Open your mind. LUT.
Lappeenranta University of Technology

PO4AO: XAO control with model-based reinforcement learning

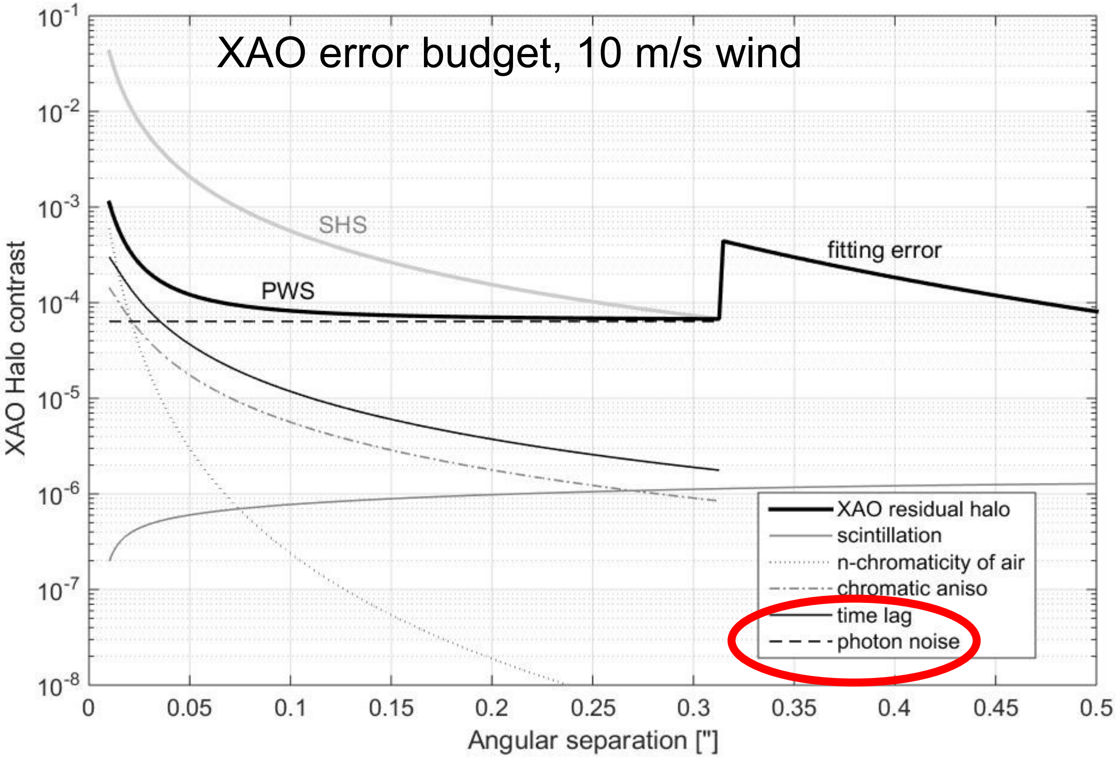
Finland: **Jalo Nousiainen** (LUT), Chang Rajani (UoH), Tapio Helin (LUT)

ESO: Markus Kasper, Byron Engler, Christophe Verinaud, Taissir Heritier

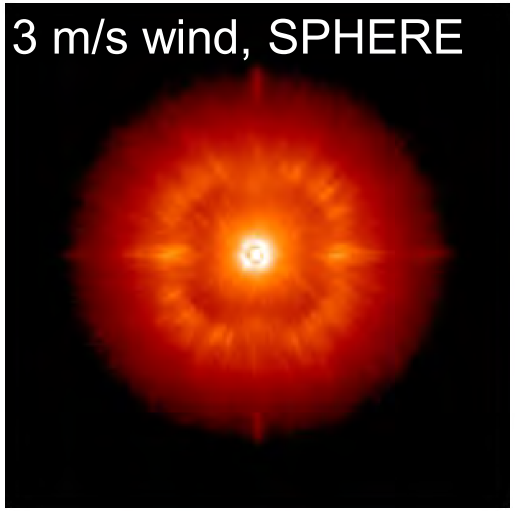
UoA: Sebastiaan Haffert

XAO error budget dominated by time lag

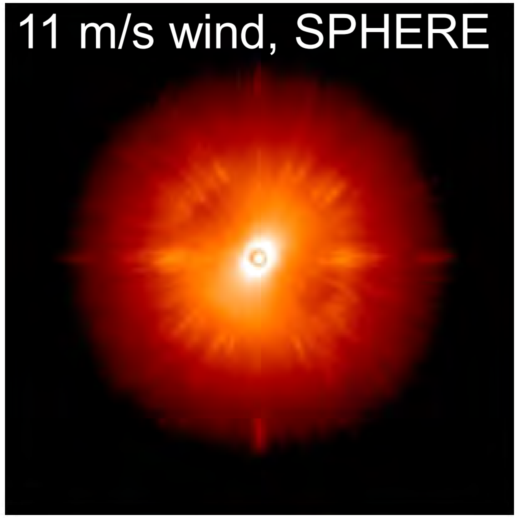
- AO error budget dominated by temporal error and photon noise
- Observing time $t_{exp} \propto contrast$
- Goal: improve contrast by factor 3-10



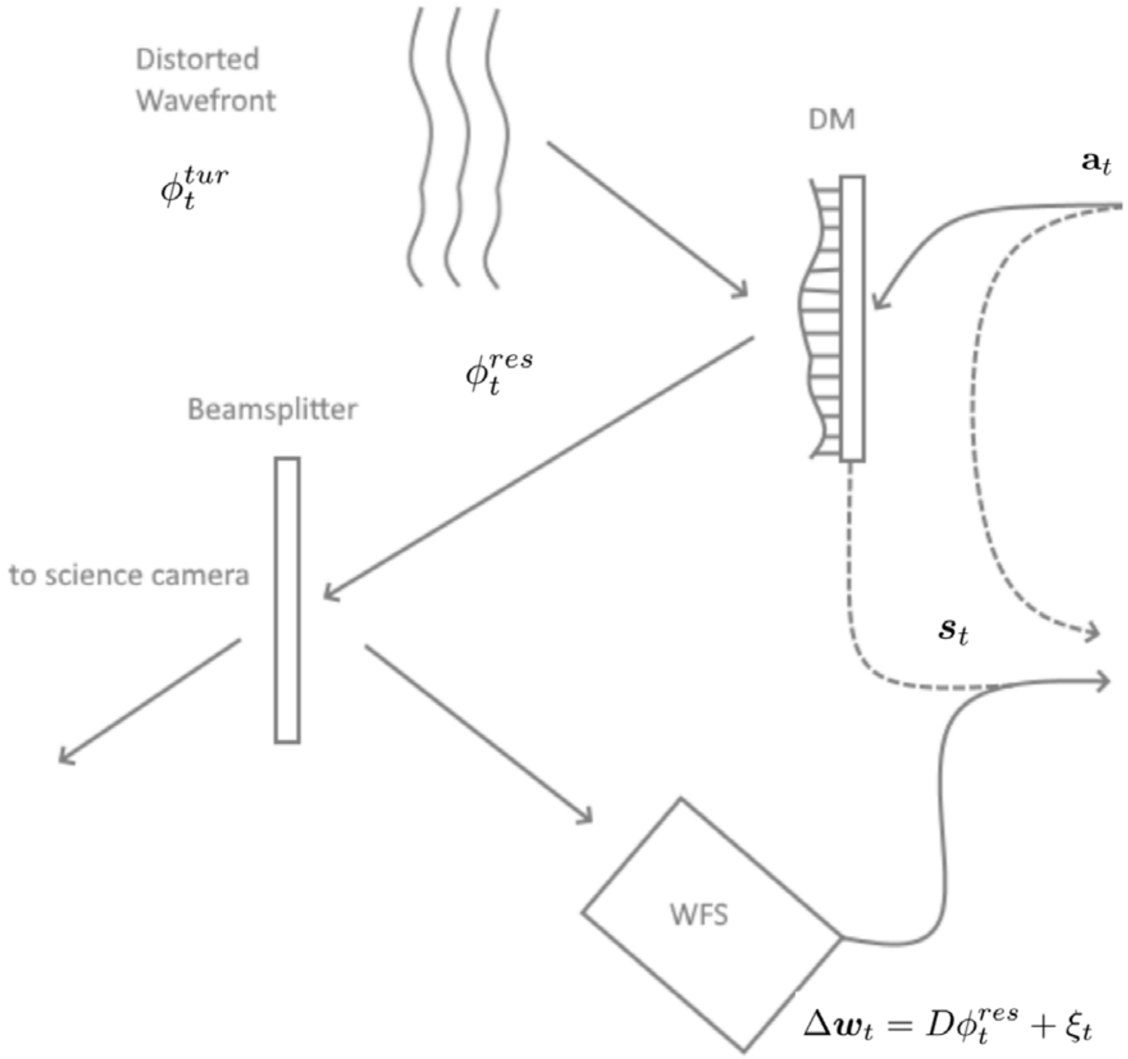
3 m/s wind, SPHERE



11 m/s wind, SPHERE



Classical AO control (I-controller)

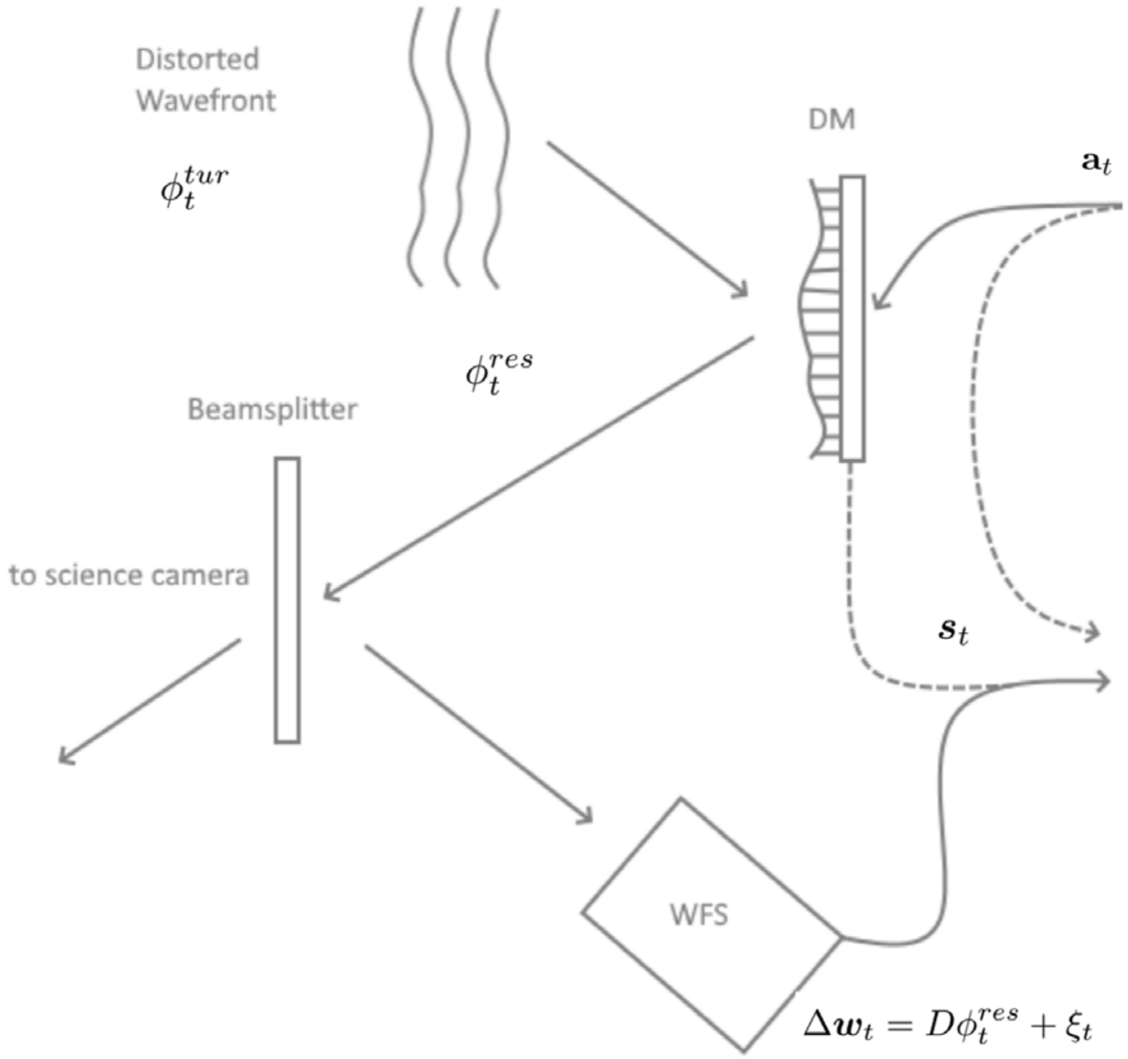


Classical AO control:

$\Delta a = R\Delta w$

$a_t = la_{t-1} + g\Delta a$

Can we do better?



Classical AO control:

$\Delta a = R\Delta w$

$a_t = l a_{t-1} + g\Delta a$

Time delay, photon noise

Mis-registration

Optical gain compensation

DM dynamics?

Non-linearities?

Reinforcement learning

"Reinforcement learning is learning what to do—how to map situations to actions—so as to maximize a numerical reward signal. The learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them."

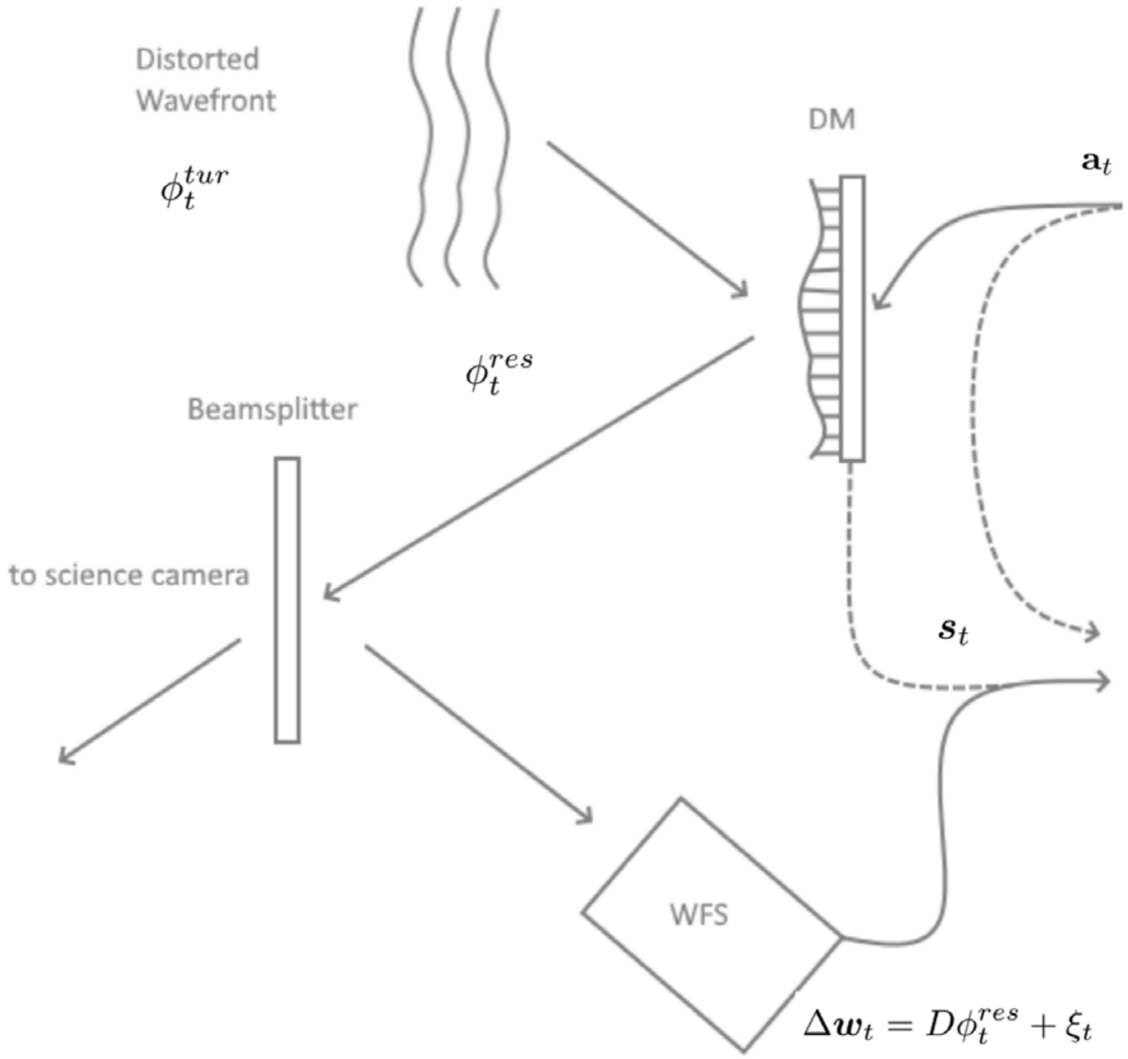
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

RL Literature in AO

1. Motivation and first steps:
   - J. Nousiainen et al., "Adaptive optics control using model-based reinforcement learning," Opt. Express (2021)
2. Refined method and first lab results (MagAO-X):
   - J. Nousiainen et al., “Toward on-sky adaptive optics control using reinforcement learning”, A&A (2022)
3. Preliminary GHOST test bench results:
   - J. Nousiainen et al. "Advances in model-based reinforcement learning for adaptive optics control." SPIE proceeding
4. More to come soon!

5. Other groups:
   - Landman, Rico, et al, JATIS (2021), SPIE (2020)
   - Pou Mulet et al. Opt. Express (2022)
   - Haffert, Sebastiaan Y., et al. JATIS (2021)

Classical AO control (I-controller)

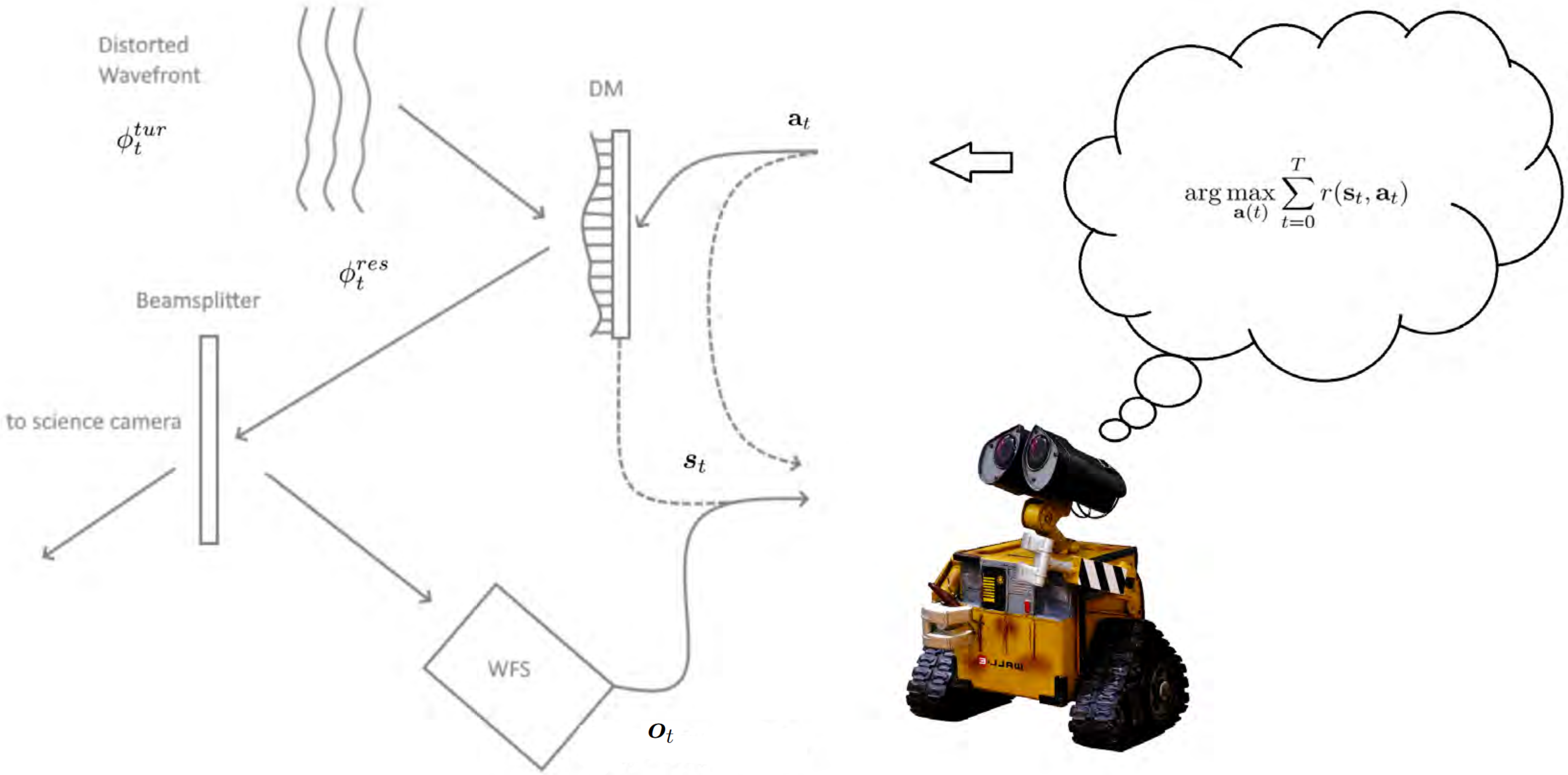


Classical AO control:

$\Delta a = R\Delta w$

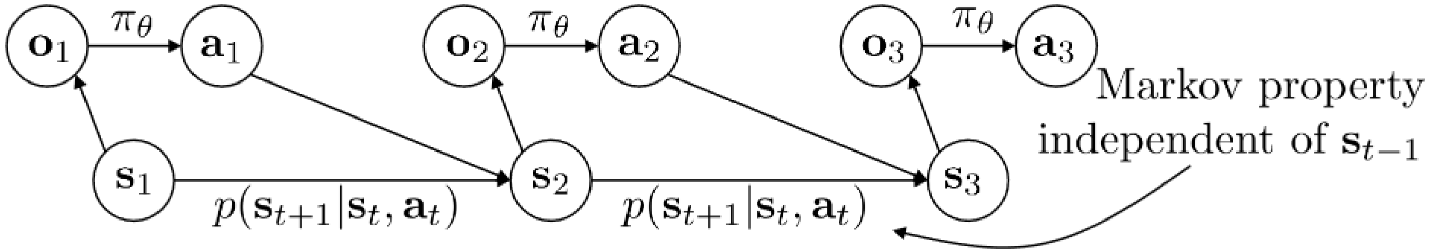
$a_t = l a_{t-1} + g\Delta a$

Reinforcement Learning for AO



Mathematical framework for RL: Describe the system as Markov decision process

- A Markov decision process contains:
  - A set of possible environment states $s \in S$
  - A set of possible actions $a \in A$
  - A real valued reward function $r(s_t, a_t)$
  - Transition dynamics $p(s_{t+1}|s_t, a_t)$
- Markov property
  - Next state, s_{t+1}, depends on the current state, s_t, and the decision makers action, a_t, **only**
- Partially observed MDP
  - State is observer through a measurement model o_t = f(s_t)



Mathematical framework for RL: Describe the system as Markov decision process

- A Markov decision process contains:
  - A set of possible environment states $s \in \mathrm{S}$
  - A set of possible actions $a \in \mathrm{A}$
  - A real valued reward function $r(s_t, a_t)$
  - Transition dynamics $p(s_{t+1}|s_t, a_t)$
- Markov property
  - Next state, s_{t+1}, depends on the current state, s_t, and the decision makers action, a_t
- Partially observed MDP
  - Can be handled as MDP (in some cases) by adding past actions and observation:

State: $s_t = \{o_t, o_{t-1} \cdots, o_{t-k}; a_t, a_{t-1} \cdots a_{t-k}\}$

Transition dynamics: $p(o_{t+1}|s_t, a_t) \approx p(o_{t+1}|o_{t-h:t}, a_{t-h:t})$

AO system as MDP

- A Markov decision process contains:
  - A set of possible environment states $s \in S$ a set of history DM commands, a, and WFS frames, o
  - A set of possible actions $a \in A$ The residual DM control voltages
  - A real valued reward function $r(s_t, a_t)$ e.g., negative distance from the flat reference
  - Transition dynamics $p(o_{t+1}|s_t, a_t)$ contains information on atmosphere evolution, mis.reg, OG, latency..
- Approximative Markov property

$$s_t = \{o_t, o_{t-1} \cdots, o_{t-k}; a_t, a_{t-1} \cdots a_{t-k}\}$$

Model-based RL for AO

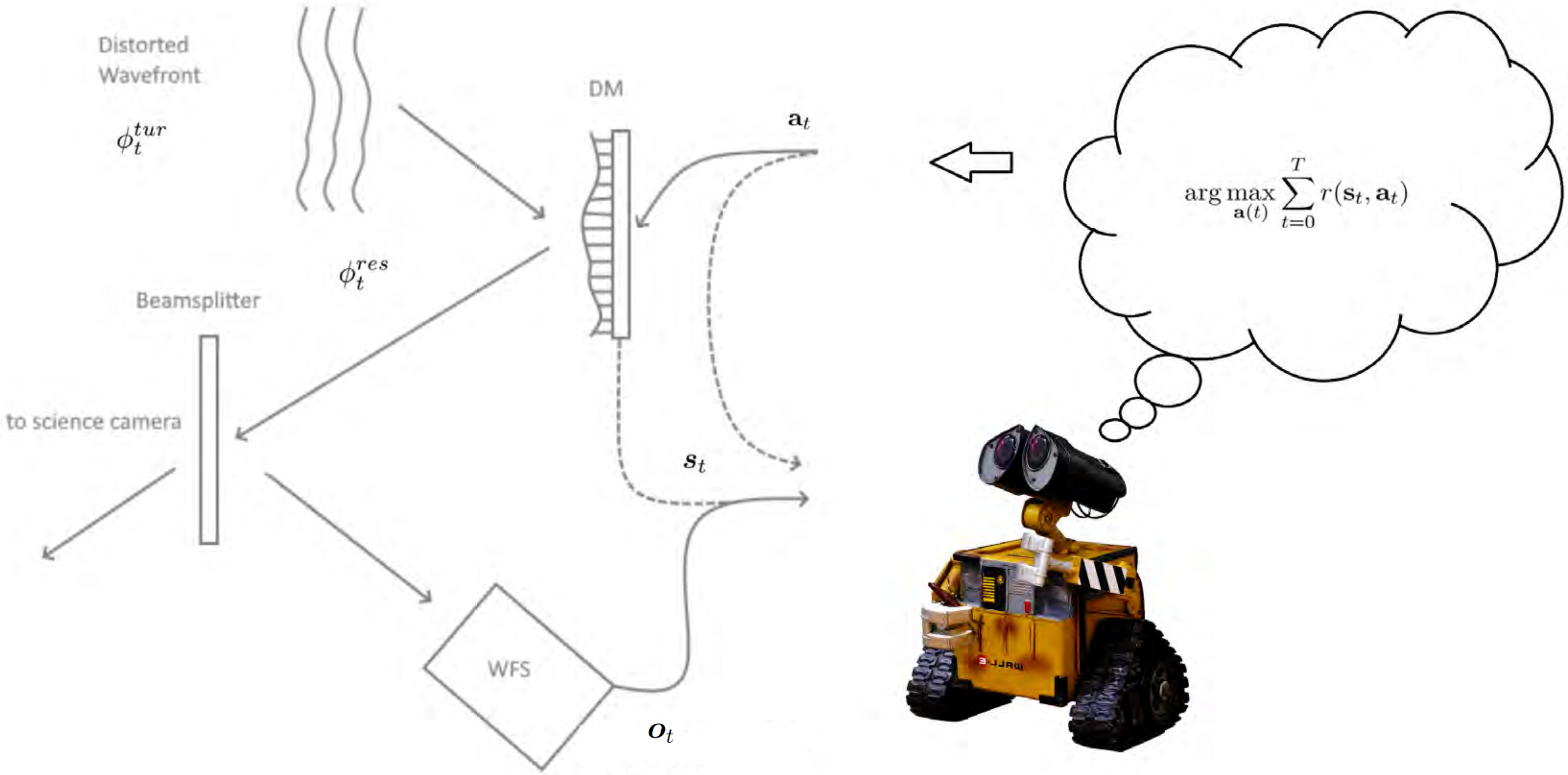
- A Markov decision process contains:
  - A set of possible environment states $s \in S$ contains a set of history commands and WFS frames
  - A set of possible actions $a \in A$ the residual DM control volatages
  - A real valued reward function $r(s_t, a_t)$ Negative distance from the flat reference
  - Transition dynamics $p(o_{t+1}|s_t, a_t)$ contains information on atmosphere evolution, mis.reg, OG, latency..
- Approximative Markov property

$$s_t = \{o_t, o_{t-1} \cdots, o_{t-k}; a_t, a_{t-1} \cdots a_{t-k}\}$$

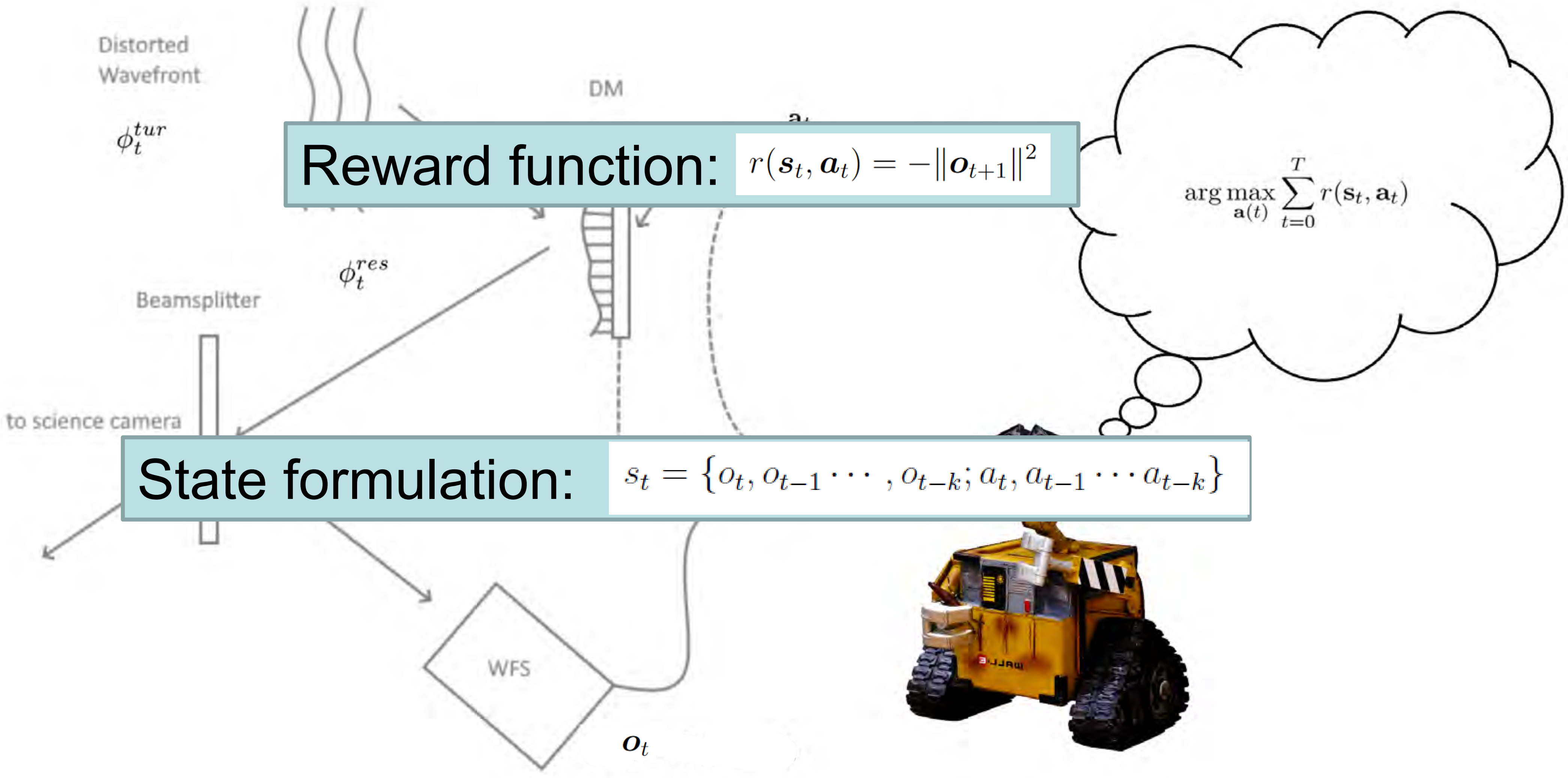
Model-based RL aims to learn the transition dynamics and use it to derive optimal controller

1. Dynamics: $\hat{p}_\omega(s_{t+1}|s_{t-1}, a_{t-1}) \approx p(s_{t+1}|s_t, a_t)$
2. The controller is for example
   a policy function (predictive control law) $\pi_\phi(a_t|s_t)$

PO4AO: Policy optimization for AO



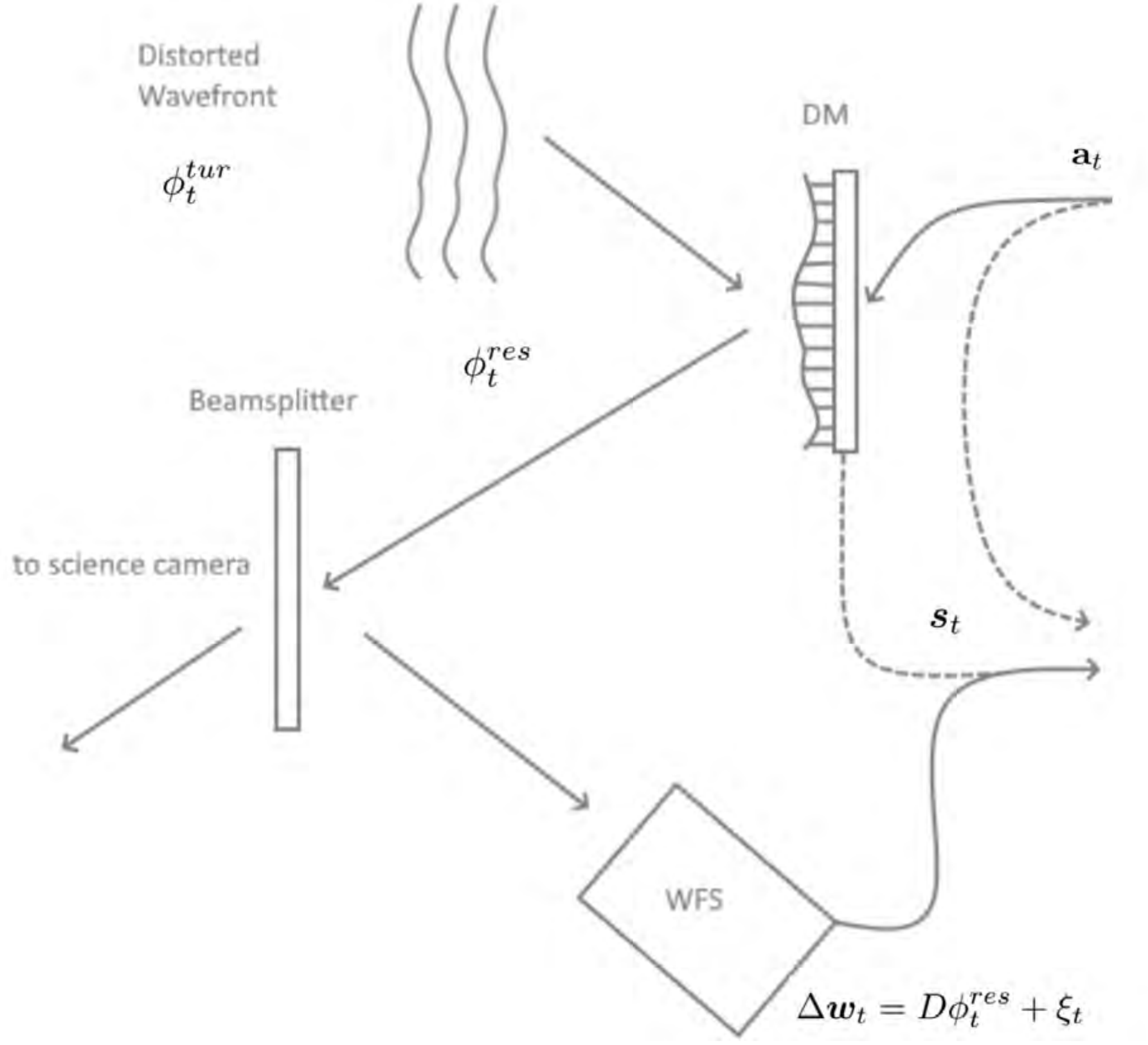
PO4AO: Policy optimization for AO

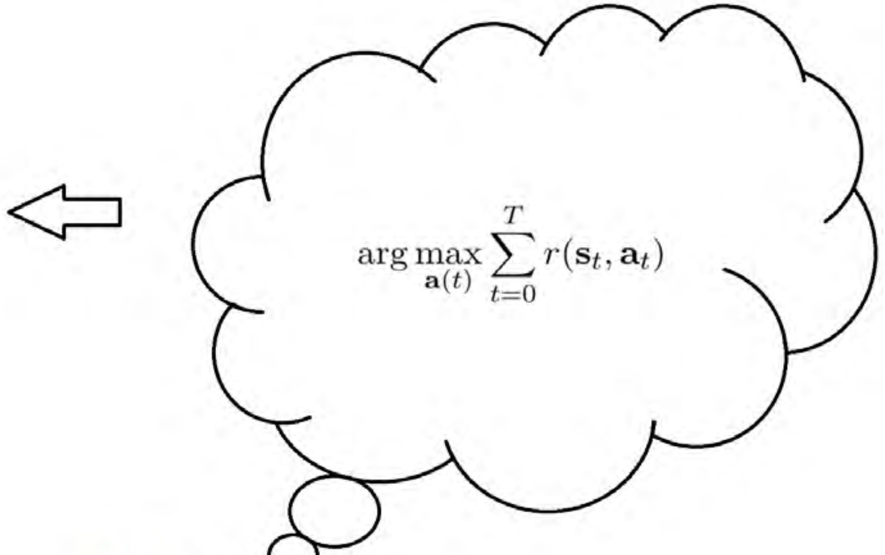


Reward function: $r(\boldsymbol{s}_t, \boldsymbol{a}_t) = -\|\boldsymbol{o}_{t+1}\|^2$

State formulation: $s_t = \{o_t, o_{t-1} \cdots, o_{t-k}; a_t, a_{t-1} \cdots a_{t-k}\}$

PO4AO: Policy optimization for AO





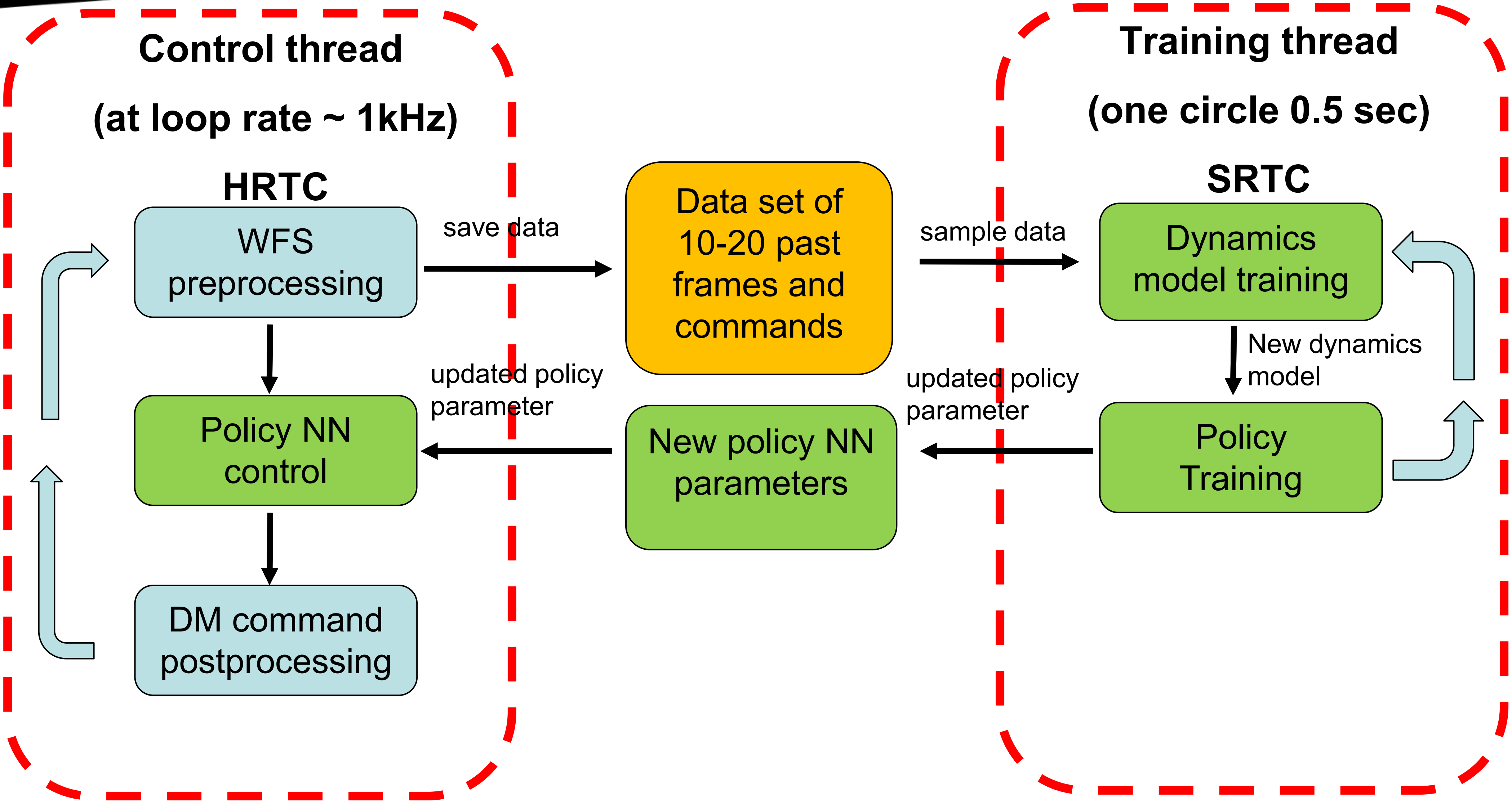
$$\arg\max_{\mathbf{a}(t)} \sum_{t=0}^{T} r(\mathbf{s}_t, \mathbf{a}_t)$$

PO4AO: Policy Optimization for AO

Dynamics Model (CNN): $p_\omega(s_{t+1}|s_t, a_t)$
Policy (recon and control, CNN): $\pi_\theta(a_t|s_t)$

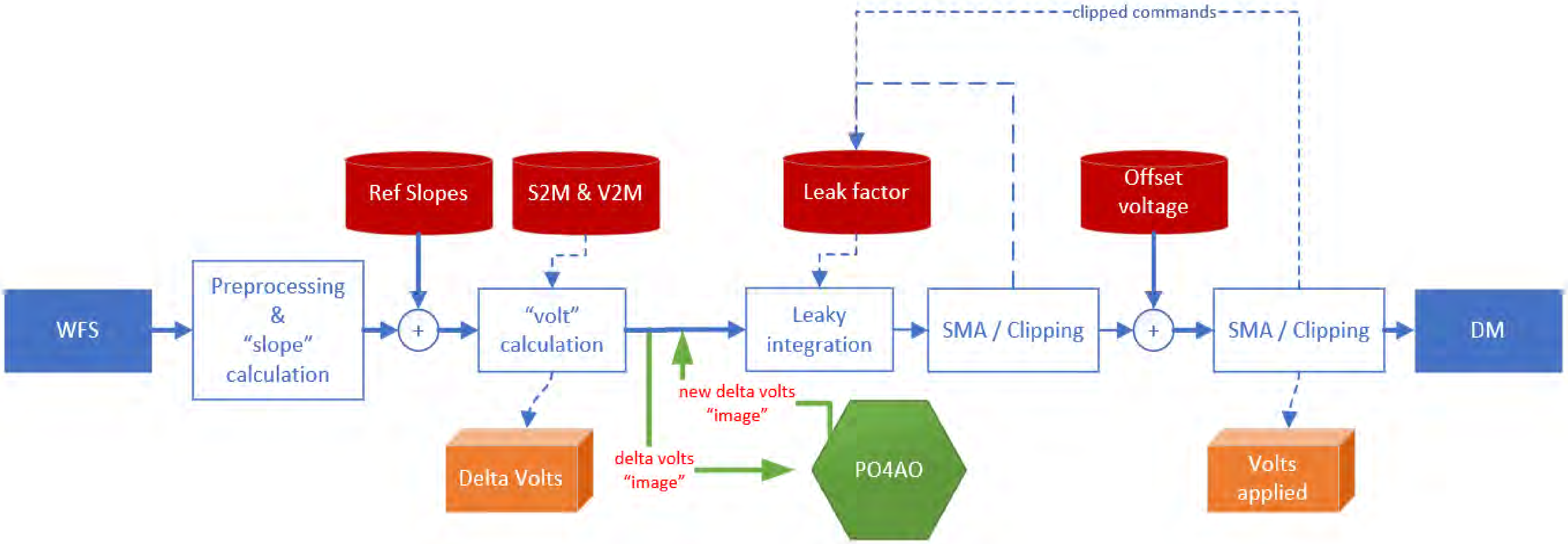
Iterate over episodes:
1. Run policy, collect data
2. Improve dyn. model (supervised learning)
3. Improve policy using improved dyn.model

PO4AO contains two parallel processes



Open your mind. LUT.
Lappeenranta University of Technology

PO4AO on GHOST



- Control thread connected to COSMIC RTC using single GPU
- Training thread uses different GPU and is fully Python

Code (PyTorch) available:
https://github.com/jnousi/PO4AO

Code and lab results:
Nousiainen, J. et al . JATIS submitted in August

Results

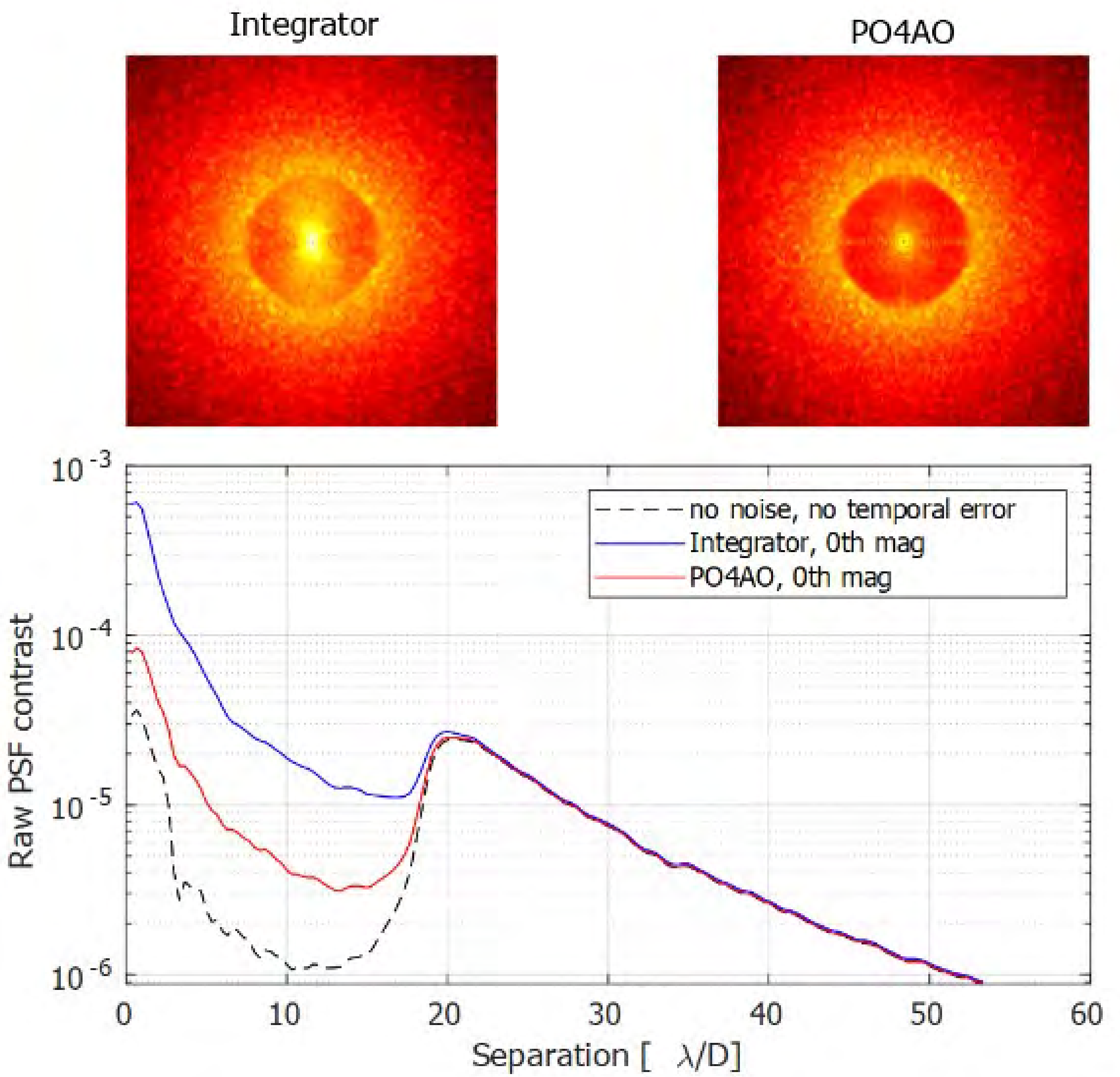
- PO4AO is a non-linear method
  - Hard to analyze
  - No analytical stability bounds can be established

- … but we can test methods in different simulations

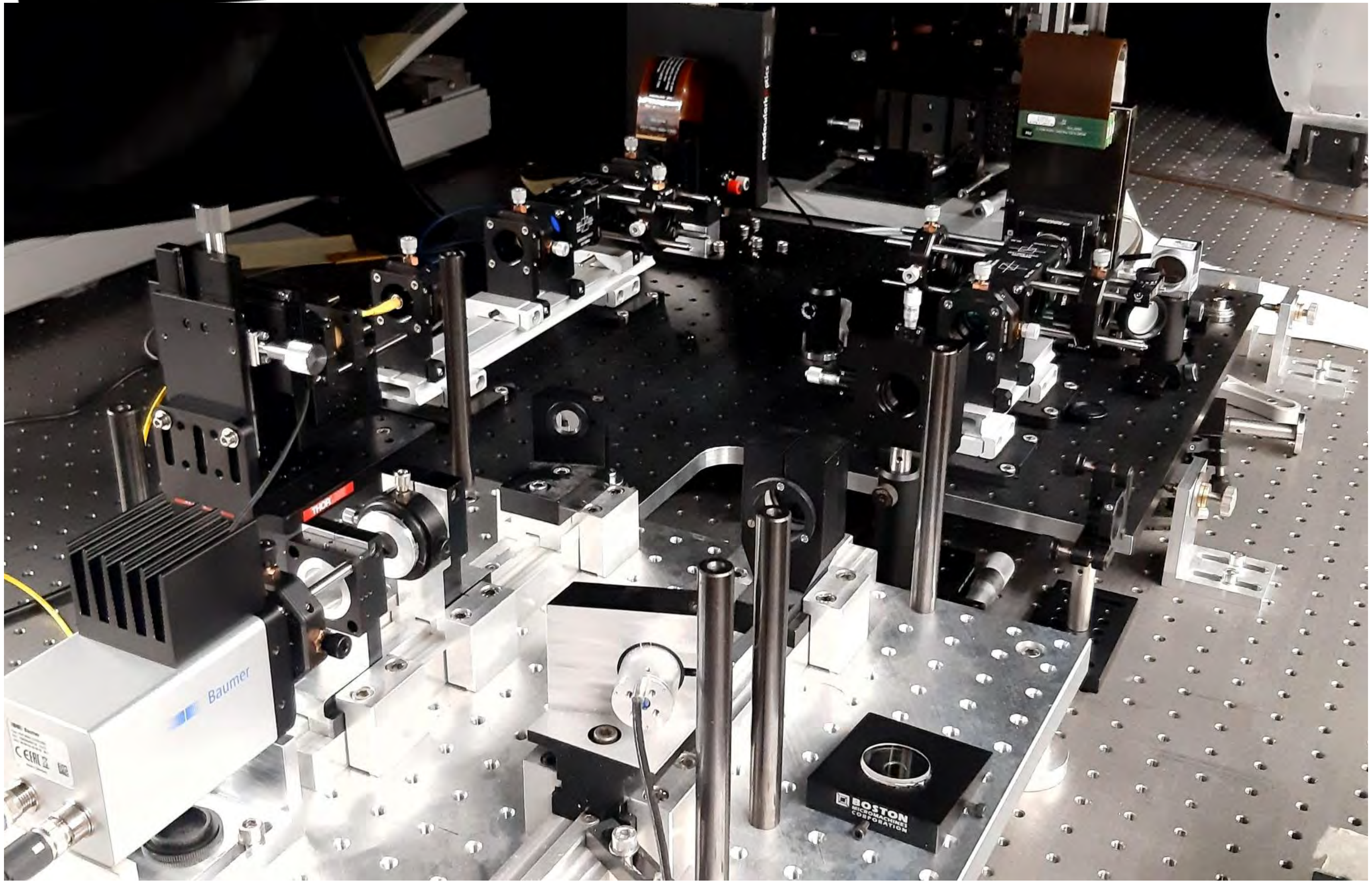
PO4AO simulations

Nousiainen et al., A&A, 2022

- 40x40 (VLT) and 120x120 (ELT) fixed PWS with Policy latency (< 1 ms)
- Training times ~10s for 1 kHz framerate, method follows environmental changes on such a timescale
- **Factor 4-7 contrast improvement with PWS** reconstruction (factor 10-20 with ideal WFS, limited by DM infl. functions)
- Features: Self-calibrating, Predictive, Robust to noise, Robust to data-mismatch, can correct unexpected errors (?)



GHOST bench at ESO

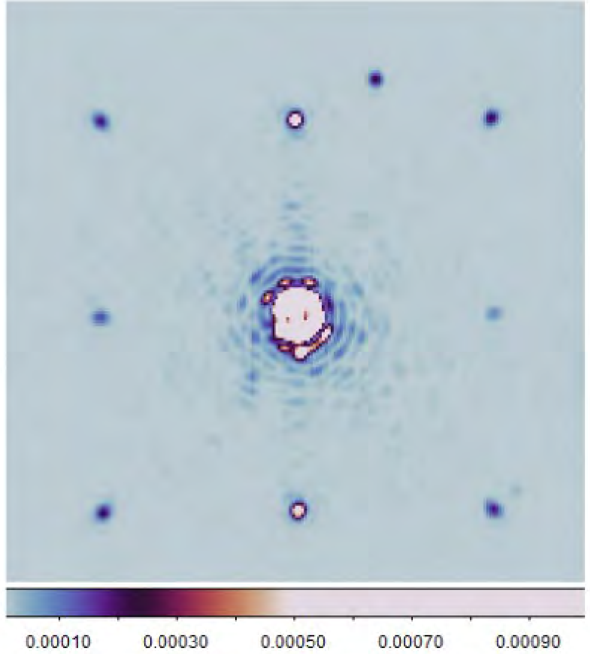


- SLM Meadowlark injects turbulence at 420Hz
- BMC 492-1.5 DM (ETH loan)
  - 300 um pitch
  - 100% actuator yield
- PWS (Arcetri design)
  - 10 GigE camera (Sony IMX426 CMOS)
  - PI modulation mirror SL-325
- GPU RTC implementing
  - COSMIC platform (ANU/LESIA, August 2022)
  - Python code (ready, B. Engler)
- Now we also have a Lyot Coronagraph

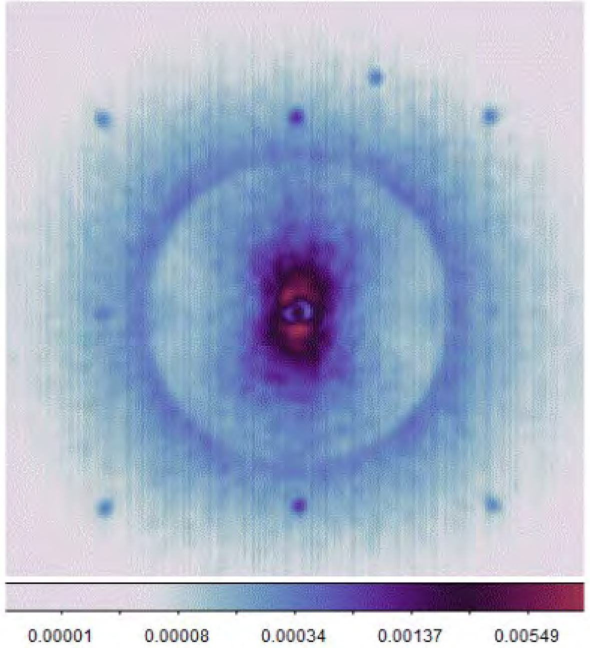
PO4AO on GHOST

We simulate a cascaded AO system

1. Numerically simulated 40x40 first stage
2. SLM replays the residual phase
3. 2nd stage runs ~2 times faster
4. Light source 770 nm

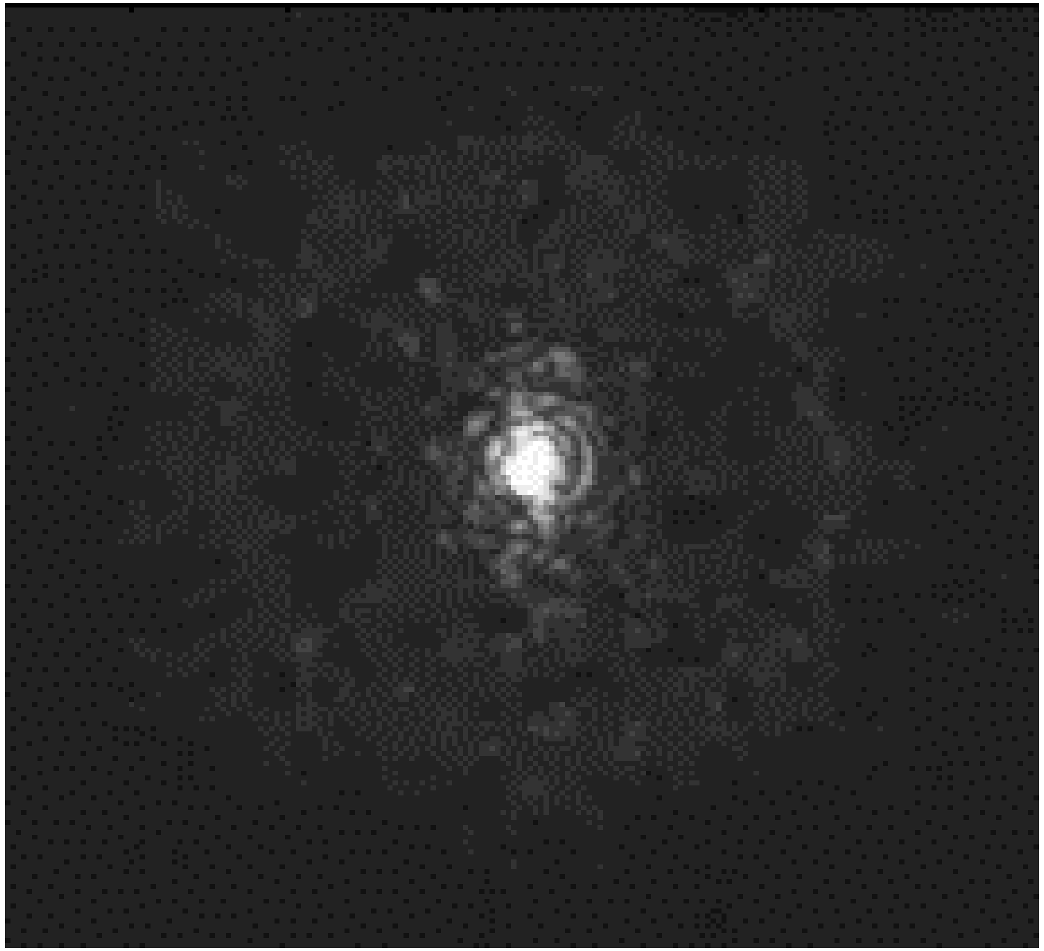


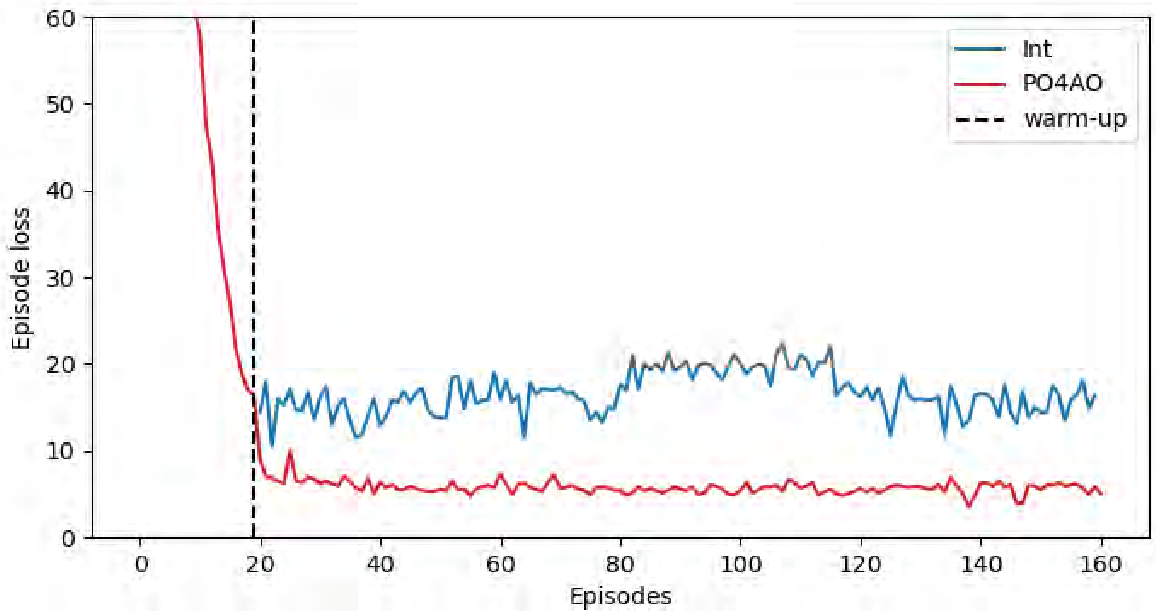
(a) GHOST PSF without turbulence



(b) GHOST open-loop PSF with SLM simulated turbulence.

PO4AO GHOST

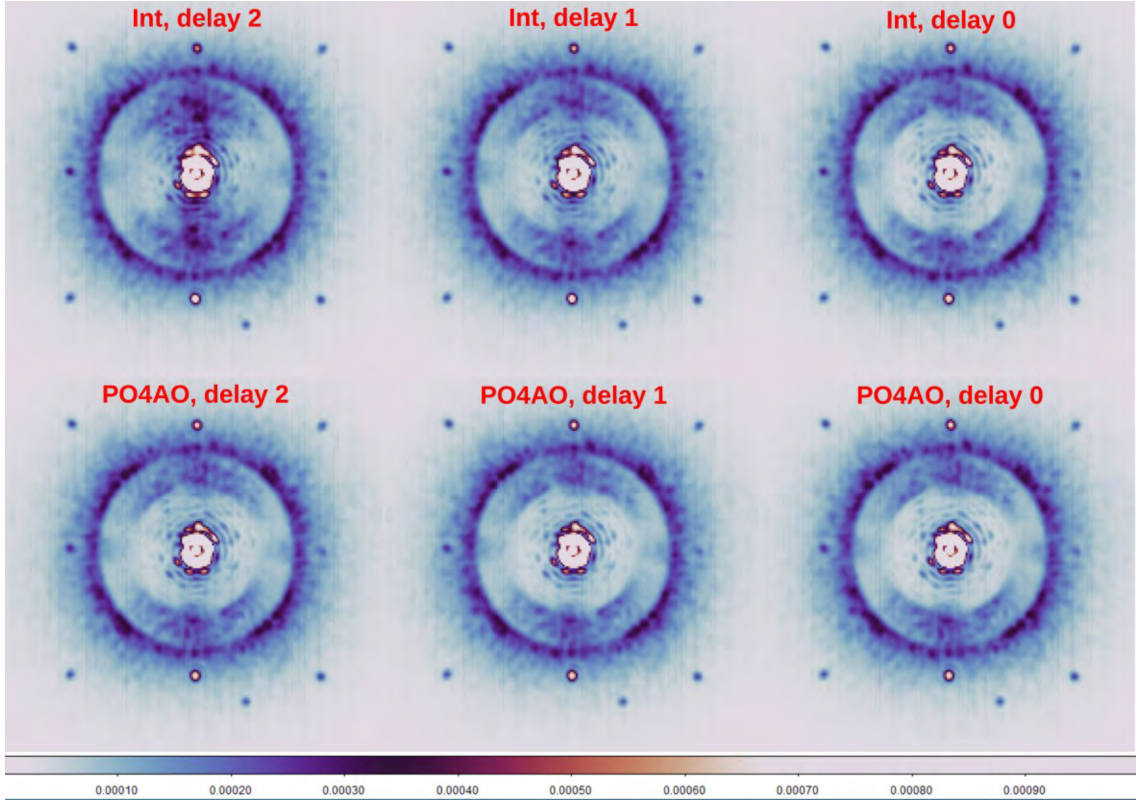




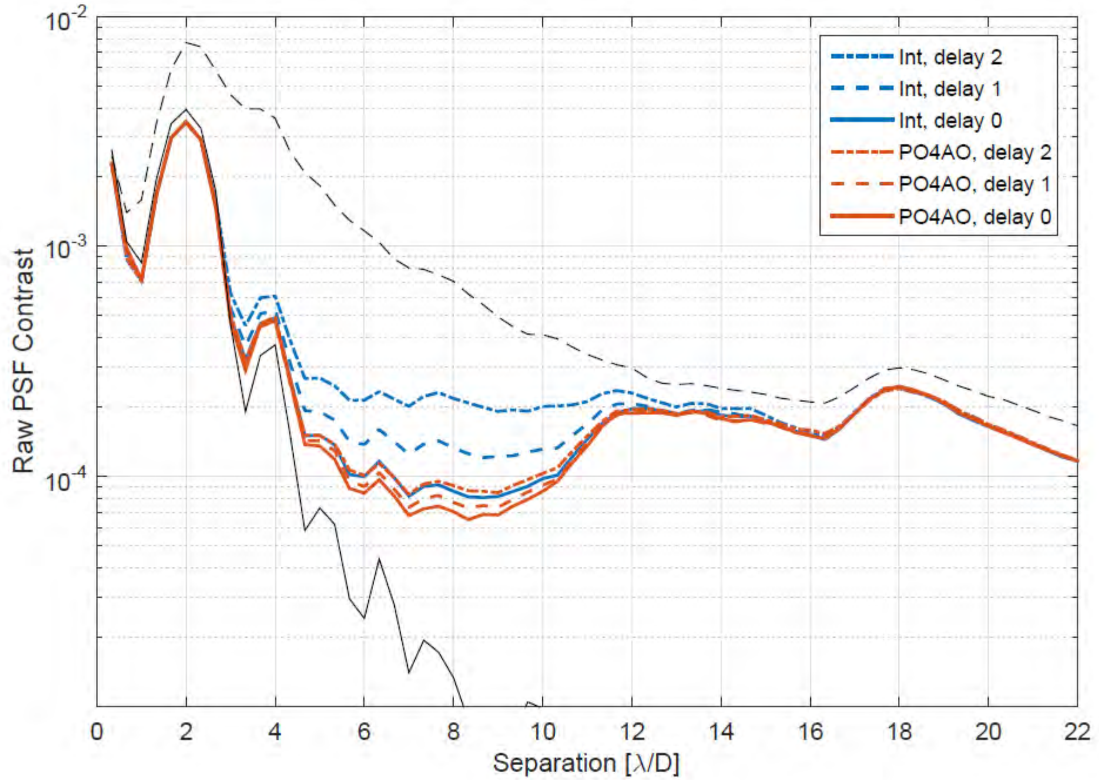
Episode loss
Episodes
0
10
20
30
40
50
60
0
20
40
60
80
100
120
140
160
Int
PO4AO
warm-up

Predictive control

Long exposure PSF

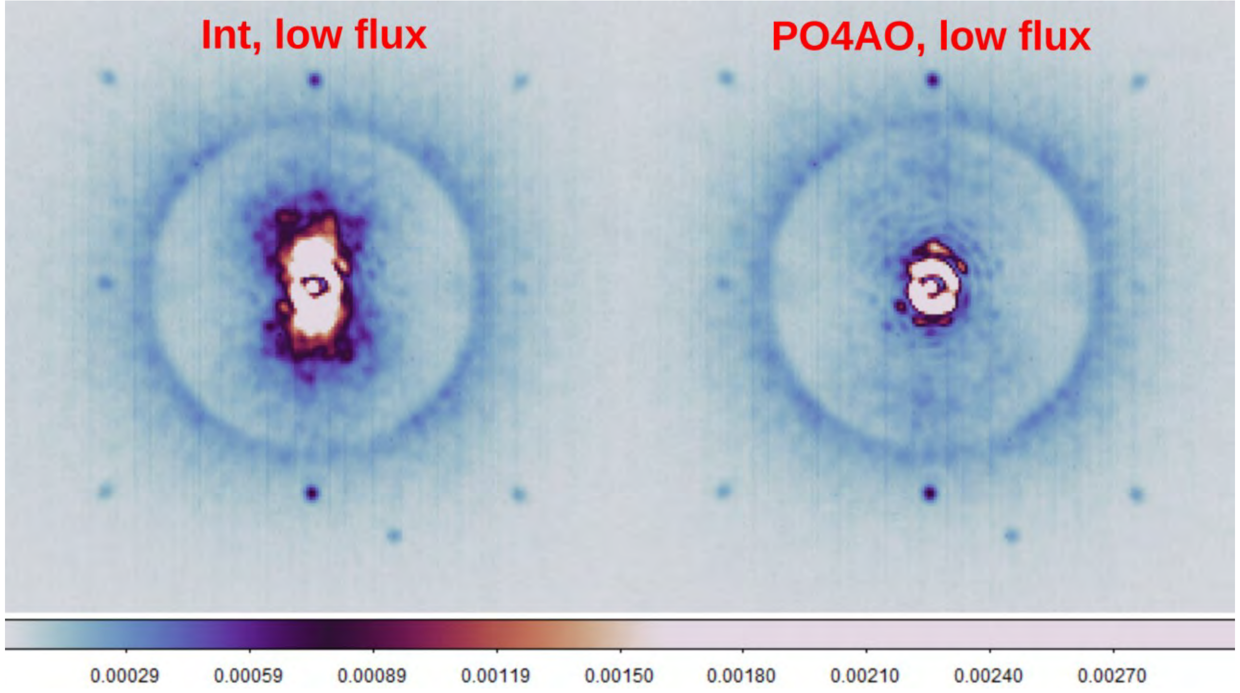


Corresponding contrast



GHOST results:
Nousiainen, J. et al . JATIS submitted in August

Low flux experiment

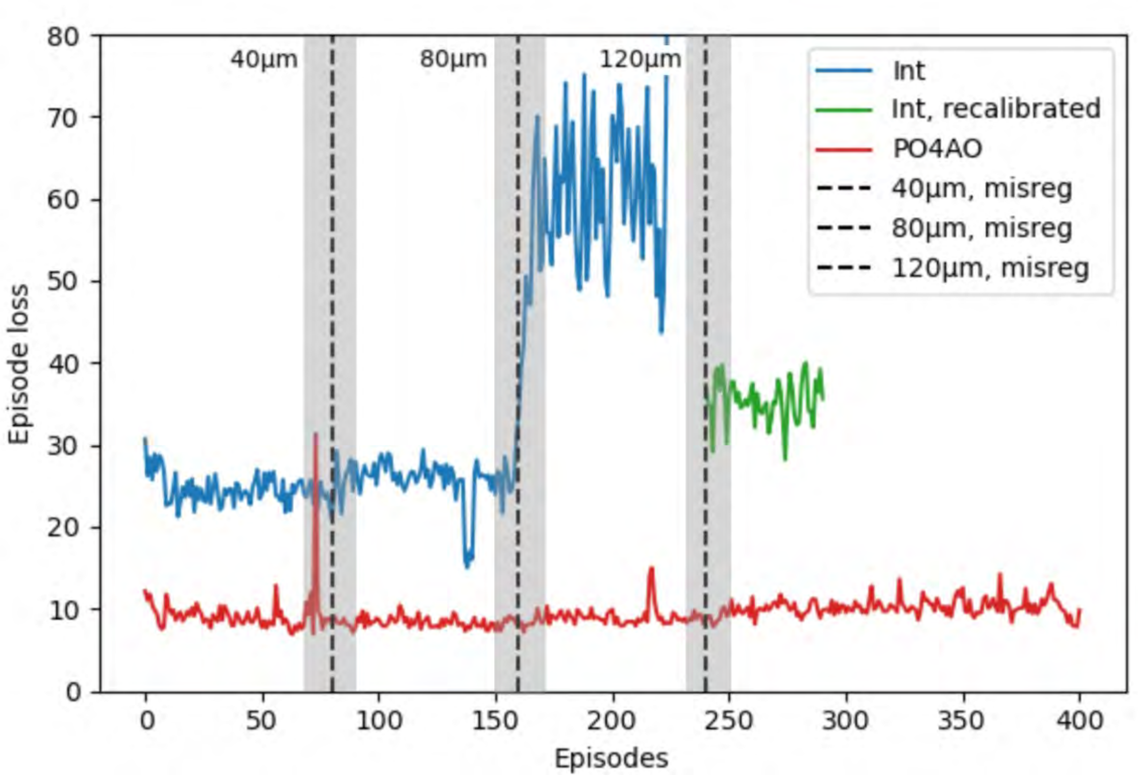


- S/N approximately 1
- The optimal integrator gain was 0.1

GHOST results:
Nousiainen, J. et al . JATIS submitted in August

Open your mind. LUT.
Lappeenranta University of Technology

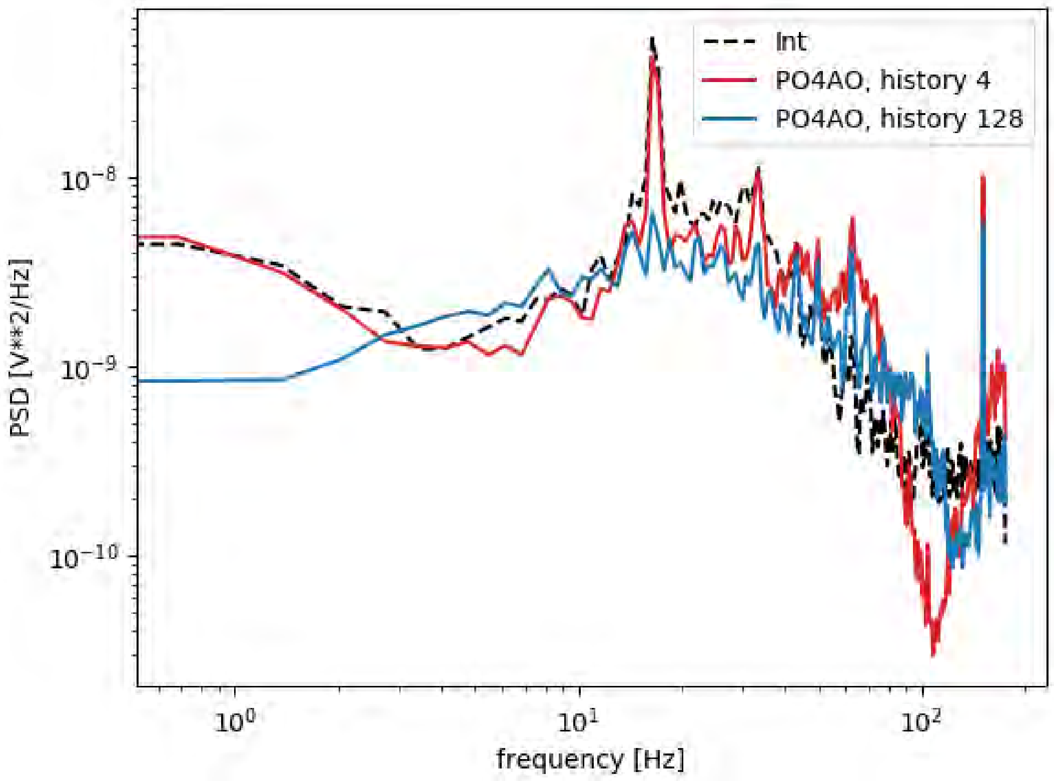
Mis-registration experiment



- We start with Integrator and PO4AO calibrated with centered DM
- While the loop is closed, we start to manually shift the DM off-axis: first 40, then 80 and finally 120 microns (40%).

GHOST results:
Nousiainen, J. et al . JATIS submitted in August

How many history frames do we need?



- We recorded the temporal PSD (KL mode #1) of converged PO4AO on different history lengths
- Short history is enough for atmospheric disturbances but cannot correct low frequency vibrations

GHOST results:
Nousiainen, J. et al . JATIS submitted in August

Conclusion

1. RL gives consistent results in different simulations (numeric and lab)
    - Simulation (1000 DoF, 10k DoF, PWFS), Lab (MagAO-X, GHOST)
2. It is fast to train and to use (training < 10 sec (from scratch), inference << 1 ms)
    - Convolutional NN utilize the spatial structure of turbulence
3. Next step is to go on-sky (SCExAO, MagAO-X)
    1. From Python to tensor RT to reduce latency from 1 ms to 200-400 us (TBC)
4. lots of directions for future work, e.g., NCPA correction and dark hole digging, telescope wavefront control (?)
5. Understanding the physics is essential for designing an effective algorithm

Open your mind. LUT.
Lappeenranta University of Technology

Latency

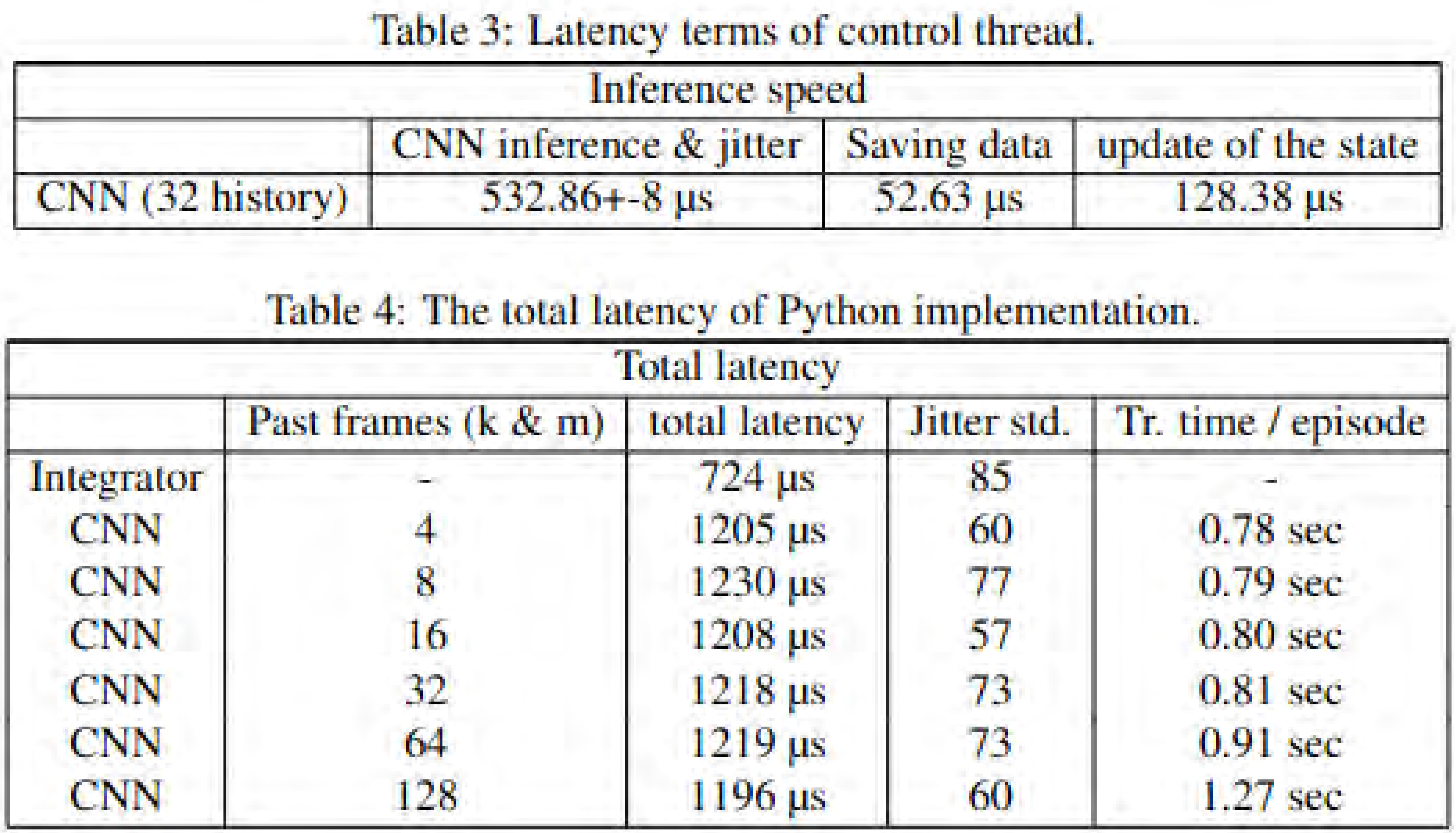


Table 3: Latency terms of control thread.

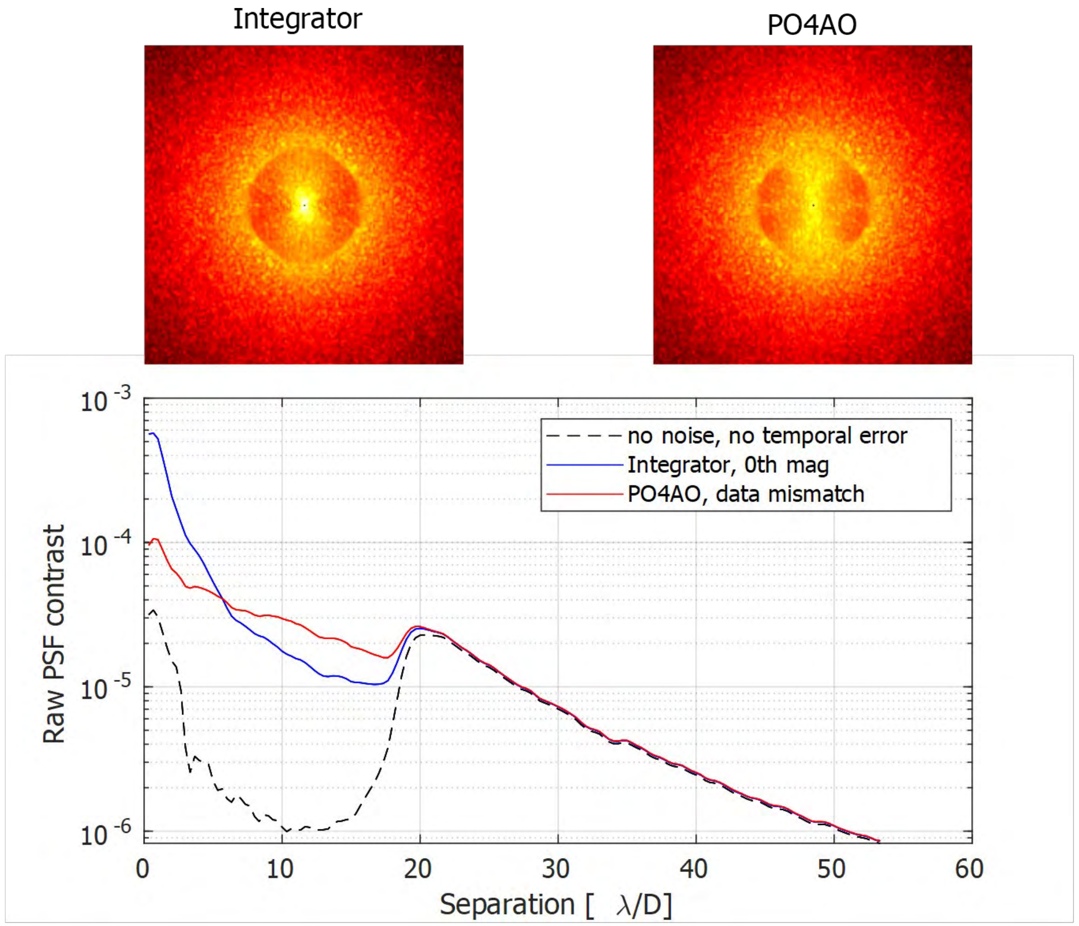
Inference speed			
	CNN inference & jitter	Saving data	update of the state
CNN (32 history)	532.86+-8 µs	52.63 µs	128.38 µs

Table 4: The total latency of Python implementation.

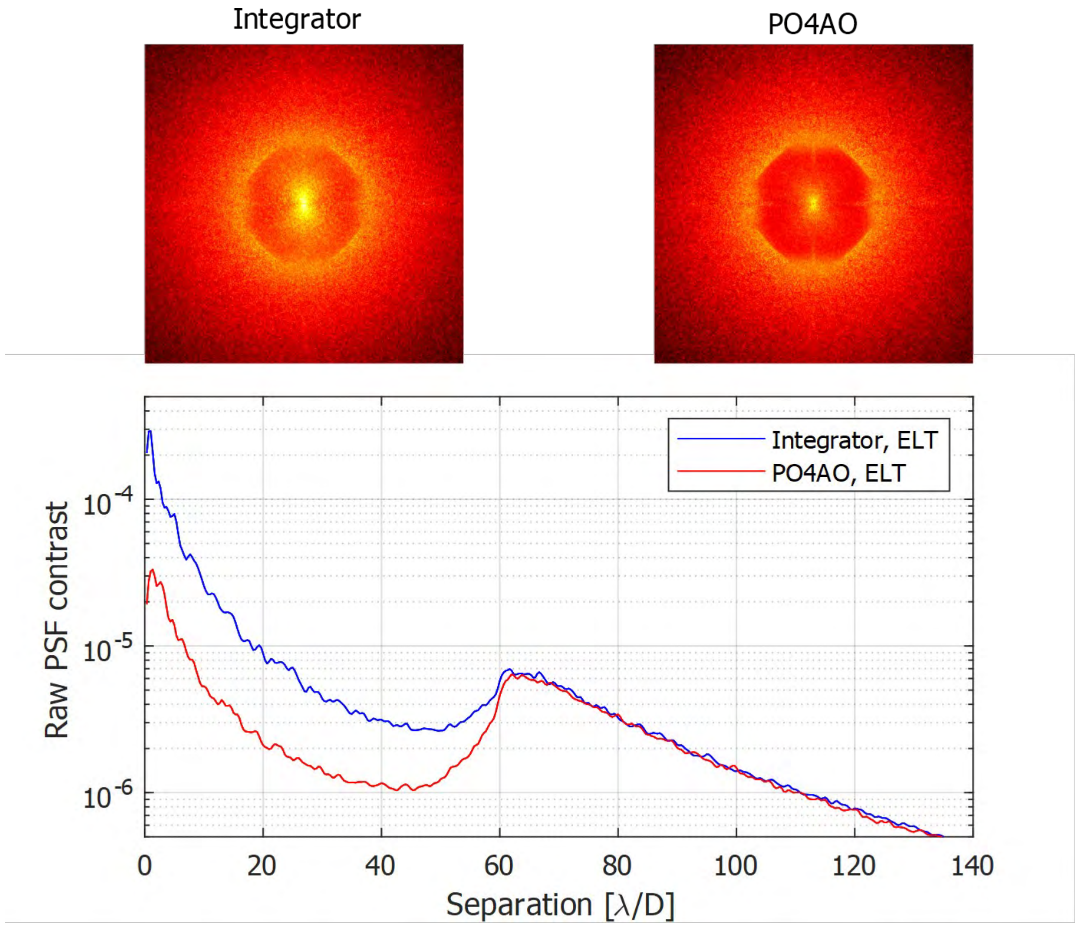
Total latency				
	Past frames (k & m)	total latency	Jitter std.	Tr. time / episode
Integrator	-	724 µs	85	-
CNN	4	1205 µs	60	0.78 sec
CNN	8	1230 µs	77	0.79 sec
CNN	16	1208 µs	57	0.80 sec
CNN	32	1218 µs	73	0.81 sec
CNN	64	1219 µs	73	0.91 sec
CNN	128	1196 µs	60	1.27 sec

Robustness against data mismatch and scalability

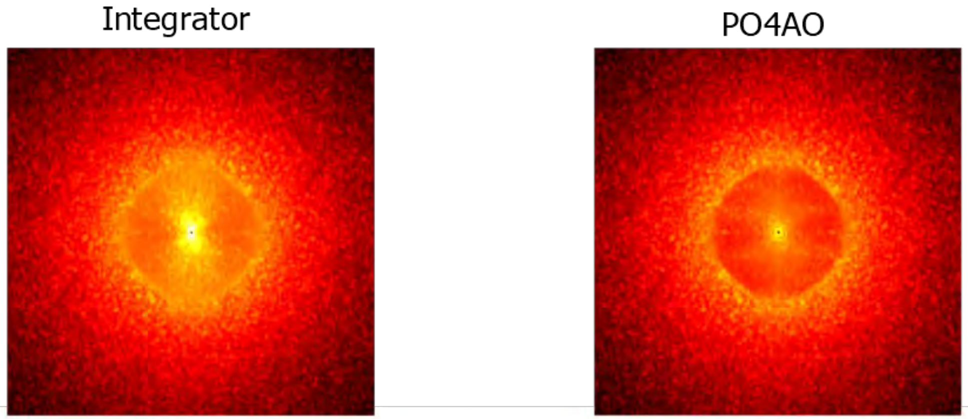
8-meter telescope, data mismatch

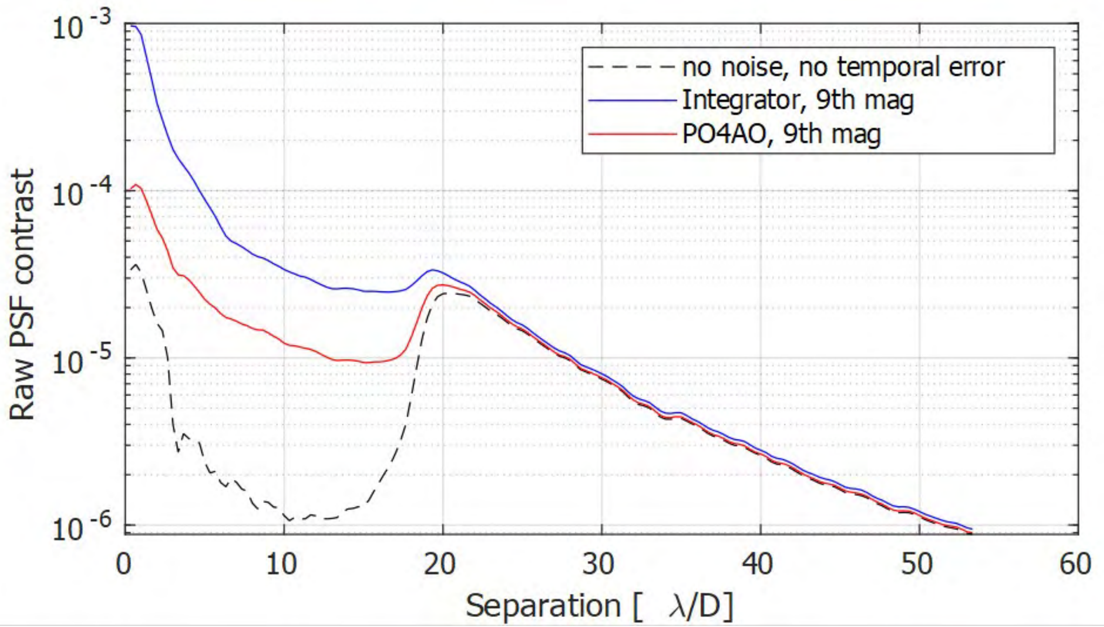


40-meter telescope with ~10000 DoF



Noise robustness





- 8-meter telescope with 40 X 40 actuators, non-modulated PWFS, 9th magnitude guide star
- Takes ~5 sec to beat the integrator and 20- 30 sec to fully converge