Tensorforce: building an applied reinforcement learning framework using TensorFlow

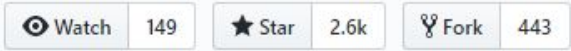
Alexander Kuhnle

28th January 2020

Content

- Motivation
- Key features
- TensorFlow as implementation platform
- User feedback
- Applications

About the framework

- Originally developed by Michael Schaarschmidt, Kai Fricke and myself
- Introduction blog post: 11/07/2017
- Since mid-2018 developed by myself
- GitHub: https://github.com/tensorforce/tensorforce
- ~200 pull requests by ~50 contributors
- 

Why build yet another reinforcement learning library?

Existing frameworks

Example: OpenAI Baselines

openai / baselines

Watch 505 | Star 9.1k | Fork 3.1k

Code | Issues 334 | Pull requests 57 | Projects 0 | Security | Insights

Branch: master — baselines / baselines / — Create new file | Find file | History

pzhokhov Fix build with latest gym (#1034) ... — Latest commit 9ee399f on Nov 10, 2019

..		
a2c	Report episode rewards/length in A2C and ACKTR (#856)	10 months ago
acer	fix a bug in acer saving and loading model (#990)	4 months ago
acktr	release Internal changes (#895)	9 months ago
bench	run test_monitor through pytest: fix the test, add flake8 to bench di...	8 months ago
common	Fix build with latest gym (#1034)	2 months ago
ddpg	Remove duplicate code in adaptive param noise. (#976)	6 months ago
deepq	Fix typo (#930)	6 months ago
gail	Fix behavior cloning due to API changes (#1014)	3 months ago
her	Remove duplicate code in adaptive param noise. (#976)	6 months ago
ppo1	fix shuffling bug in ppo1	10 months ago
ppo2	release Internal changes (#895)	9 months ago
trpo_mpi	mpi-less baselines (#689)	last year
__init__.py	Initial commit	3 years ago
logger.py	add log_path flag to command line utility (#917)	8 months ago
results_plotter.py	Fix ppo2 with MPI bug, other minor fixes (#735)	last year
run.py	Fix RuntimeError (#910) (#1015)	3 months ago

Branch: master — baselines / baselines / deepq / — Create new file | Find file | History

timokau and pzhokhov Fix typo (#930) ... — Latest commit 7cadef7 on Aug 6, 2019

..		
experiments	Interface for U.make_session changed (#865)	10 months ago
README.md	deduplicate algorithms in rl-algs and baselines (#18)	last year
__init__.py	Update __init__.py	last year
build_graph.py	Fix typo (#930)	6 months ago
deepq.py	Fix typo (#930)	6 months ago
defaults.py	refactor a2c, acer, acktr, ppo2, deepq, and trpo_mpi (#490)	last year
models.py	Remove model def from deepq. (#946)	7 months ago
replay_buffer.py	prioritized experience replay bug (#527)	last year
utils.py	fix the definition of `TfInput.make_feed_dict`. (#812)	10 months ago

Branch: master — baselines / baselines / ppo2 / — Create new file | Upload files | Find file | History

pzhokhov release Internal changes (#895) ... — Latest commit 9b68103 on May 8, 2019

..		
README.md	update readmes (#514)	last year
__init__.py	Add ACER, PPO2, and results_plotter.py	2 years ago
defaults.py	release Internal changes (#800)	11 months ago
microbatched_model.py	release Internal changes (#895)	9 months ago
model.py	release Internal changes (#895)	9 months ago
ppo2.py	release Internal changes (#895)	9 months ago
runner.py	microbatches in ppo2, custom frame size in WarpFrame, matching fc lay...	last year
test_microbatches.py	raised the tolerance on the test_microbatches test	11 months ago

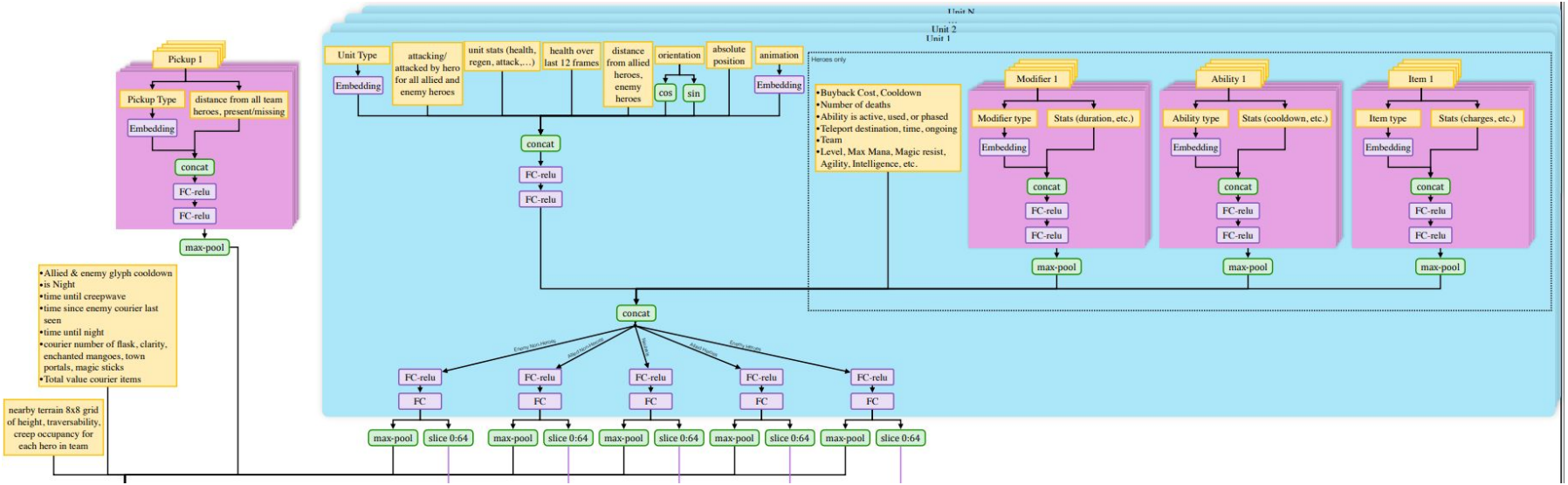
Largely independent agent implementations

Research frameworks vs practical requirements

- “Standardized” state/action space: single float-array state, int/float action
- States/action space with multiple components, various types and shapes

Research frameworks vs practical requirements

- “Standardized” state/action space: single float-array state, int/float action
- States/action space with multiple components, various types and shapes



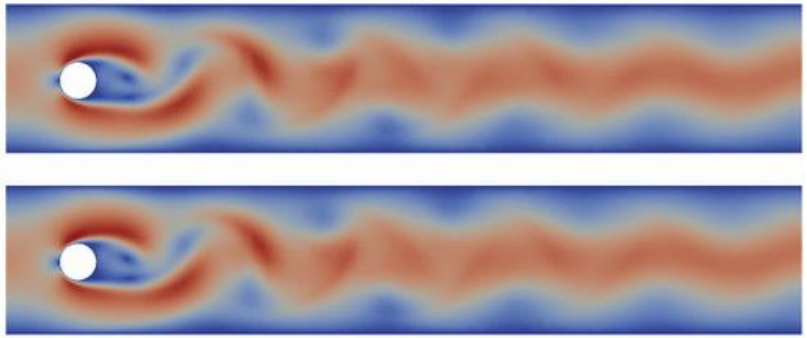
Research frameworks vs practical requirements

- “Standardized” state/action space: single float-array state, int/float action
- Interaction in training episodes with terminal/goal state

- States/action space with multiple components, various types and shapes
- Continuous interaction, no “natural” termination of interaction

Research frameworks vs practical requirements

- “Standardized” state/action space: single float-array state, int/float action
- Interaction in training episodes with terminal/goal state
- States/action space with multiple components, various types and shapes
- Continuous interaction, no “natural” termination of interaction



Research frameworks vs practical requirements

- “Standardized” state/action space: single float-array state, int/float action
- Interaction in training episodes with terminal/goal state
- Agent reference implementations, may include environment-specific details
- States/action space with multiple components, various types and shapes
- Continuous interaction, no “natural” termination of interaction
- (Re-)combination of techniques to suit characteristics of application

Research frameworks vs practical requirements

- “Standardized” state/action space: single float-array state, int/float action
- Interaction in training episodes with terminal/goal state
- Agent reference implementations, may include environment-specific details
- Mix of Python and TensorFlow(/PyTorch)

- States/action space with multiple components, various types and shapes
- Continuous interaction, no “natural” termination of interaction
- (Re-)combination of techniques to suit characteristics of application
- Single implementation platform

Tensorforce: *“TF Estimators for RL”*

tensorforce / **tensorforce**

Watch 149 | Star 2.6k | Fork 441

Code | Issues 4 | Pull requests 0 | Projects 0 | Security | Insights

Branch: master — tensorforce / tensorforce / **core** / — Create new file | Find file | History

AlexKuhnle Parallelizable remote environments, merged ParallelRunner into Runner... — Latest commit 5da11b7 3 days ago

..		
distributions	auto memory capacity	4 days ago
estimators	added tf assert exception messages	18 days ago
layers	added reshape layer, added main-level imports, improved gym and envir...	9 days ago
memories	added tf assert exception messages	18 days ago
models	Parallelizable remote environments, merged ParallelRunner into Runner...	3 days ago
networks	finished upgrade to TF2, improved summary handling, other internal ch...	2 months ago
objectives	version upgrade	27 days ago
optimizers	module specification fix	17 days ago
parameters	continued upgrade to TF2 API	2 months ago
policies	finished upgrade to TF2, improved summary handling, other internal ch...	2 months ago
utils	Support for JSON-encoding specific numpy types	6 days ago
__init__.py	initial commit for final major revision version	8 months ago
module.py	module specification fix	17 days ago

Modular component-based library design

Branch: master — tensorforce / tensorforce / core / **models** / — Create new file | Find file | History

AlexKuhnle Parallelizable remote environments, merged ParallelRunner into Runner... — Latest commit 5da11b7 3 days ago

..		
__init__.py	saving includes agent spec, Agent.load function, PolicyAgent/Model mo...	4 months ago
constant.py	upgrade to TF2.0	2 months ago
model.py	Parallelizable remote environments, merged ParallelRunner into Runner...	3 days ago
random.py	upgrade to TF2.0	2 months ago
tensorforce.py	Parallelizable remote environments, merged ParallelRunner into Runner...	3 days ago

Branch: master — tensorforce / tensorforce / **agents** / — Create new file | Find file | History

AlexKuhnle Parallelizable remote environments, merged ParallelRunner into Runner... — Latest commit 5da11b7 3 days ago

..		
__init__.py	improved Environment.create, introduced wrapper env for max-steps, an...	2 months ago
a2c.py	removed conv transpose layers, improved create functions, changed som...	10 days ago
ac.py	removed conv transpose layers, improved create functions, changed som...	10 days ago
agent.py	Parallelizable remote environments, merged ParallelRunner into Runner...	3 days ago
constant.py	version upgrade	27 days ago
dpg.py	removed conv transpose layers, improved create functions, changed som...	10 days ago
dqn.py	removed conv transpose layers, improved create functions, changed som...	10 days ago
dueling_dqn.py	removed conv transpose layers, improved create functions, changed som...	10 days ago
ppo.py	removed conv transpose layers, improved create functions, changed som...	10 days ago
random.py	version upgrade	27 days ago
tensorforce.py	auto memory capacity	4 days ago
trpo.py	removed conv transpose layers, improved create functions, changed som...	10 days ago
vpg.py	removed conv transpose layers, improved create functions, changed som...	10 days ago

Tensorforce: *“TF Estimators for RL”*

tensorforce / tensorforce

Watch 149 | Star 2.6k | Fork 441

Code | Issues 4 | Pull requests 0 | Projects 0 | Security | Insights

Branch: master | tensorforce / tensorforce / core / | Create new file | Find file | History

AlexKuhnle Parallelizable remote environments, merged ParallelRunner into Runner... | Latest commit 5da11b7 3 days ago

..		
distributions	auto memory capacity	4 days ago
estimators	added tf assert exception messages	18 days ago
layers	added reshape layer, added main-level imports, improved gym and envir...	9 days ago
memories	added tf assert exception messages	18 days ago
models	Parallelizable remote environments, merged ParallelRunner into Runner...	3 days ago
networks	finished upgrade to TF2, improved summary handling, other internal ch...	2 months ago
objectives	version upgrade	27 days ago
optimizers	module specification fix	17 days ago
parameters	continued upgrade to TF2 API	2 months ago
policies	finished upgrade to TF2, improved summary handling, other internal ch...	2 months ago
utils	Support for JSON-encoding specific numpy types	6 days ago
__init__.py	initial commit for final major revision version	8 months ago
module.py	module specification fix	17 days ago

Modular component-based library design

Branch: master | tensorforce / tensorforce / core / models / | Create new file | Find file | History

AlexKuhnle Parallelizable remote environments, merged ParallelRunner into Runner... | Latest commit 5da11b7 3 days ago

..		
__init__.py	saving includes agent spec, Agent.load function, PolicyAgent/Model mo...	4 months ago
constant.py	upgrade to TF2.0	2 months ago
model.py	Parallelizable remote environments, merged ParallelRunner into Runner...	3 days ago
random.py	upgrade to TF2.0	2 months ago
tensorforce.py	Parallelizable remote environments, merged ParallelRunner into Runner...	3 days ago

Branch: master | tensorforce / tensorforce / agents / | Create new file | Find file | History

AlexKuhnle Parallelizable remote environments, merged ParallelRunner into Runner... | Latest commit 5da11b7 3 days ago

..		
__init__.py	improved Environment.create, introduced wrapper env for max-steps, an...	2 months ago
[illegible]	[illegible]	10 days ago
ac.py	removed conv transpose layers, improved create functions, changed som...	10 days ago
[illegible]	[illegible]	3 days ago
constant.py	version upgrade	27 days ago
dpg.py	removed conv transpose layers, improved create functions, changed som...	10 days ago
dqn.py	removed conv transpose layers, improved create functions, changed som...	10 days ago
dueling_dqn.py	removed conv transpose layers, improved create functions, changed som...	10 days ago
ppo.py	removed conv transpose layers, improved create functions, changed som...	10 days ago
random.py	version upgrade	27 days ago
tensorforce.py	auto memory capacity	4 days ago
trpo.py	removed conv transpose layers, improved create functions, changed som...	10 days ago
vpg.py	removed conv transpose layers, improved create functions, changed som...	10 days ago

No fundamental differences internally,
all a matter of modular configuration!

Tensorforce: *“TF Estimators for RL”*

Usage example: DQN agent (configured manually, for illustration)

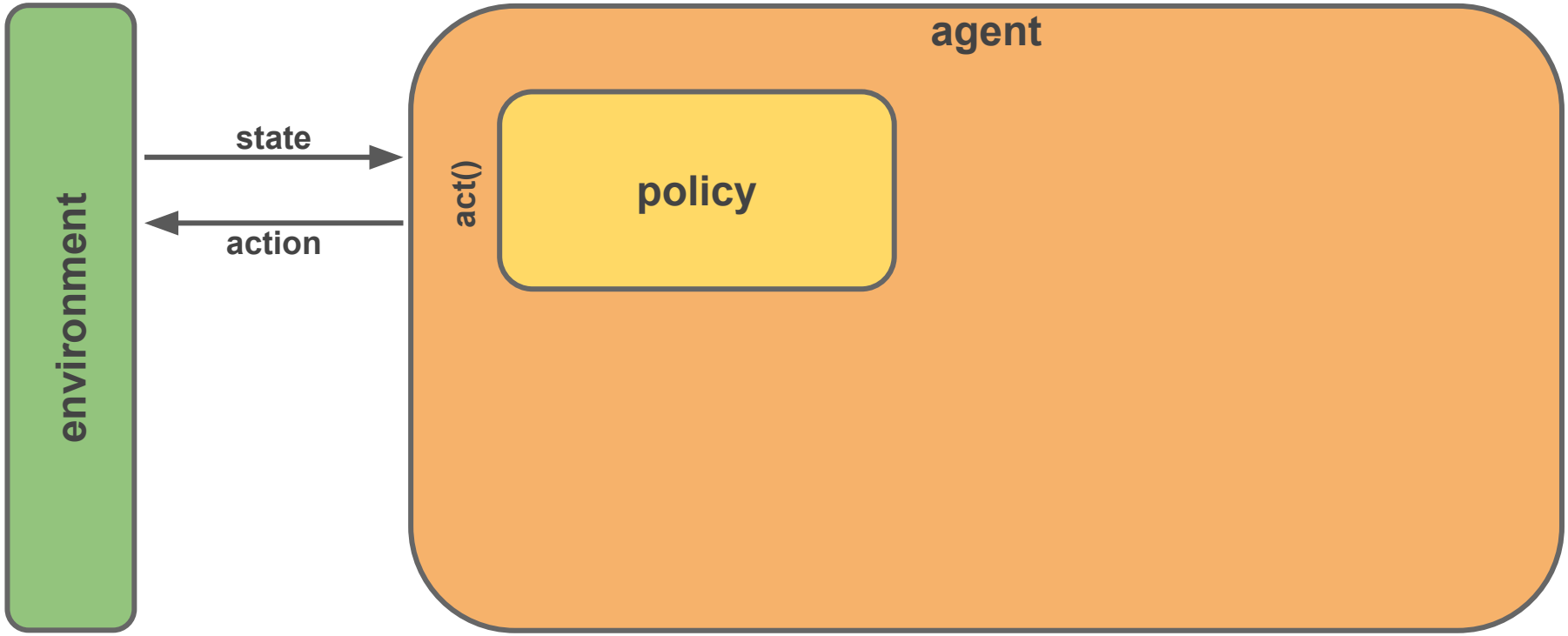
```
agent = Agent.create(
    policy=dict(network='auto', temperature=0.0),
    memory=dict(type='replay', capacity=100000),
    update=dict(unit='timesteps', batch_size=64, frequency=8),
    optimizer=dict(type='adam', learning_rate=3e-4),
    objective=dict(type='value', value='action', huber_loss=0.0),
    reward_estimation=dict(horizon=0, discount=0.99, estimate_horizon='late'),
    baseline_policy=dict(network='auto', temperature=0.0),
    baseline_optimizer=dict(type='synchronization', update_weight=0.2)
)

states = environment.reset()
actions = agent.act(states=states)
states, terminal, reward = environment.execute(actions=actions)
agent.observe(terminal=terminal, reward=reward)
```

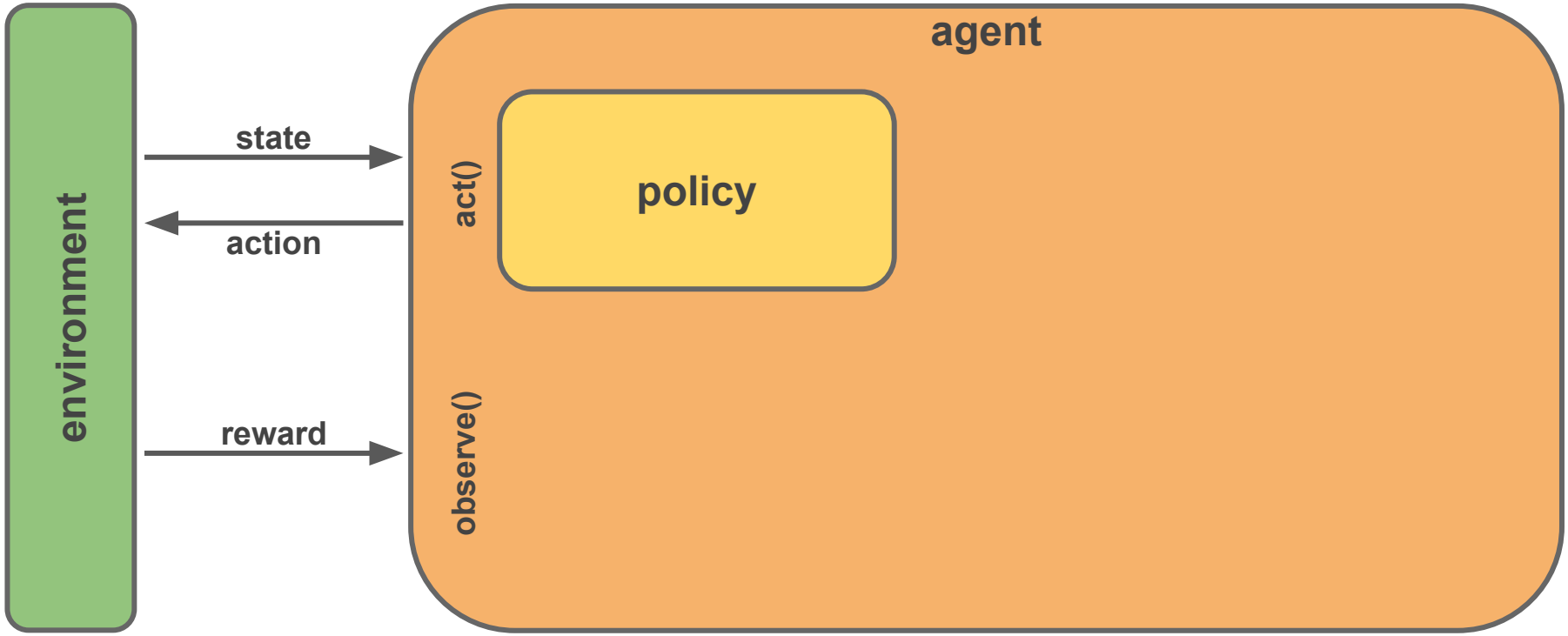
Key features of Tensorforce

Reinforcement learning architecture

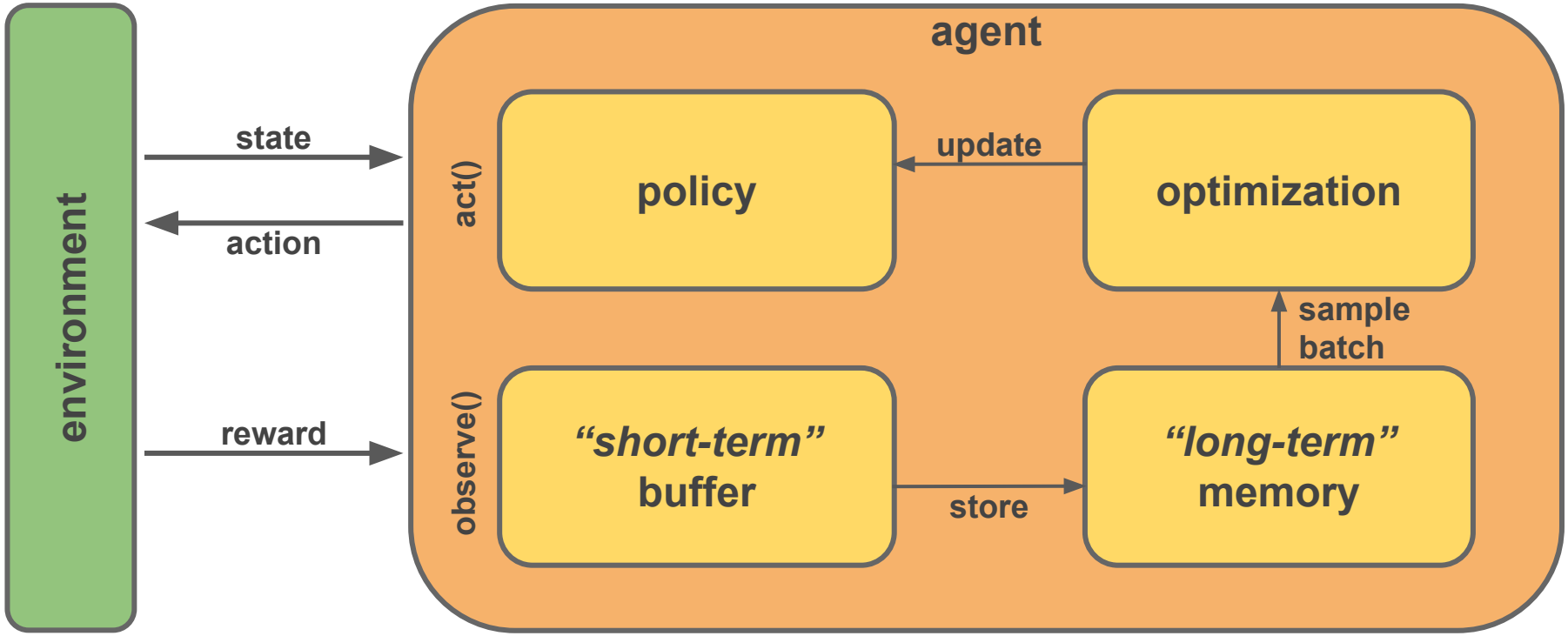
Reinforcement learning architecture



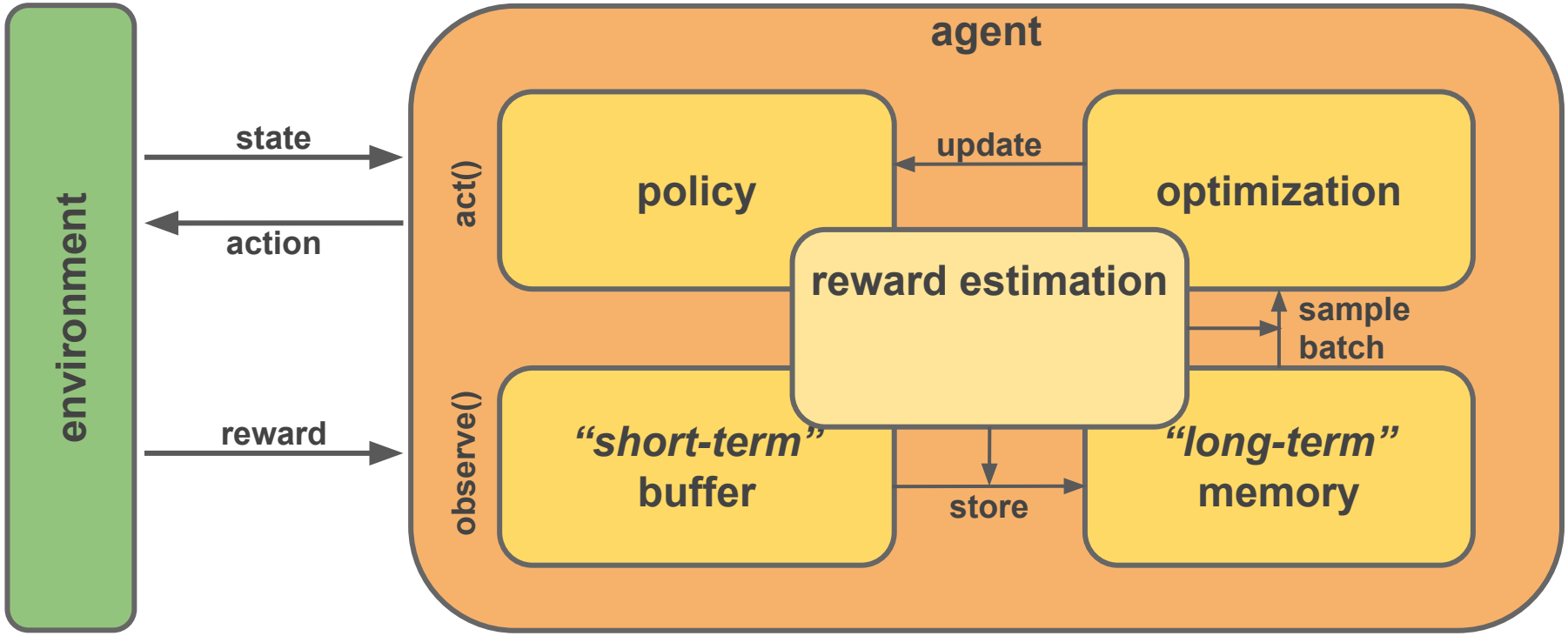
Reinforcement learning architecture



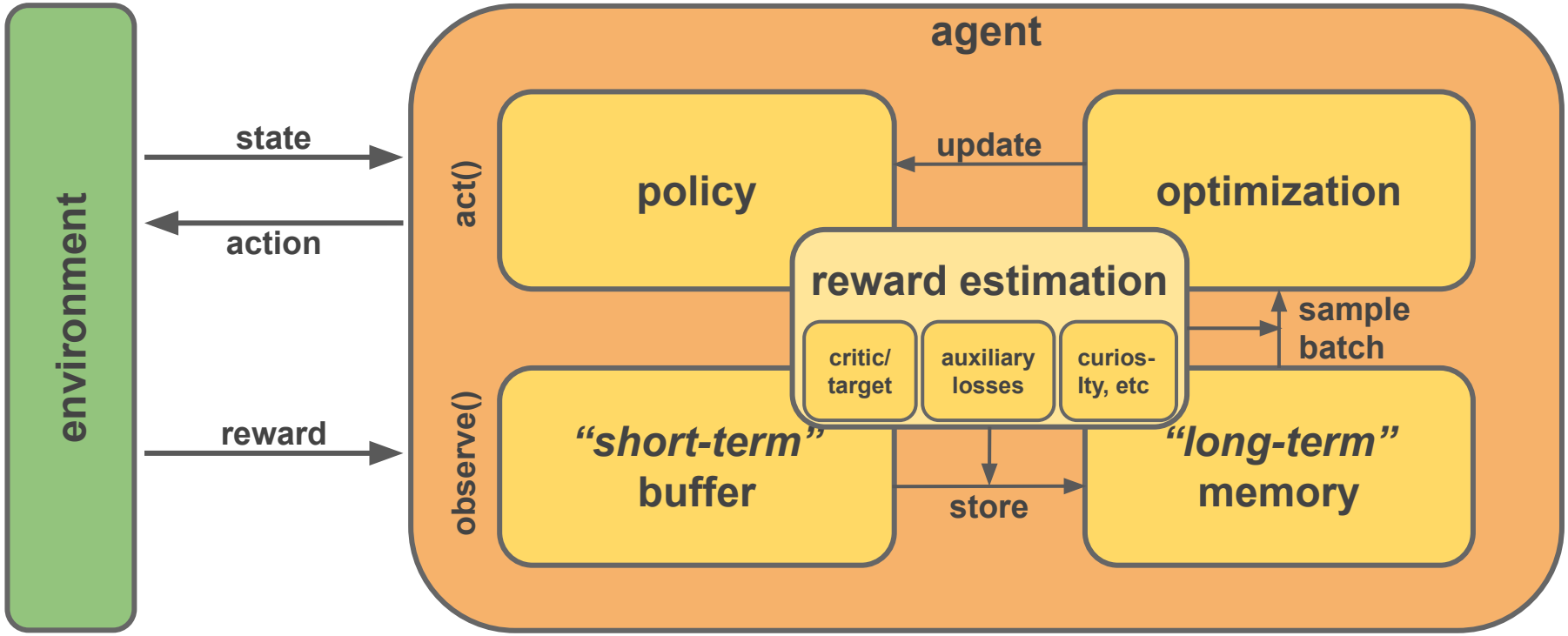
Reinforcement learning architecture



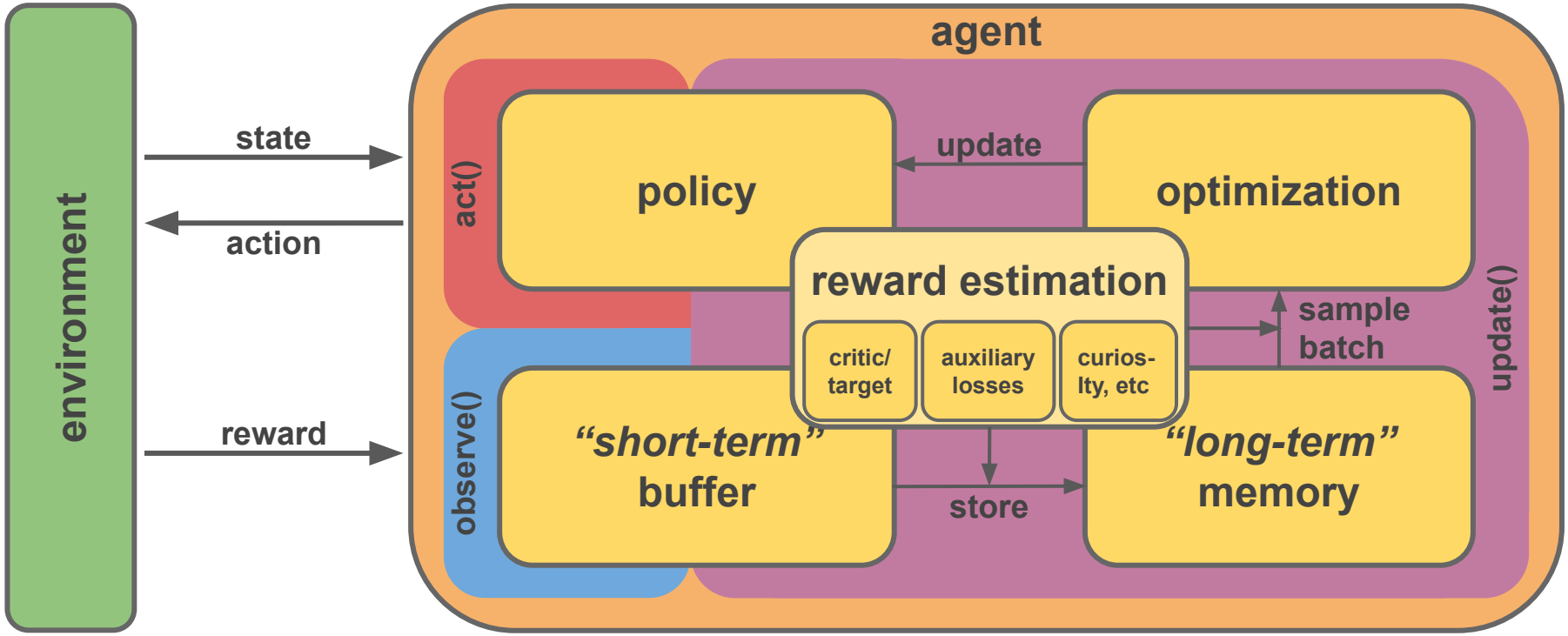
Reinforcement learning architecture



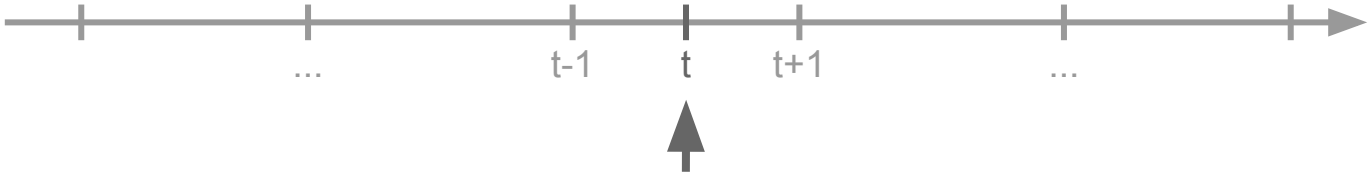
Reinforcement learning architecture



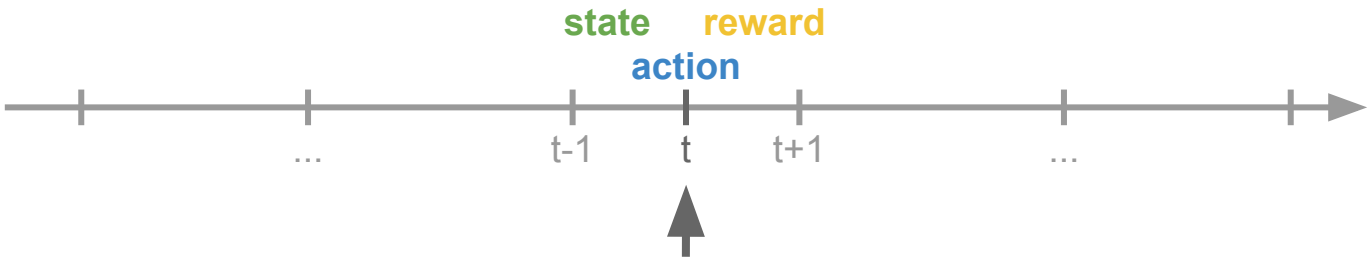
Reinforcement learning architecture



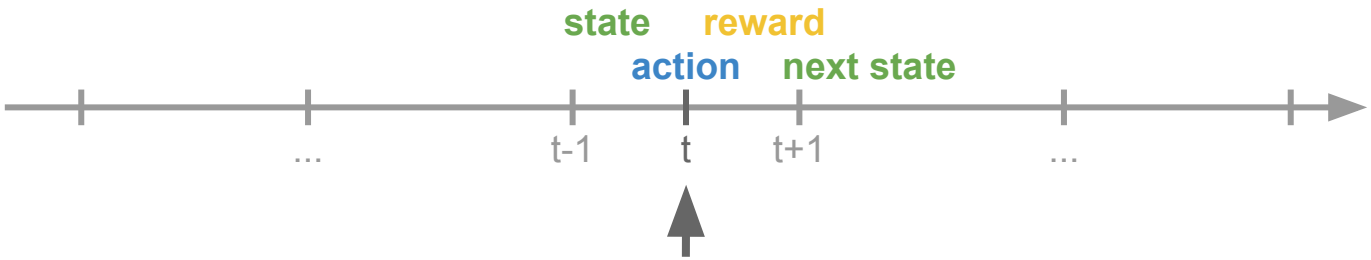
RL timestep and dependencies



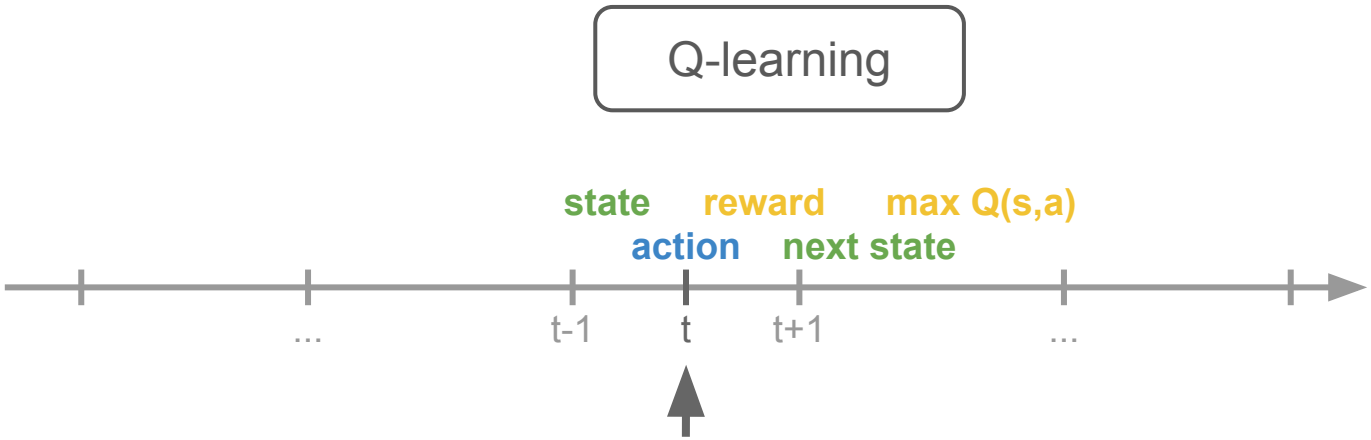
RL timestep and dependencies



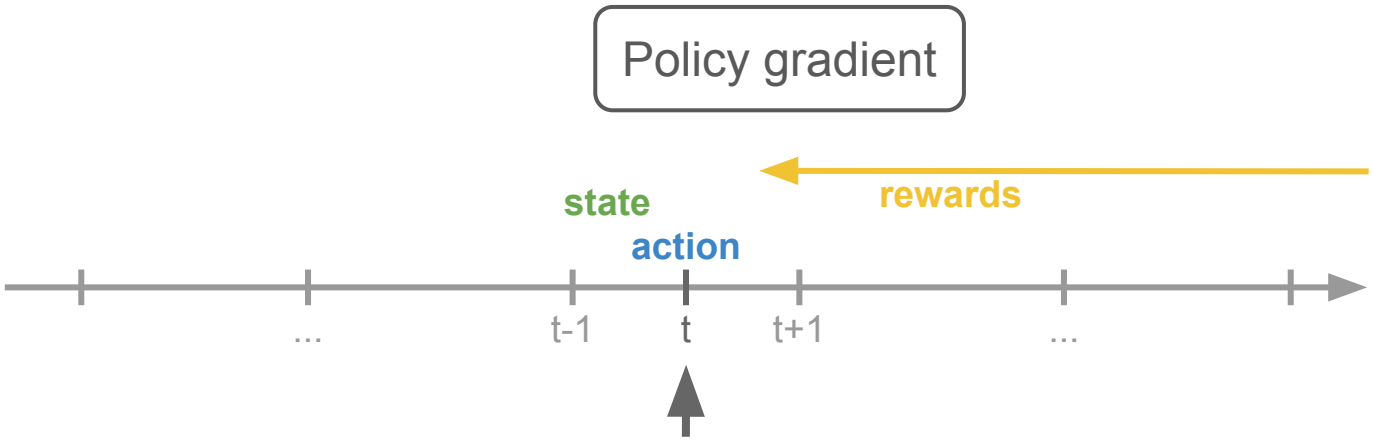
RL timestep and dependencies



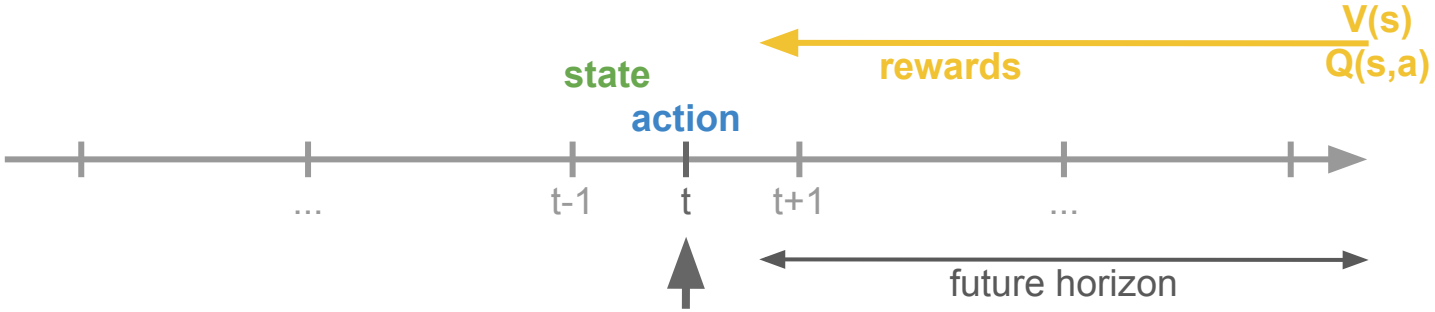
RL timestep and dependencies



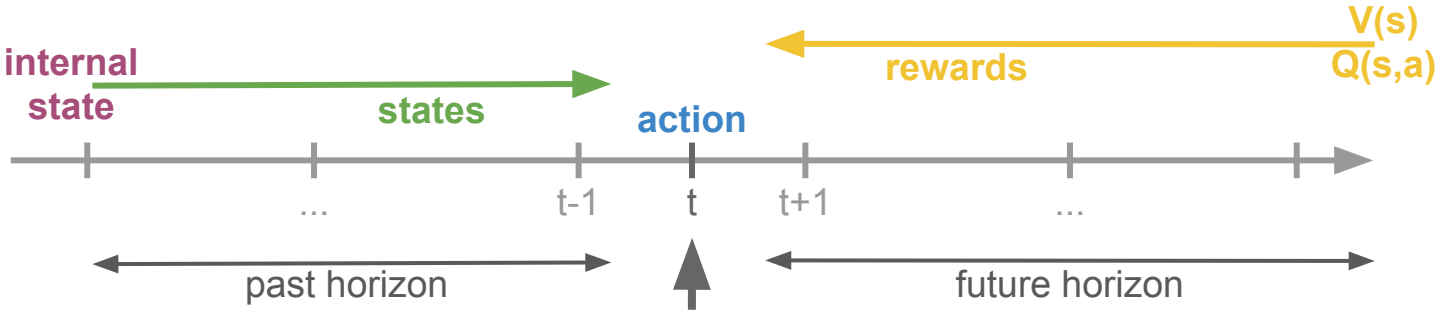
RL timestep and dependencies



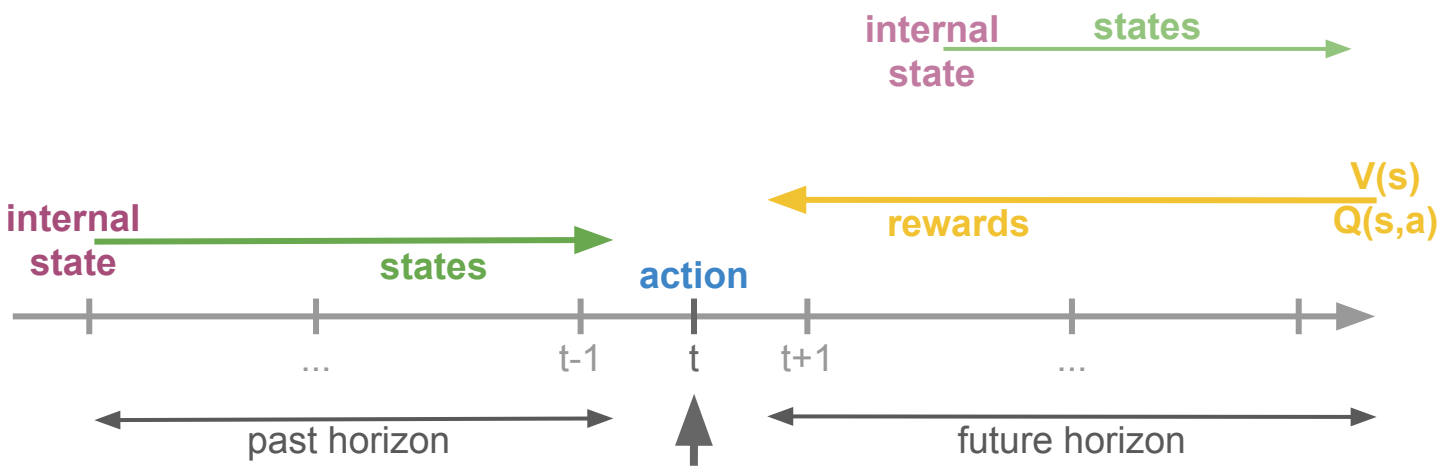
RL timestep and dependencies



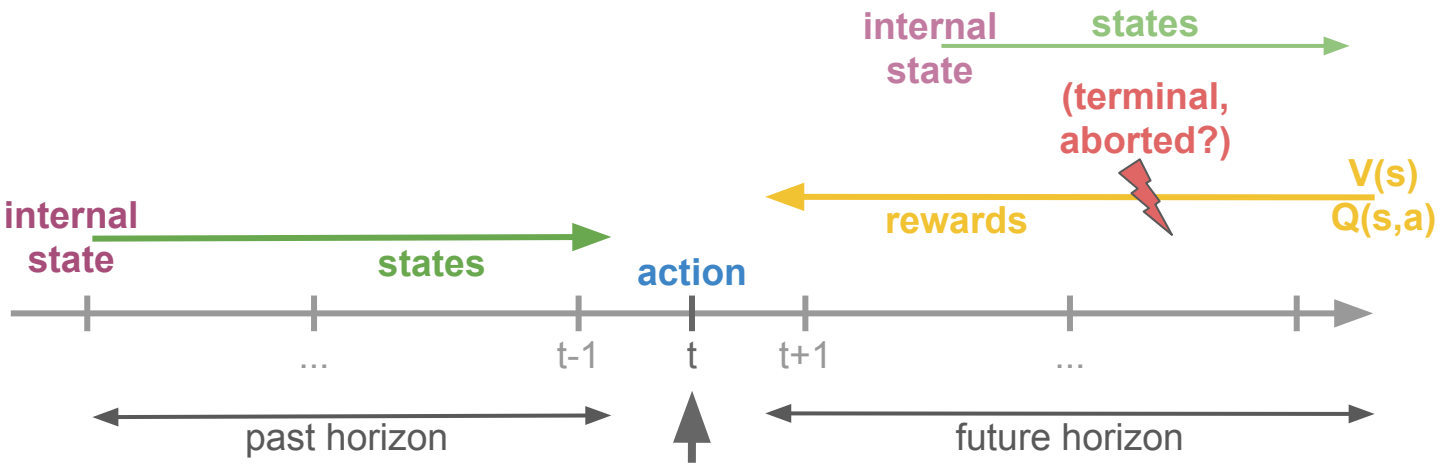
RL timestep and dependencies



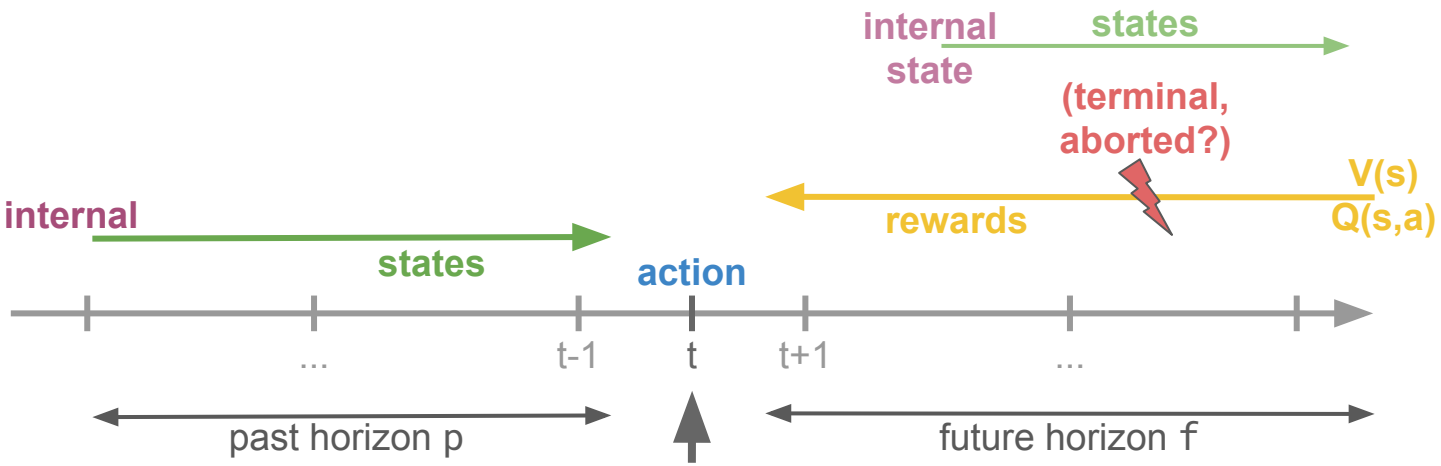
RL timestep and dependencies



RL timestep and dependencies



RL timestep and dependencies



Actual batch instance consists of:

`internals[t-p]`

`actions[t]`

`internals[t+f-p-1]`

`states[t+f-p:t+f]`

`states[t-p+1:t]`

`rewards[t:t+f]`

`(actions[t+f])`

Other framework features

Optimizers as graph assemblers:

- TensorFlow 1.X: based on loss-tensor
- Keras/PyTorch: based on loss-function

Other framework features

Optimizers as graph assemblers:

- TensorFlow/Keras/PyTorch: based on loss-tensor/-function
- Tensorforce
  - generic “updaters” with a range of potential inputs: loss, KL-divergence, source-vars, etc
  - update modifiers: multi-step, update clipping, batch subsampling

Other framework features

Optimizers as graph assemblers:

- TensorFlow/Keras/PyTorch: based on loss-tensor/-function
- Tensorforce
  - generic “updaters” with a range of potential inputs: loss, KL-divergence, source-vars, etc
  - update modifiers: multi-step, update clipping, batch subsampling

Static vs dynamic hyperparameters:

- TensorFlow/PyTorch: seemingly only learning-rate

```
tf.keras.optimizers.schedules.LearningRateSchedule
torch.optim.lr_scheduler.*
```

Other framework features

Optimizers as graph assemblers:

- TensorFlow/Keras/PyTorch: based on loss-tensor/-function
- Tensorforce
  - generic “updaters” with a range of potential inputs: loss, KL-divergence, source-vars, etc
  - update modifiers: multi-step, update clipping, batch subsampling

Static vs dynamic hyperparameters:

- TensorFlow/PyTorch: seemingly only learning-rate
- Tensorforce:
  - All dynamic parameters are of type `Parameter`: constant, decaying, piecewise, etc
  - Parameters scheduled based on timestep/episode/update,... (loss?)
  - Placeholder-with-default for straightforward experimentation

TensorFlow as an implementation platform

TensorFlow as an implementation platform

- Static graph compilation great for verification and TF/Python separation

TensorFlow as an implementation platform

- Static graph compilation great for verification and TF/Python separation
- However, problems have persisted with respect to:
  - Nesting `while` and `cond` in combination with gradients and TensorBoard summaries
  - Recently, almost every TF upgrade breaks one thing and/or fixes another

TensorFlow as an implementation platform

- Static graph compilation great for verification and TF/Python separation
- However, problems have persisted with respect to:
  - Nesting `while` and `cond` in combination with gradients and TensorBoard summaries
  - Recently, almost every TF upgrade breaks one thing and/or fixes another
- TensorFlow 2.0: Exceptions seem harder to interpret

```
    raise value
  File "/home/[illegible]/tensorforce-env/lib/python3.6/site-packages/tensorflow_core/python/training/monitored_session.py", line 1345, in run
    return self._sess.run(*args, **kwargs)
  File "/home/[illegible]/tensorforce-env/lib/python3.6/site-packages/tensorflow_core/python/training/monitored_session.py", line 1418, in run
    run_metadata=run_metadata)
  File "/home/[illegible]/tensorforce-env/lib/python3.6/site-packages/tensorflow_core/python/training/monitored_session.py", line 1176, in run
    return self._sess.run(*args, **kwargs)
  File "/home/[illegible]/tensorforce-env/lib/python3.6/site-packages/tensorflow_core/python/client/session.py", line 956, in run
    run_metadata_ptr)
  File "/home/[illegible]/tensorforce-env/lib/python3.6/site-packages/tensorflow_core/python/client/session.py", line 1180, in _run
    feed_dict_tensor, options, run_metadata)
  File "/home/[illegible]/tensorforce-env/lib/python3.6/site-packages/tensorflow_core/python/client/session.py", line 1359, in _do_run
    run_metadata)
  File "/home/[illegible]/tensorforce-env/lib/python3.6/site-packages/tensorflow_core/python/client/session.py", line 1384, in _do_call
    raise type(e)(node_def, op, message)
tensorflow.python.framework.errors_impl.InvalidArgumentError: indices[30] = 84 is not in [0, 84)
         [[{{node GatherV2}}]]
```

TensorFlow as an implementation platform

- Static graph compilation great for verification and TF/Python separation
- However, problems have persisted with respect to:
  - Nesting `while` and `cond` in combination with gradients and TensorBoard summaries
  - Recently, almost every TF upgrade breaks one thing and/or fixes another
- TensorFlow 2.0: Exceptions are harder to interpret
- TensorFlow 2.1: Version upgrades still change/break basic things

```
  File "/home/[illegible]/tensorforce-env/lib/python3.6/site-packages/tensorflow_core/python/ops/gradients_util.py", line 336, in _MaybeCompile
    return grad_fn()  # Exit early
  File "/home/[illegible]/tensorforce-env/lib/python3.6/site-packages/tensorflow_core/python/ops/gradients_util.py", line 669, in <lambda>
    lambda: grad_fn(op, *out_grads))
  File "/home/[illegible]/tensorforce-env/lib/python3.6/site-packages/tensorflow_core/python/ops/cond_v2.py", line 183, in _IfGrad
    building_gradient=True,
  File "/home/[illegible]/tensorforce-env/lib/python3.6/site-packages/tensorflow_core/python/ops/cond_v2.py", line 219, in _build_cond
    _make_indexed_slices_indices_types_match(_COND, [true_graph, false_graph])
  File "/home/[illegible]tensorforce-env/lib/python3.6/site-packages/tensorflow_core/python/ops/cond_v2.py", line 652, in _make_indexed_slices_ind
    (current_index, len(branch_graphs[0].outputs)))
ValueError: Insufficient elements in branch_graphs[0].outputs.
Expected: 38
Actual: 30
```

User feedback

Reasons for choosing Tensorforce

- No code digging: easy to get started and obtain results
- Modular structure: clean API and extensive configurability
- Full-on TensorFlow: computation graph can be extracted
- Focus on “core RL” performance, in particular reward estimation

Reasons for choosing Tensorforce

- No code digging: easy to get started and obtain results
- Modular structure: clean API and extensive configurability
- Full-on TensorFlow: computation graph can be extracted
- Focus on “core RL” performance, in particular reward estimation

Limitations and areas for development

- No code digging: hard to modify/extend beyond what’s supported
- Modular structure: no single script, no SOTA reference implementations
- Full-on TensorFlow: incomprehensible exceptions
- No focus on sophisticated hardware management and distributed execution

Applications

DeepCrawl: DRL-controlled game AI

(Alessandro Sestini, Università degli Studi di Firenze)

DeepCrawl: DRL-controlled game AI

(Alessandro Sestini, Università degli Studi di Firenze)

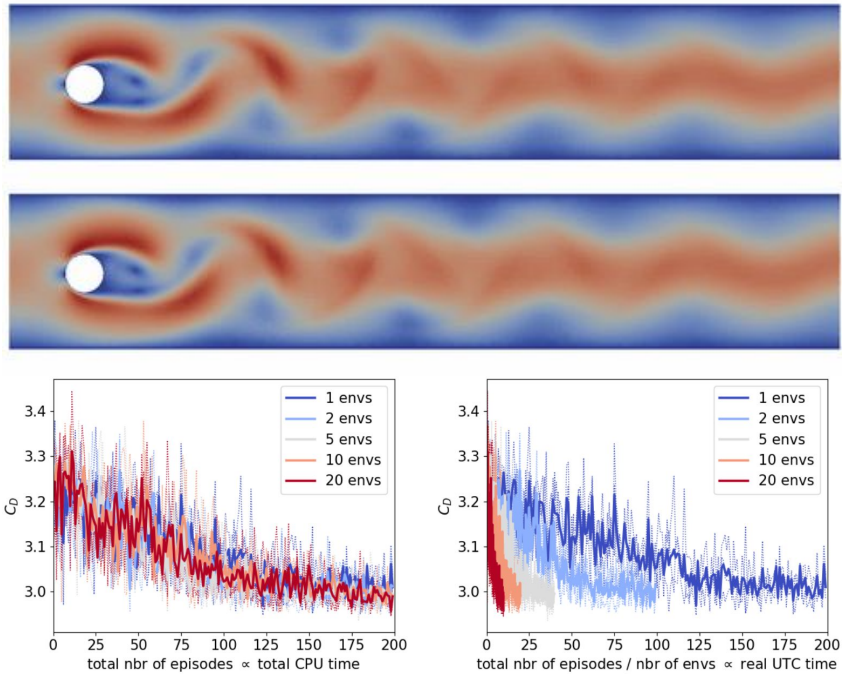
RL / Tensorforce takeaways:

- State space with multiple components
  - Global and ego-centric views of map
  - Categorical and continuous game state values
- Handling of discrete values
  - Main motivation for auto-network
- Exploration also for imperfect behavior
- Deployment to C#

Paper: http://www.exag.org/papers/EXAG_2019_paper_1.pdf
GitHub: https://github.com/SestoAle/DeepCrawl

Flow Control of the 2D Kármán Vortex Street

(Jean Rabault et al., University of Oslo)

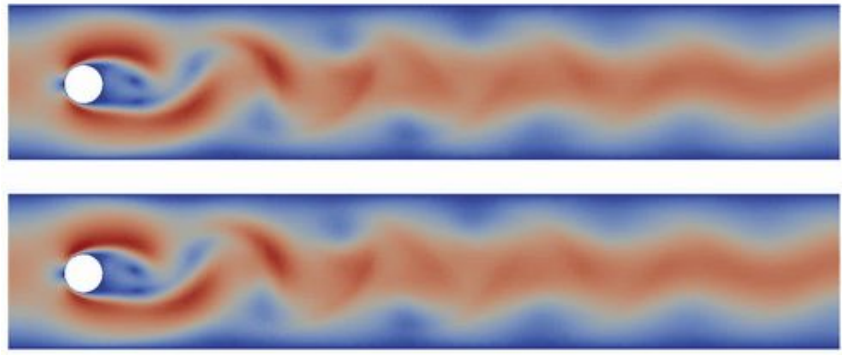


Flow Control of the 2D Kármán Vortex Street

(Jean Rabault et al., University of Oslo)

RL / Tensorforce takeaways:

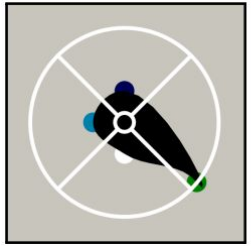
- Costly simulations using FEniCS
  - Simple parallelized environment execution
  - Speedup almost linear <= 60
- Importance of choosing the right characteristic timescales
  - Agent vs simulation timestep rate
  - Horizons and terminal



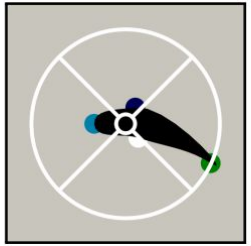
Papers: https://arxiv.org/abs/1808.07664 https://arxiv.org/abs/1906.10382
GitHub: https://github.com/jerabaul29/Cylinder2DFlowControlDRL

Direct shape optimization through DRL

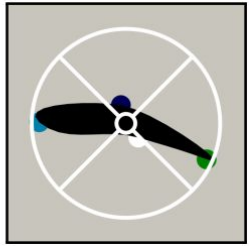
(Jonathan Viquerat et al., MINES ParisTech)



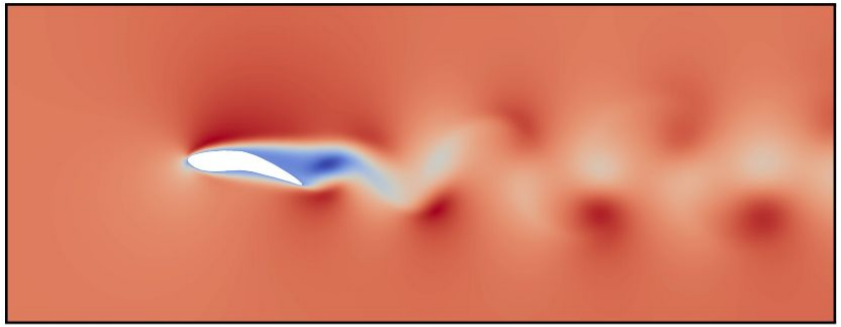
(a) Best shape with 4 points, 1 free point (3 d.o.f.s)



(b) Best shape with 4 points, 3 free points (9 d.o.f.s)



(c) Best shape with 4 points, 4 free points (12 d.o.f.s)



(d) Computed v_x velocity field at $Re \sim 600$ around shape 5c (the domain is cropped).

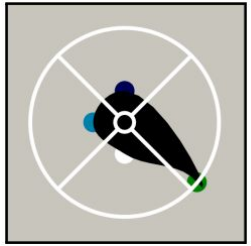
Direct shape optimization through DRL

(Jonathan Viquerat et al., MINES ParisTech)

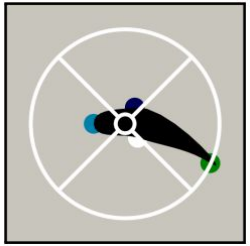
RL / Tensorforce takeaways:

- Importance of state/action parametrization
  - Unambiguous, normalized
- “Degenerate” 1-step RL
  - Non-differentiable optimization
- Potential of reward shaping:
  - Constraints via additional terms

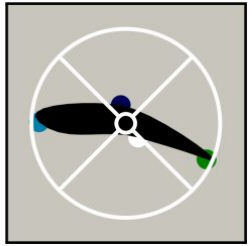
Paper: https://arxiv.org/abs/1908.09885



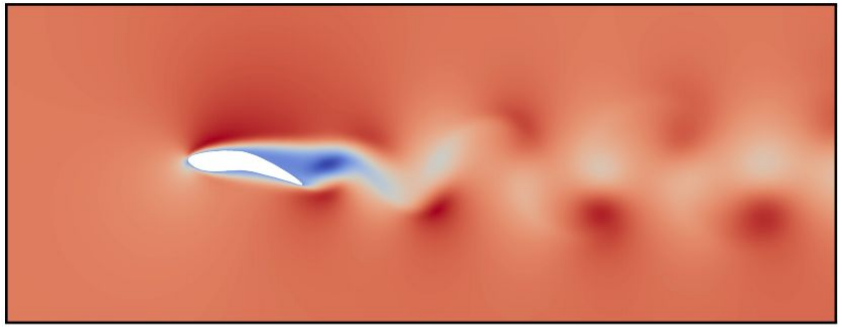
(a) Best shape with 4 points, 1 free point (3 d.o.f.s)



(b) Best shape with 4 points, 3 free points (9 d.o.f.s)



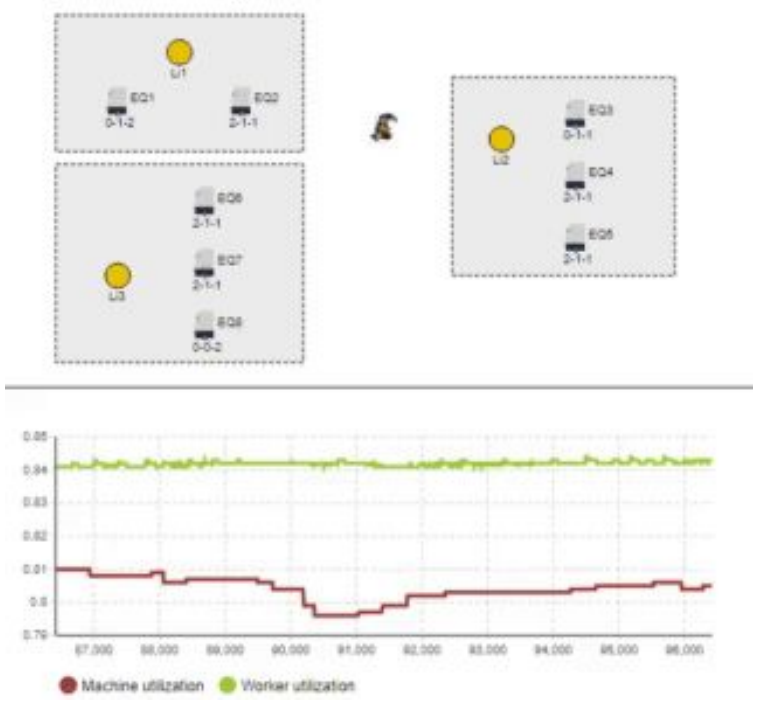
(c) Best shape with 4 points, 4 free points (12 d.o.f.s)



(d) Computed v_x velocity field at $Re \sim 600$ around shape 5c (the domain is cropped).

Autonomous order dispatching in the semiconductor industry

(KIT Institute of Production Science, Infineon)

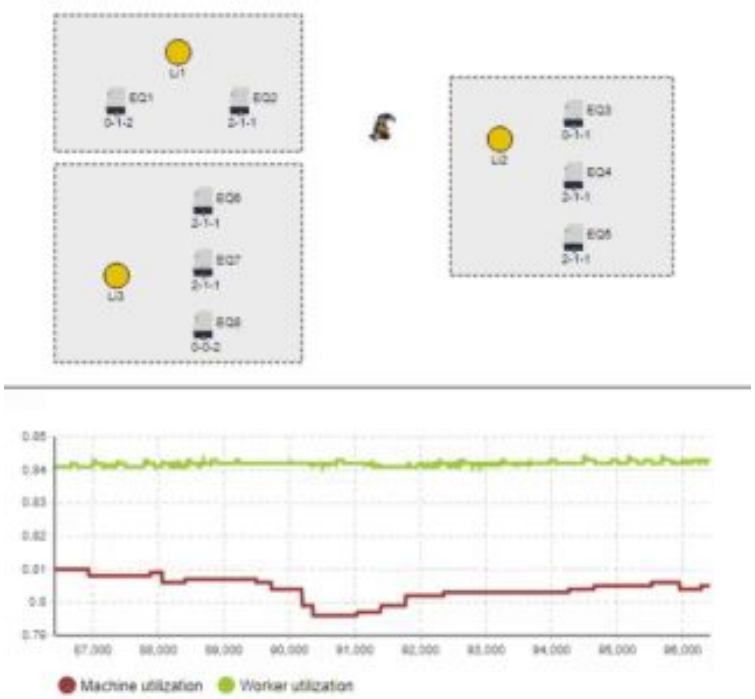


Autonomous order dispatching in the semiconductor industry

(KIT Institute of Production Science, Infineon)

RL / Tensorforce takeaways:

- Agent embedded in simulation framework
- Multiple workers controlled by the same RL agent interacting simultaneously
  - Different type of parallelized execution
- Masking of invalid actions



Paper: https://publikationen.bibliothek.kit.edu/1000091435

And more...

- Drones, autonomous driving
- Recommender systems
- (Bitcoin) trading
- Games

And more...

- Drones, autonomous driving
- Recommender systems
- (Bitcoin) trading
- Games

BMW of North America

Automotive

United States

The Ultimate Driving Machine®

•\tAdditional skills: Experience with Machine Learning frameworks (TensorFLow, TensorForce, Linux, ROS, C++, C#, Python, Vehicle Simulation, or related.

HUAWEI

Research Engineer - Reinforcement Learning

3.4 ★ **Huawei Technologies** – Markham

- Software development experience with at least one of the main stream deep learning tools such as Tensorflow, Keras, PyTorch, or Tensorforce

Summary

Tensorforce: *“TF Estimators for reinforcement learning”*

- Easy-to-use framework for applied DRL
- Fully modular RL library design with extensive configurability
- TensorFlow as only implementation platform
- Vision: enable (non-ML) practitioners to apply DRL in any application

Thanks for your attention!

Questions?