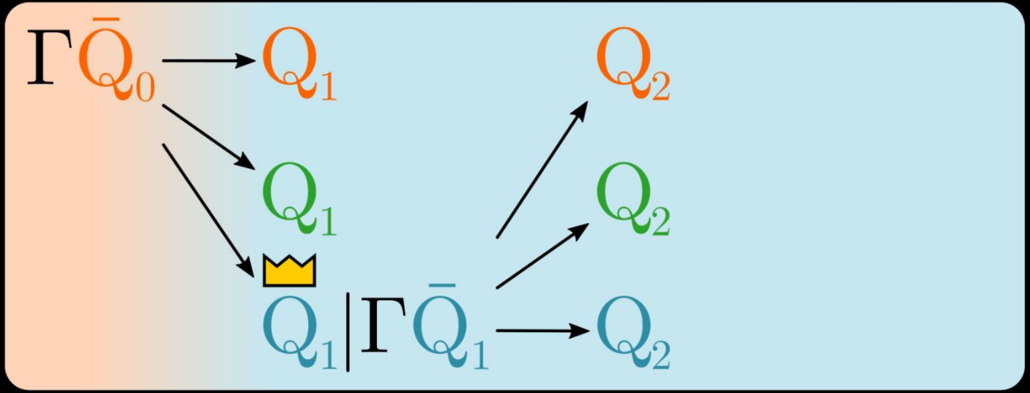
Target update

Adaptive Q-Network



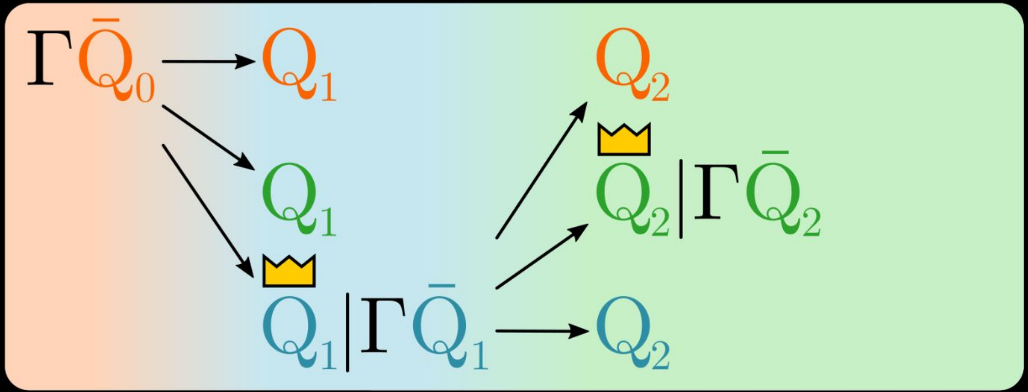
$$\bar{\theta} \leftarrow \underset{\theta^k,\ k\in\{1,\ ...,\ K\}}{\operatorname{argmin}}\ \text{TD-error}(\theta^k,\ \bar{\theta})$$

Adaptive Q-Network



Nonstationarity accounted for

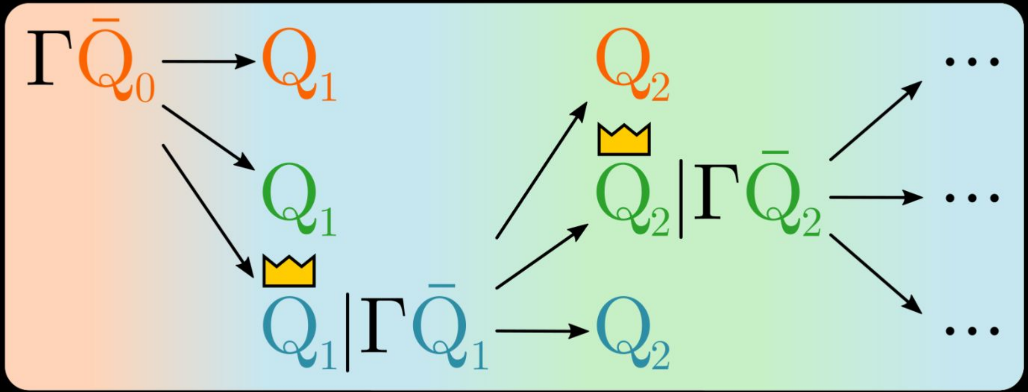
Adaptive Q-Network



$$\bar{\theta} \leftarrow \operatorname*{argmin}_{\theta^k,\ k \in \{1, \dots, K\}} \text{TD-error}(\theta^k, \bar{\theta})$$

Nonstationarity accounted for

Adaptive Q-Network



$$\bar{\theta} \leftarrow \underset{\theta^k,\ k\in\{1,\ \ldots,\ K\}}{\operatorname{argmin}} \text{TD-error}(\theta^k,\ \bar{\theta})$$

Experiments

Activation functions (18)

CELU
ELU
GELU
Hard Sigmoid
Hard SiLU
Hard tanh
Leaky ReLU
Log-Sigmoid
Log-Softmax
ReLU
SELU
Sigmoid
SiLU
Soft-sign
Softmax
Softplus
Standardize
Tanh

Activation functions (18)

CELU
ELU
GELU
Hard Sigmoid
Hard SiLU
Hard tanh
Leaky ReLU
Log-Sigmoid
Log-Softmax
ReLU
SELU
Sigmoid
SiLU
Soft-sign
Softmax
Softplus
Standardize
Tanh

Optimizers (24)

AdaBelief
AdaDelta
AdaGrad
AdaFactor
Adam
Adamax
AdamaxW
AdamW
AMSGrad
Fromage
Lamb
Lars
Lion
Nadam
NadamW
Noisy SGD
Novograd
Optimistic GD
RAdam
RMSProp
RProp
SGD
SM3
Yogi

Activation functions (18)

CELU
ELU
GELU
Hard Sigmoid
Hard SiLU
Hard tanh
Leaky ReLU
Log-Sigmoid
Log-Softmax
ReLU
SELU
Sigmoid
SiLU
Soft-sign
Softmax
Softplus
Standardize
Tanh

Optimizers (24)

AdaBelief
AdaDelta
AdaGrad
AdaFactor
Adam
Adamax
AdamaxW
AdamW
AMSGrad
Fromage
Lamb
Lars
Lion
Nadam
NadamW
Noisy SGD
Novograd
Optimistic GD
RAdam
RMSProp
RProp
SGD
SM3
Yogi

Losses (4)

Huber
L1
L2
Log cosh

Activation functions (18)

CELU
ELU
GELU
Hard Sigmoid
Hard SiLU
Hard tanh
Leaky ReLU
Log-Sigmoid
Log-Softmax
ReLU
SELU
Sigmoid
SiLU
Soft-sign
Softmax
Softplus
Standardize
Tanh

Optimizers (24)

AdaBelief
AdaDelta
AdaGrad
AdaFactor
Adam
Adamax
AdamaxW
AdamW
AMSGrad
Fromage
Lamb
Lars
Lion
Nadam
NadamW
Noisy SGD
Novograd
Optimistic GD
RAdam
RMSProp
RProp
SGD
SM3
Yogi

Losses (4)

Huber
L1
L2
Log cosh

Learning rates

From
0.0000001
to
0.001

Number of CNN layers

From
1
to
3

Number of CNN channels

From
16
to
64

Kernel size of CNN layers

From
2
to
8

Stride of CNN layers

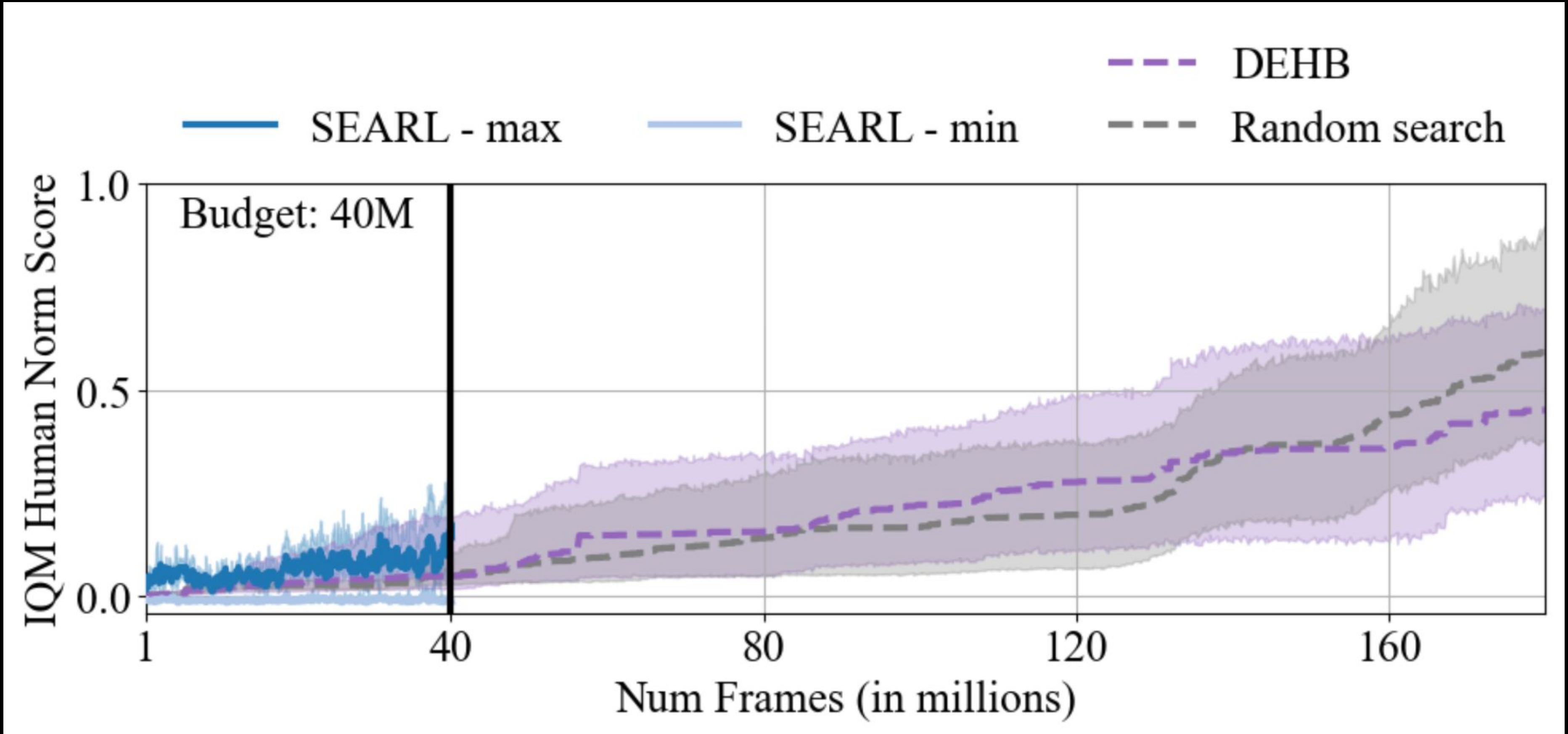
From
2
to
5

Number of hidden MLP layers

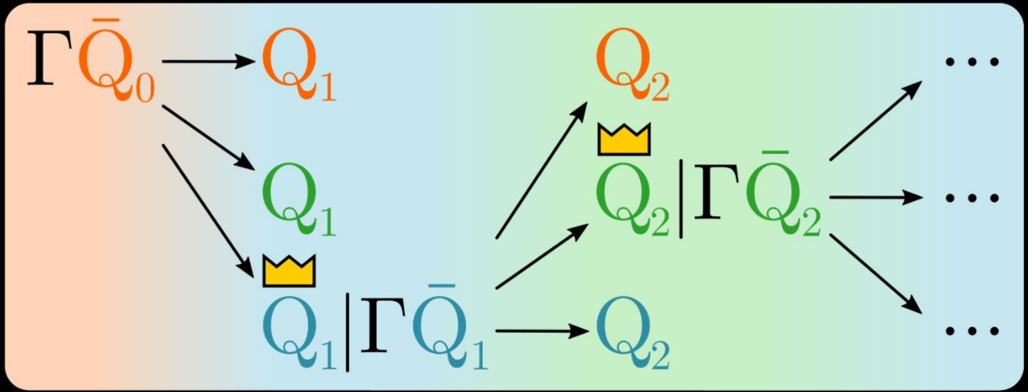
From
0
to
2

Number of MLP neurons

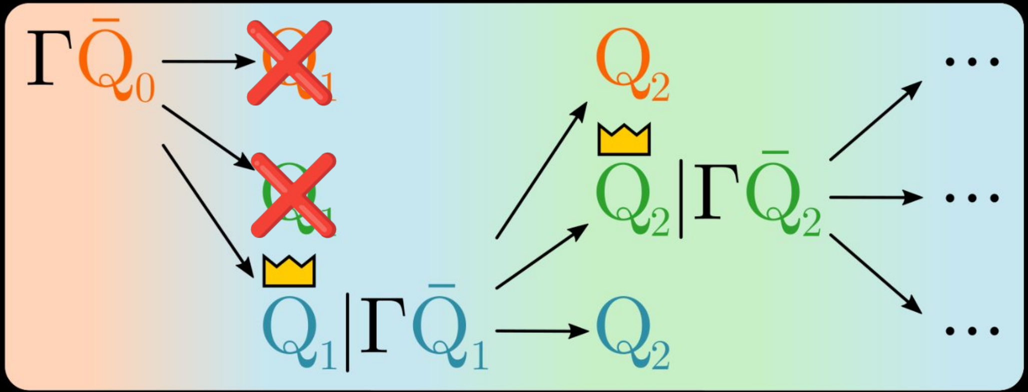
From
25
to
512


DEHB
SEARL - max
SEARL - min
Random search
Budget: 40M
IQM Human Norm Score
1.0
0.5
0.0
1
40
80
120
160
Num Frames (in millions)

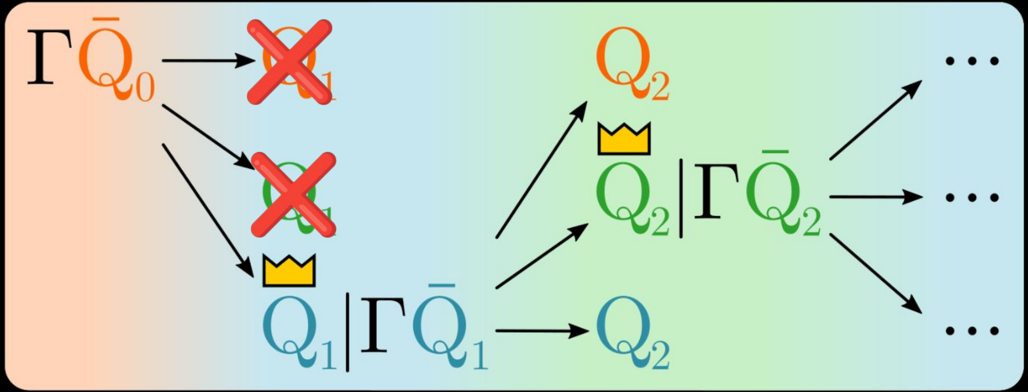
Adaptive Q-Network

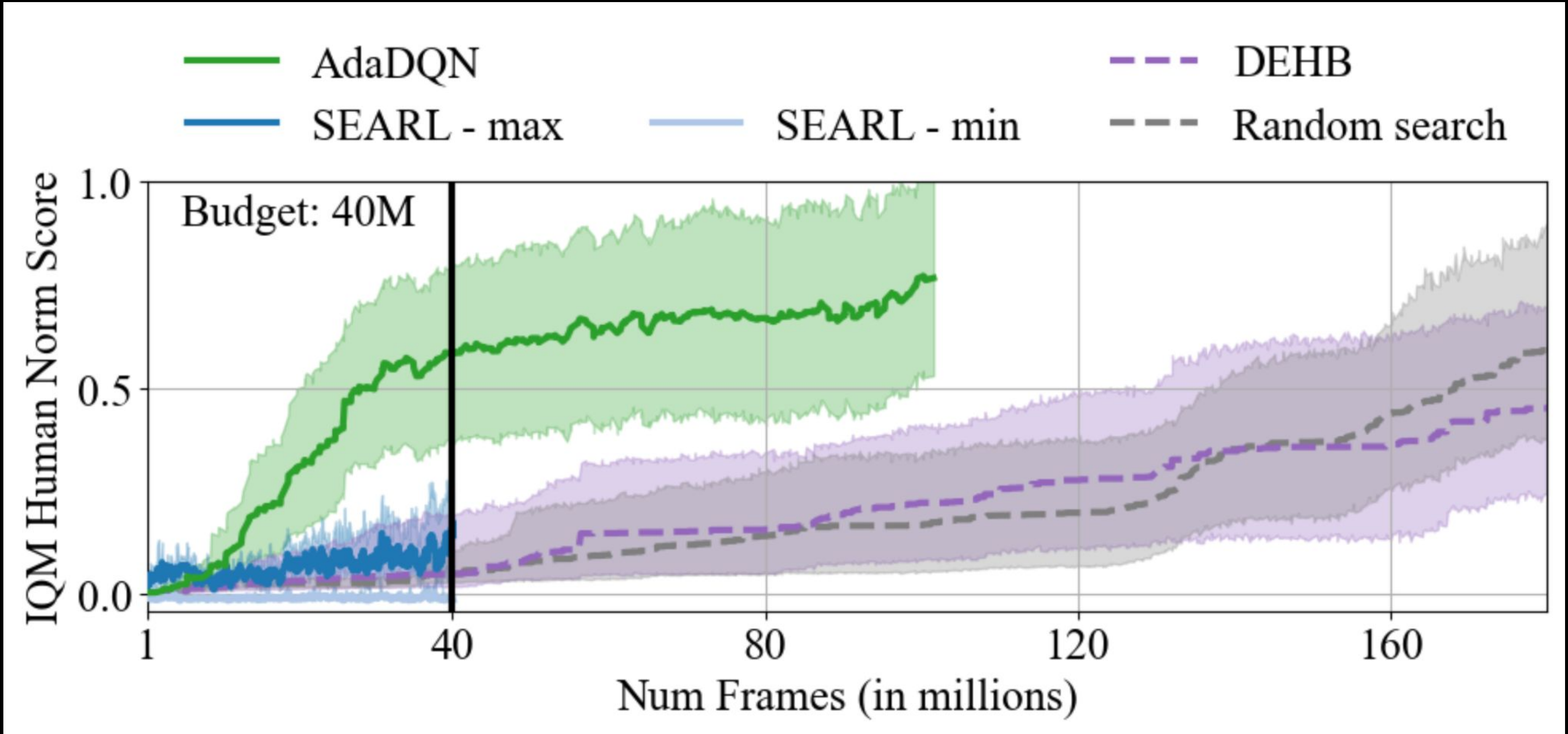


Adaptive Q-Network



Adaptive Q-Network




AdaDQN
SEARL - max
SEARL - min
DEHB
Random search
IQM Human Norm Score
Budget: 40M
1.0
0.5
0.0
1
40
80
120
160
Num Frames (in millions)

Adaptive Q-Network

Théo Vincent

Fabian Wahren

Jan Peters

Boris Belousov

Carlo D'Eramo

An AutoRL algorithm tailored for RL

AdaQN **accounts** for RL nonstationarities

AdaQN is effective in **large** hyperparameter spaces