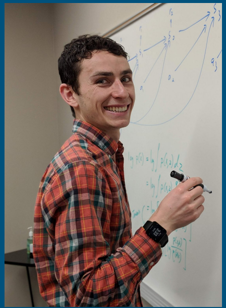
ICML
International Conference On Machine Learning

pomdp-baselines Public
Simple (but often Strong) Baselines for POMDPs in PyTorch - ICML 2022
Python 102 11

Recurrent Model-Free RL Can Be a Strong Baseline for Many POMDPs


Tianwei Ni
UdeM & Mila

Benjamin Eysenbach
CMU

Ruslan Salakhutdinov
CMU

Mila
Université de Montréal
Carnegie Mellon University

Why study POMDPs?

(Partially Observable MDPs)

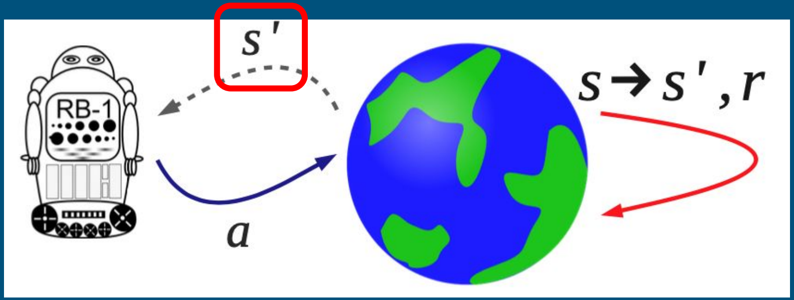
Why study POMDPs?

(Partially Observable MDPs)

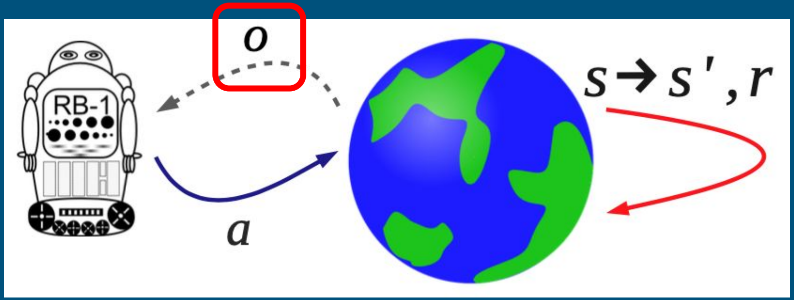
1. *They're **realistic**.*

POMDP: Observations instead of States

An MDP



A POMDP



State transition

$$s_{t+1} \sim F(s_{t+1} | s_t, a_t)$$

Observation emission

$$o_{t+1} \sim U(o_{t+1} | s_{t+1}, a_t)$$

Fig Source: Frans A Oliehoek and Christopher Amato. A concise introduction to decentralized POMDPs. Springer, 2016

Where do *States* come from?

- As long as there is error in sensors, we can only perceive *noisy* or *partial* version of *states*, i.e. observations
- In general, our real world and life could be viewed as POMDPs

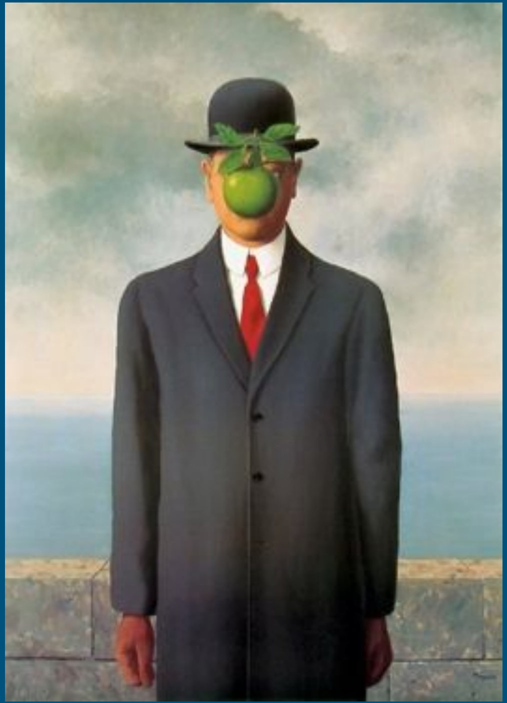
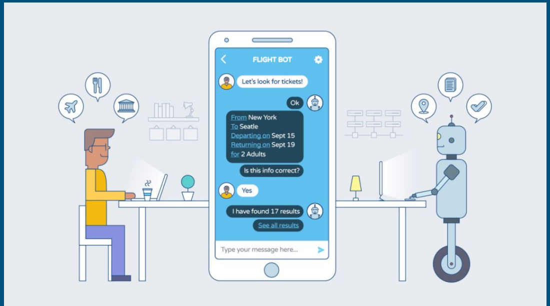


Fig Source: The Son of Man, René Magritte, 1964

POMDP applications

Robotics / Manufacturing

Healthcare / Medicine



Interactive NLP / Chatbot

Finance

Energy

Education

Why study POMDPs?

(Partially Observable MDPs)

Why study POMDPs?

(Partially Observable MDPs)

2. *They're **general**.*

A unified view of subareas in POMDPs

Subarea	s^h in dynamics?	s^h in reward?	Is s^h stationary?	Agent input	RL objective	Domain shift?
"Standard" POMDP	✓	✓	✗	`oar`	Avg	✗
Meta-RL	✗*	✓	✓	`oard`	Avg	✗
Robust RL	✓*	✗*	✓*	`oa`	Worst	✗
Generalization in RL	✓*	✗*	✓*	`oa`	Avg	✓*
Temporal credit assignment	✗	✓	✗	`oa`	Avg	✗

POMDPs are general

- Methods that can solve POMDPs can also solve each subarea
- But not vice versa

Solving POMDPs with RL

Inference and Control

Inference and Control

- **Inference**: estimate the underlying state (distribution)
- **Control**: RL on the inferred state space
- Model-based approaches: inference -> control
  - Learn an inference model and an RL algorithm separately
- Model-free approaches: inference <-> control
  - Jointly learn (implicit) inference and control with a sequence model and RL
  - Our focus

Recurrent Model-Free RL

Classic in RNN literature (1990s)
Revived in Deep RL (2016-17)

Why Recurrent Model-Free RL?

Why Recurrent Model-Free RL?

1. *It is **simple** to understand and implement.*

Memory Perspective

- In theory, we do not need explicit inference
- We just need to make sure that policy has (sufficient) memory
- A modern memory architecture is Recurrent Neural Network (RNN)
- Therefore, we can simply replace *Markovian* model (e.g. MLP) with *memory-based* model (e.g. LSTM/GRU)

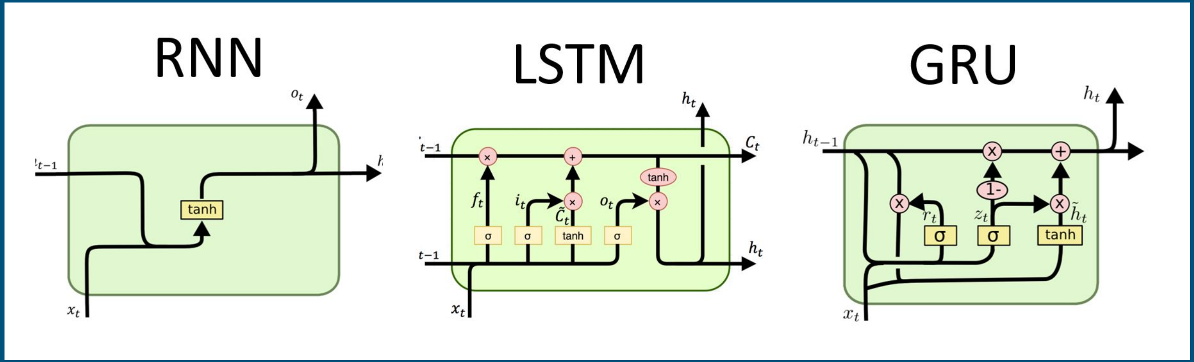
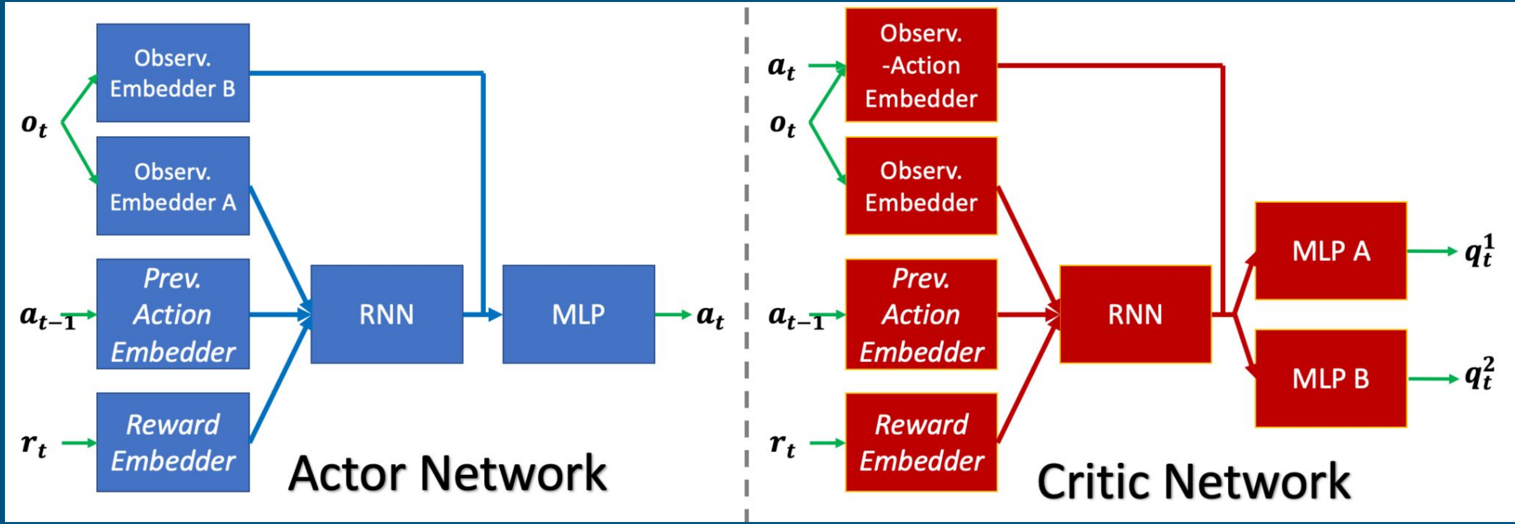


Fig Source: http://dprogrammer.org/rnn-lstm-gru

(Our) Recurrent Actor-Critic Architecture



Observation shortcut is also used in prior work and implementation

Why Recurrent Model-Free RL?

Why Recurrent Model-Free RL?

2. *It is **expressive in theory**.*

RNNs are universal function approximators.

Why Not Recurrent Model-Free RL?

*It is **poor** in practice.* ***(Many Prior work)***

Why ~~Not~~ Recurrent Model-Free RL?

*It is **poor** in practice. **(Many Prior work)***

*3. It can be **powerful** in practice. **(This work)***

Recurrent Model-Free RL: Our Key Considerations

- **Recurrent actor and critic:**
  - Share an RNN
  - Separate RNNs
- **Agent input space:**
  - Observation
  - Action
  - Reward
- **RL algorithm:**
  - On-policy such as PPO and A2C
  - Off-policy such as TD3 and SAC
- **RNN architecture and context length**
  - LSTM or GRU
  - Length: short, medium, or long

Legend
- Factor that is largely ignored in prior work
- Recommended options

How Prior Work Consider these Factors? Why Fail?

- Since recurrent model-free RL is simple, it is widely used as a baseline
- But it is shown to have poor performance in most cases

Algorithm	Domain / Benchmark	Arch	Encoder	Inputs	Len	RL
Duan et al. (2016)	Meta-RL	separate	GRU	oard	1000	TRPO, PPO
Wang et al. (2017)	Meta-RL	shared	LSTM	oart	5-150	A2C
Baseline in Rakelly et al. (2019)	Meta-RL	separate	GRU	oard	100	PPO
Baseline in Zintgraf et al. (2020)	Meta-RL	separate	GRU	oard	Max	A2C, PPO
Baseline in Fakoor et al. (2020)	Meta-RL	separate	GRU	oar	10-25	TD3
Baseline in Yu et al. (2019)	Meta-RL	separate	GRU	oard	500	PPO
Kostrikov (2018)	POMDP	shared	GRU	o	5-2048	PPO, A2C
Ding (2019)	POMDP	separate	LSTM	oa	150	TD3, SAC
Meng et al. (2021)	POMDP	separate	LSTM	oa	1-5	TD3
Yang & Nguyen (2021)	POMDP	separate	both	oa	Max	TD3, SAC
Baseline in Igl et al. (2018)	POMDP	shared	GRU	oa	25	A2C
Baseline in Han et al. (2020)	POMDP	shared	LSTM	o	64	SAC
Baseline in Zhang et al. (2021)	Robust RL	separate	LSTM	o	100	PPO
Baseline 1 in Packer et al. (2018)	Generalization in RL	shared	LSTM	o	128-512	PPO, A2C
Baseline 2 in Packer et al. (2018)	Generalization in RL	separate	LSTM	oard	128-512	PPO, A2C
Baseline in Hung et al. (2018)	Temporal credit assignment	shared	LSTM	oar	Max	A3C
Baseline in Liu et al. (2019)	Temporal credit assignment	separate	LSTM	oa	Max	PPO
Baseline in Raposo et al. (2021)	Temporal credit assignment	shared	LSTM	oar	10-60	IMPALA
Our work	Meta-RL (Dorfman et al., 2020)	separate	LSTM	oard	64	TD3
	Meta-RL (Zintgraf et al., 2020)	separate	GRU	oard	Max	SAC
	POMDP (Han et al., 2020)	separate	GRU	oa	64	TD3
	Robust RL (Jiang et al., 2021)	separate	LSTM	o	64	TD3
	Generalization in RL (Packer et al., 2018)	separate	LSTM	o	64	TD3
	Temporal credit assignment (Raposo et al., 2021)	separate	LSTM	o	Max	SAC-D

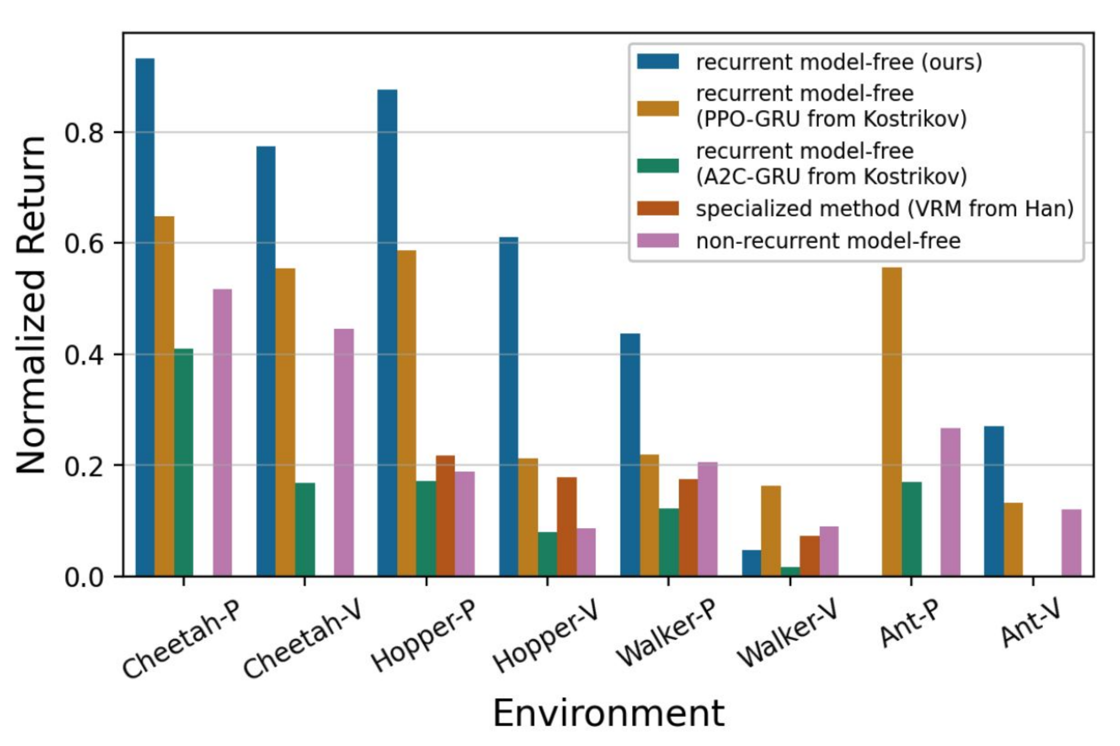
best single variants

A Large-Scale Empirical Study on Many POMDPs

Comparison on several benchmarks

- In each subarea, we compare the corresponding **specialized (more complex)** methods on the benchmark **where they were evaluated in their paper**
- 6 benchmarks with **21 environments**
  - Mostly state-based, continuous control
  - Also image-based, discrete control
- Our implementation of RNN policy is at least on par with (if not greatly outperforms) them in **18 environments**

Example: Standard POMDPs benchmark from VRM

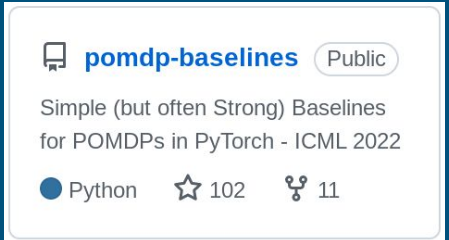


- VRM: a model-based off-policy approach
- PPO/A2C-GRU: recurrent model-free on-policy approaches
- Our recurrent model-free RL is better than VRM and PPO-GRU in 6/8 environments

Closing Remarks

Code

- Open-sourced in GitHub
  - We value reproducibility
- Welcome to use it as a baseline!



https://github.com/twni2016/pomdp-baselines

Takeaway

- While MDPs prevail in RL research, POMDPs prevail in real world and life
- Recurrent model-free RL, a simple approach to POMDP, can be a strong baseline in many environments, contrary to common belief
- Implementation matters: several design choices in recurrent model-free RL
- Consider using our code to incentivize future research on history-dependent policies and POMDPs

Acknowledgement


Pierre-Luc Bacon


Pierluca D’Oro


Murtaza Dalal


Paul Pu Liang


Sergey Levine


Michel Ma


Evgenii Nikishin


Hao Sun


Maxime Wabartha


Luisa Zintgraf

Thank you for watching!