Do Androids Dream of Electric Fences?

Safety-Aware Reinforcement Learning with Latent Shielding

Joint work with

Chloe He

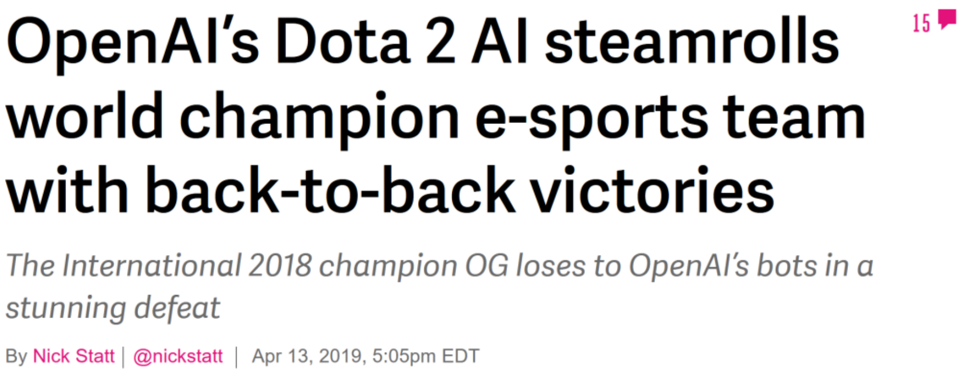
University College London

Borja Gonzalez Leon

Imperial College London

Best Paper award at SafeAI@AAAI2022

Reinforcement Learning is HOT

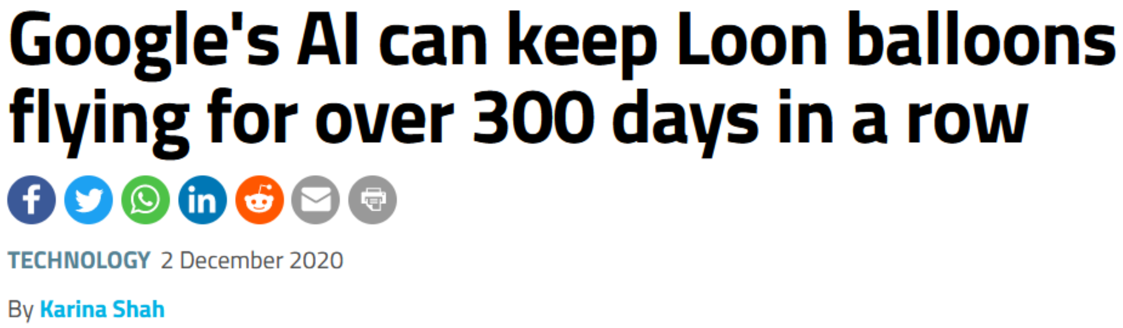


OpenAI's Dota 2 AI steamrolls world champion e-sports team with back-to-back victories

15

The International 2018 champion OG loses to OpenAI's bots in a stunning defeat

By Nick Statt | @nickstatt | Apr 13, 2019, 5:05pm EDT

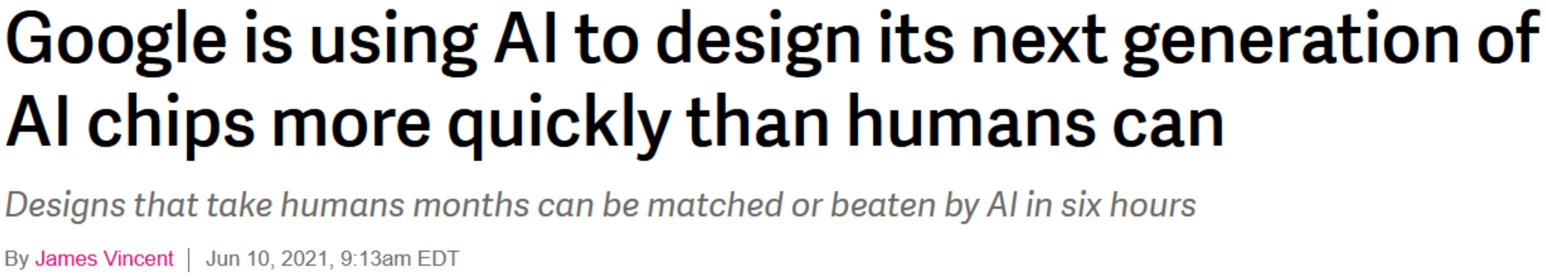


Google's AI can keep Loon balloons flying for over 300 days in a row

TECHNOLOGY 2 December 2020

By Karina Shah

Ed Johns: *"Powerful DRL comes up with solutions which are better than those from even the brightest human engineers."*



Google is using AI to design its next generation of AI chips more quickly than humans can

Designs that take humans months can be matched or beaten by AI in six hours

By James Vincent | Jun 10, 2021, 9:13am EDT

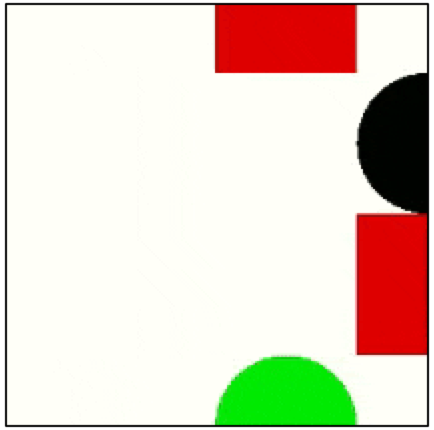
But so is Safety

Sources (left to right): ESA / Getty Images / AP

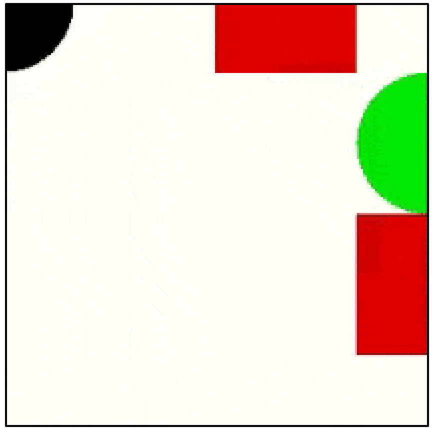
Coming Up In Today's Presentation...

- A short ride through the landscape of Safe RL
- Imagination-based agents for Safe RL
- A whizz through our interesting findings
- A peek into the future

Coming Up In Today's Presentation...



SOTA (Unsafe) RL Agent



Our Method

D. Hafner, et al. *Dream to Control: Learning Behaviors by Latent Imagination*. ICLR 2020.
D. Hafner, et al. *Mastering Atari with Discrete World Models*. ICLR 2021.

Safety

“Bad things shouldn’t happen”

Safety

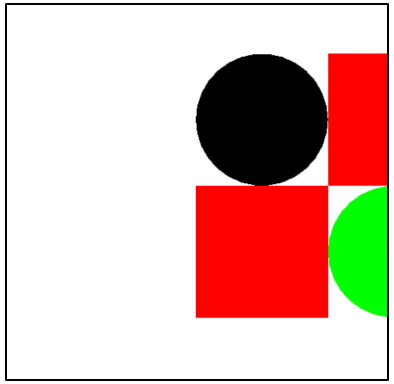
“Bad things shouldn’t happen”

$$\phi, \phi' ::= true \mid p \mid \neg\phi \mid \phi \wedge \phi' \mid \bigcirc\phi \mid \phi \cup \phi'$$

LTL as a convenient framework for temporal (safety/reachability) properties.

Safety

$$\phi, \phi' ::= true \mid p \mid \neg\phi \mid \phi \wedge \phi' \mid \bigcirc\phi \mid \phi \cup \phi'$$

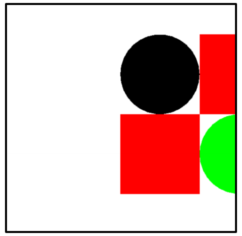


$\neg\texttt{agent\_in\_red\_square} \cup \texttt{episode\_ended}$

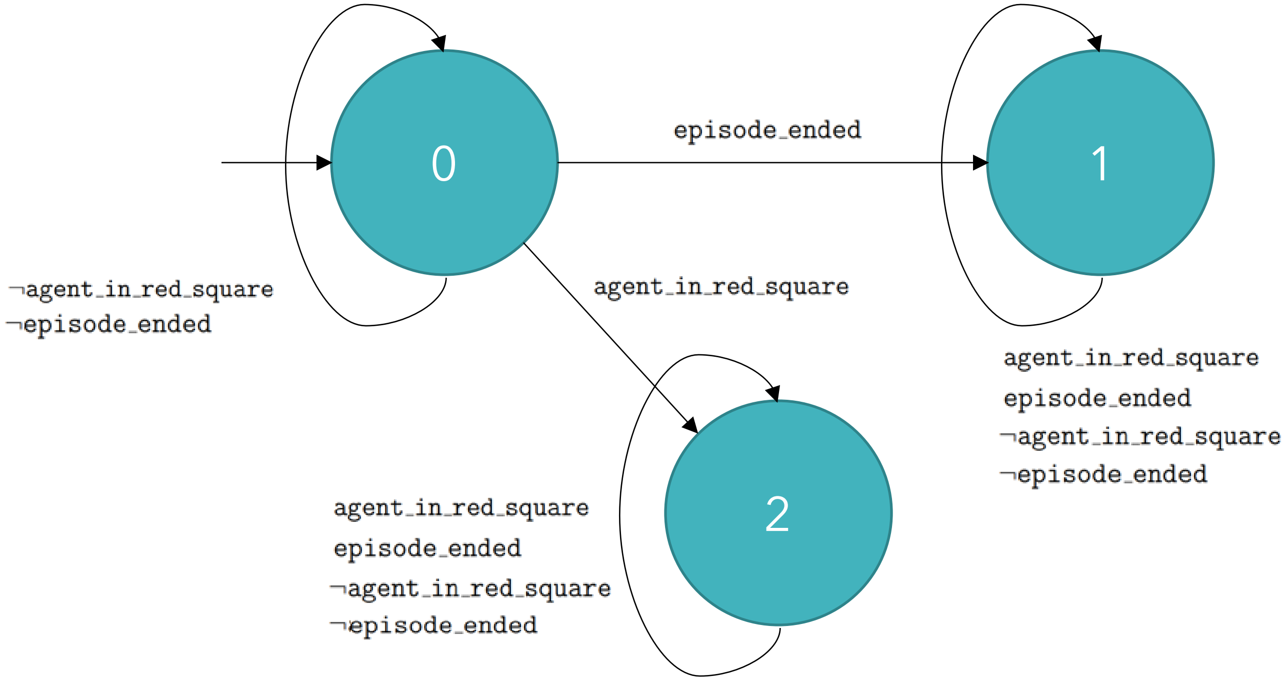
Verifying Safety

- We encode the safety constraint on the environment as some propositional formula Φ.
- **Goal:** find a policy π that maximises expected reward, while minimizing violations of the safety constraint Φ during training.

Safety

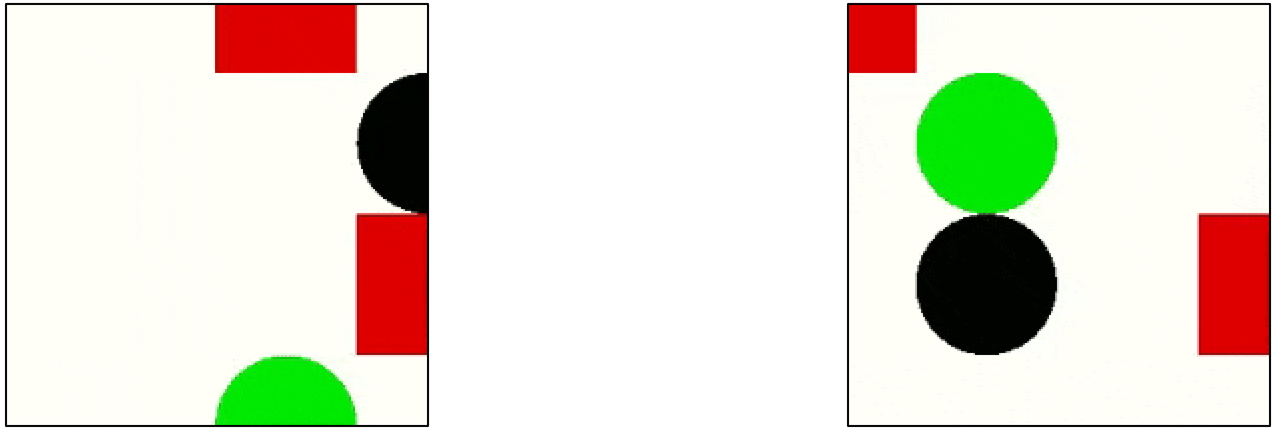


¬agent_in_red_square ∪ episode_ended

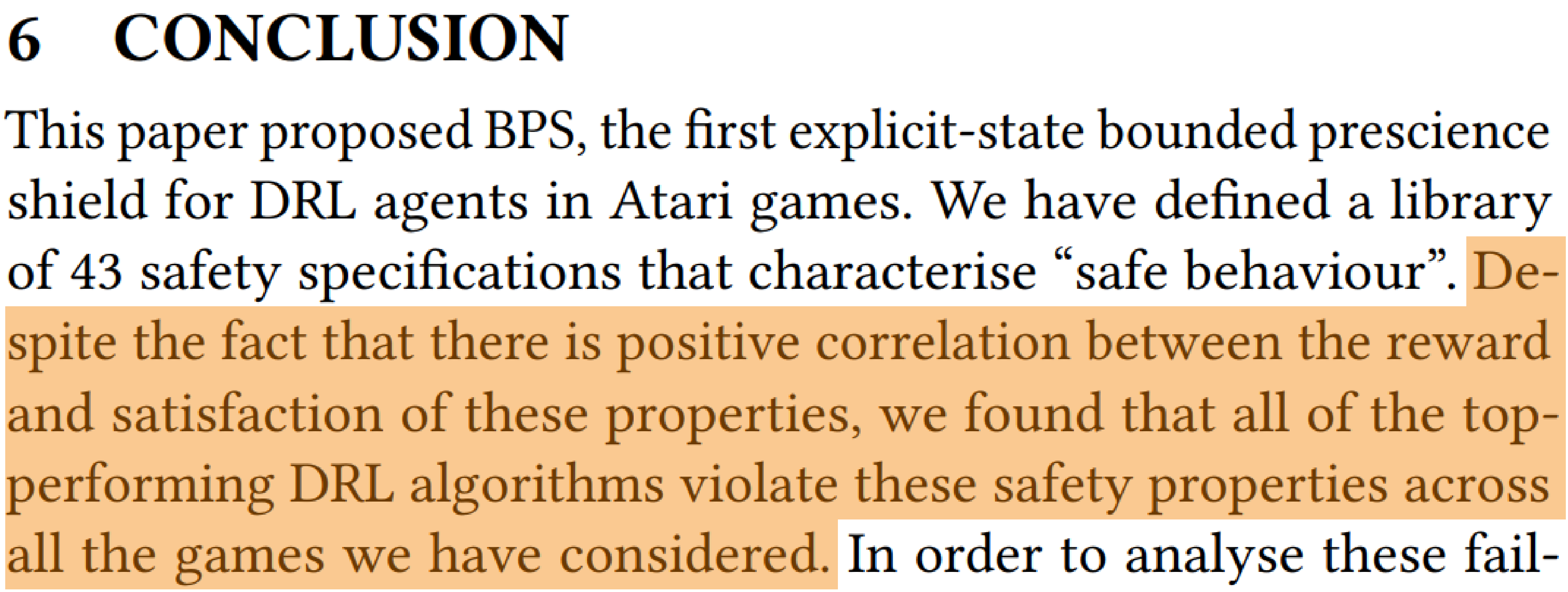


The labelling $\lambda\phi : S \to \{\mathit{safe}, \mathit{unsafe}\}$ can be synthesized effectively.

$$\neg\texttt{agent\_in\_red\_square} \cup \texttt{episode\_ended}$$



D. Hafner, et al. *Dream to Control: Learning Behaviors by Latent Imagination*. ICLR 2020.
D. Hafner, et al. *Mastering Atari with Discrete World Models*. ICLR 2021.



6 CONCLUSION

This paper proposed BPS, the first explicit-state bounded prescience shield for DRL agents in Atari games. We have defined a library of 43 safety specifications that characterise “safe behaviour”. Despite the fact that there is positive correlation between the reward and satisfaction of these properties, we found that all of the top-performing DRL algorithms violate these safety properties across all the games we have considered. In order to analyse these fail-

M. Giacobbe et al. *Shielding Atari Games with Bounded Prescience*. AAMAS 2021.

LOTS of work into Safe RL

- Constrained Policy Optimisation [Achiam at al., 2017]
- Curriculum Learning/Learning from Demonstration [Turchetta et al., 2020]
- Safety Critics [Srinivasan et al. 2020]
- Symbolic Policy Verification [Fulton et al., 2018]
- Reward Shaping [Toro Icarte et al., 2018]
- Shielding [Alshiek et al., 2018]

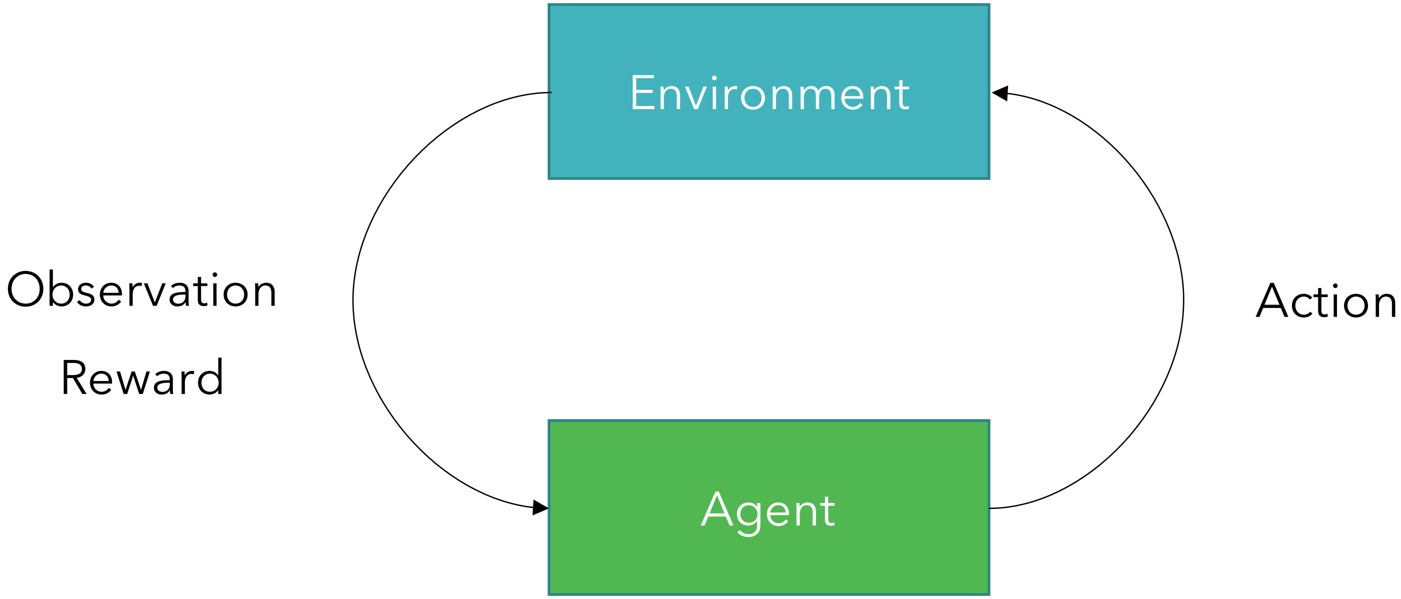
LOTS of work into Safe RL

- Constrained Policy Optimisation [Achiam at al., 2017]
- Curriculum Learning/Learning from Demonstration [Turchetta et al., 2020]
- Safety Critics [Srinivasan et al. 2020]
- Symbolic Policy Verification [Fulton et al., 2018]
- Reward Shaping [Toro Icarte et al., 2018]
- Shielding [Alshiek et al., 2018]

} inspired by Formal Methods

"Safe RL is the process of learning an optimal policy while satisfying a temporal logic safety specification during the learning and execution phases".

Reinforcement Learning



Environment

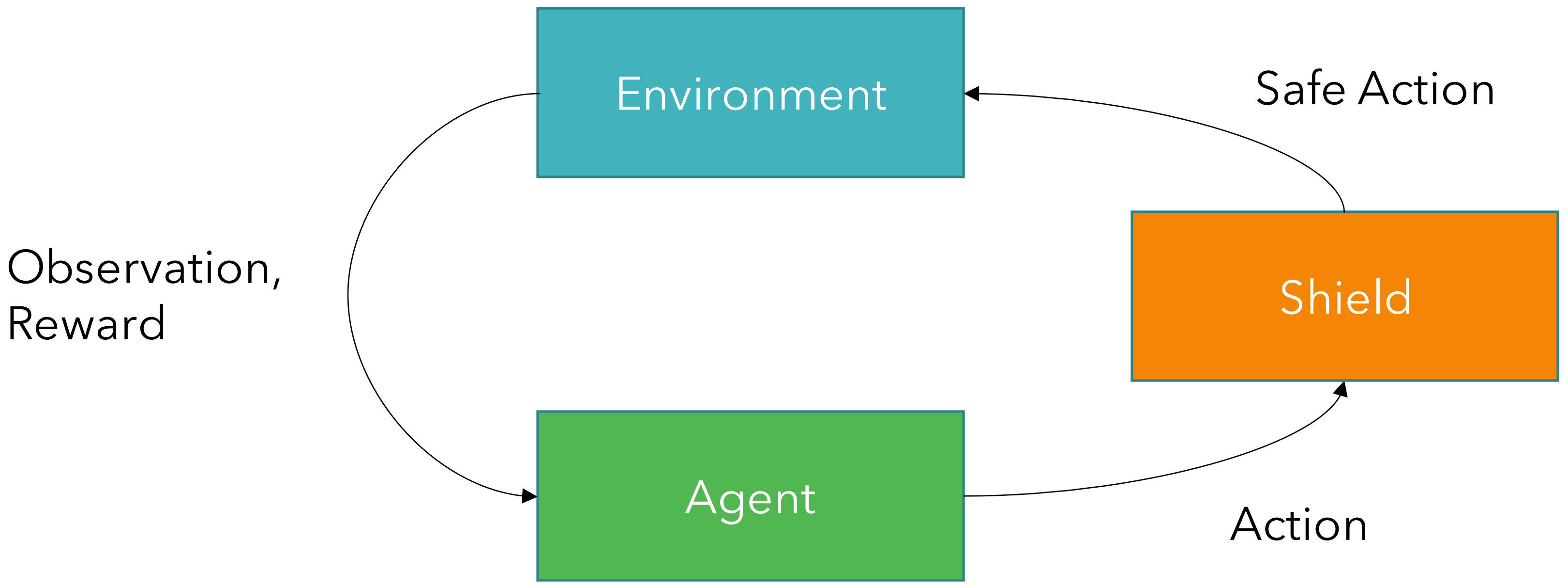
Observation

Reward

Action

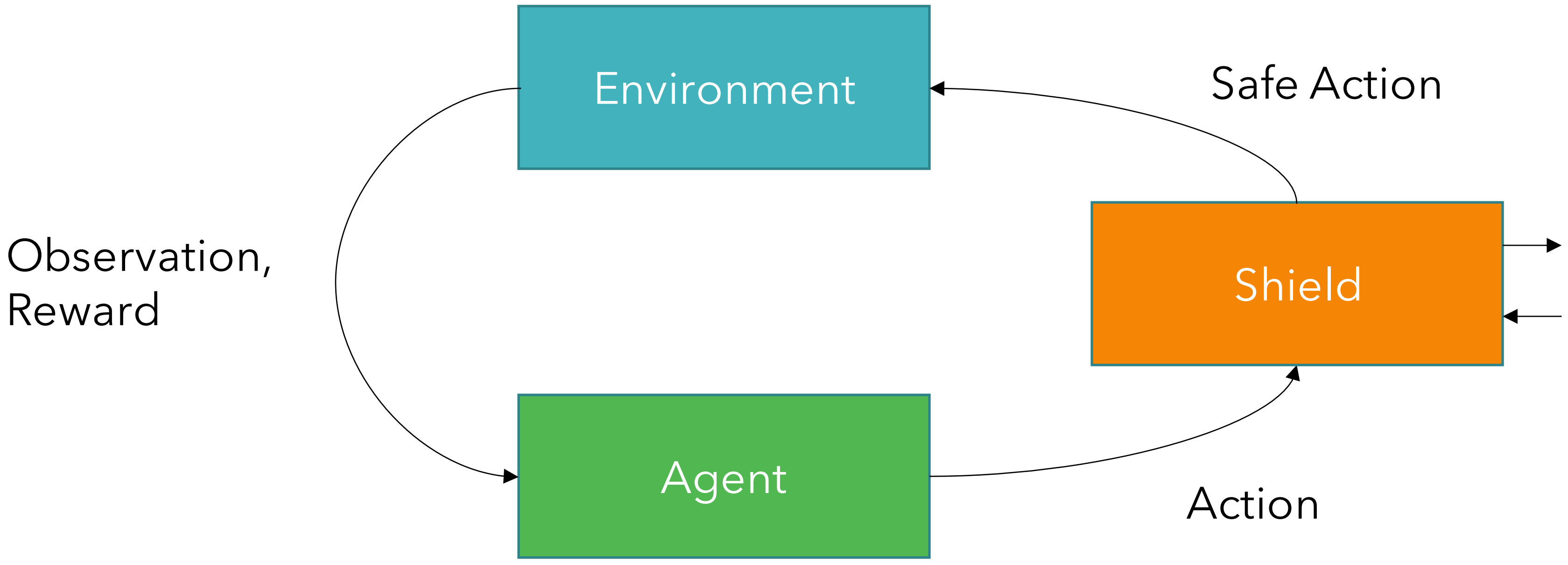
Agent

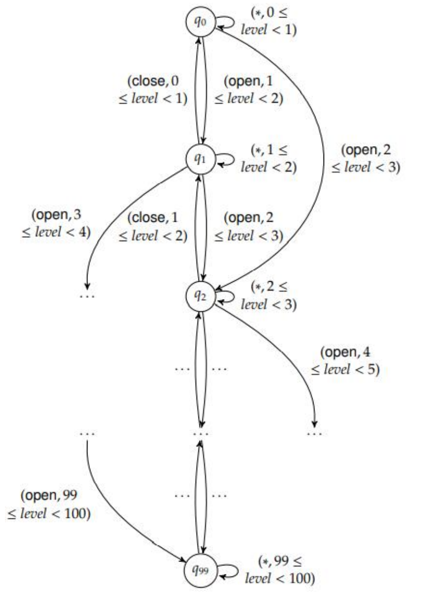
Shielding



M. Alshiekh et al., *Safe Reinforcement Learning via Shielding*. AAAI 2018

Shielding





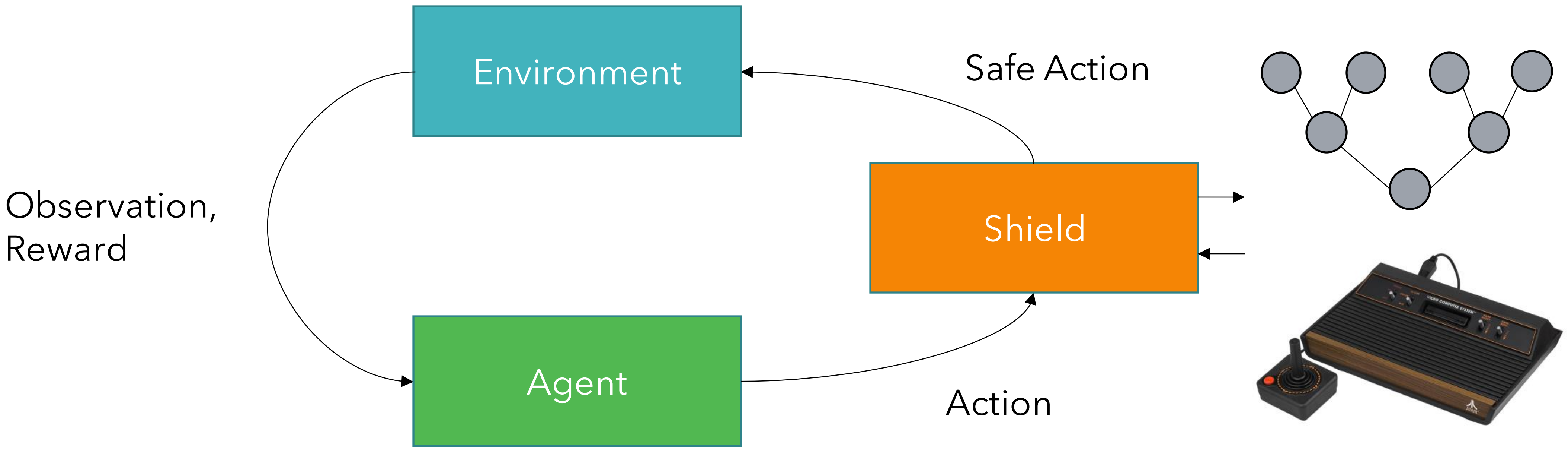
$$\begin{aligned}&\mathsf{G}(level > 0)\\ \wedge\ &\mathsf{G}(level < 100)\\ \wedge\ &\mathsf{G}((open \wedge \mathsf{X}close) \rightarrow \mathsf{XX}close \wedge \mathsf{XXX}close)\\ \wedge\ &\mathsf{G}((close \wedge \mathsf{X}open) \rightarrow \mathsf{XX}open \wedge \mathsf{XXX}open)\end{aligned}$$

+ agnostic wrt the RL algorithm

- requires a model of the environment

M. Alshiekh et al., *Safe Reinforcement Learning via Shielding*. AAAI 2018

Bounded Prescience Shielding


Environment

Safe Action

Shield

Observation,
Reward

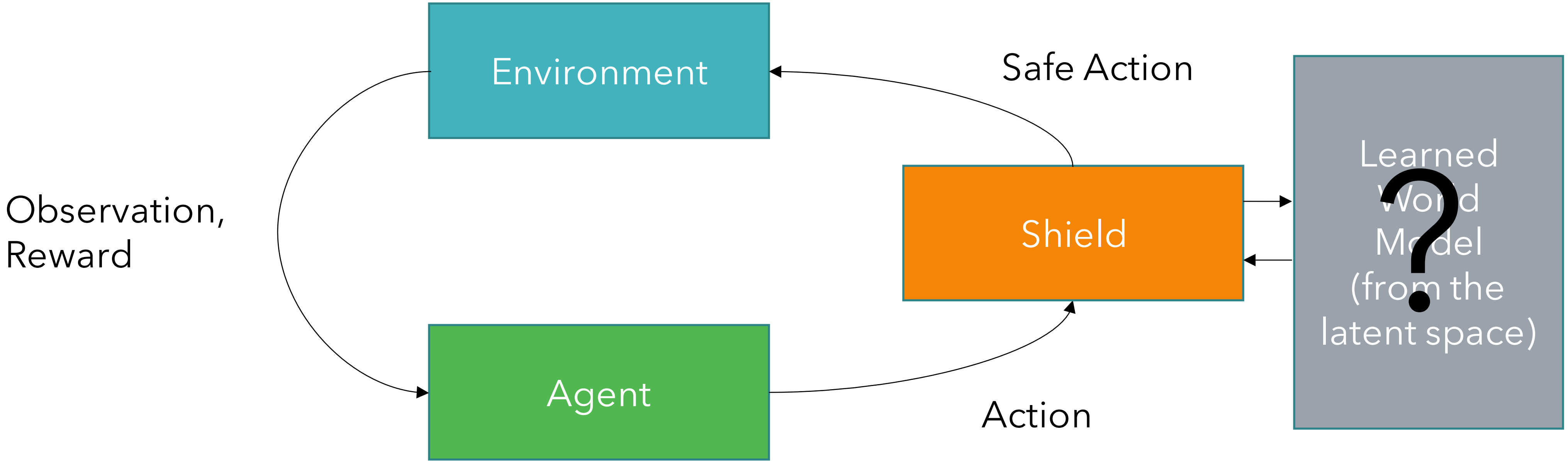
Agent

Action

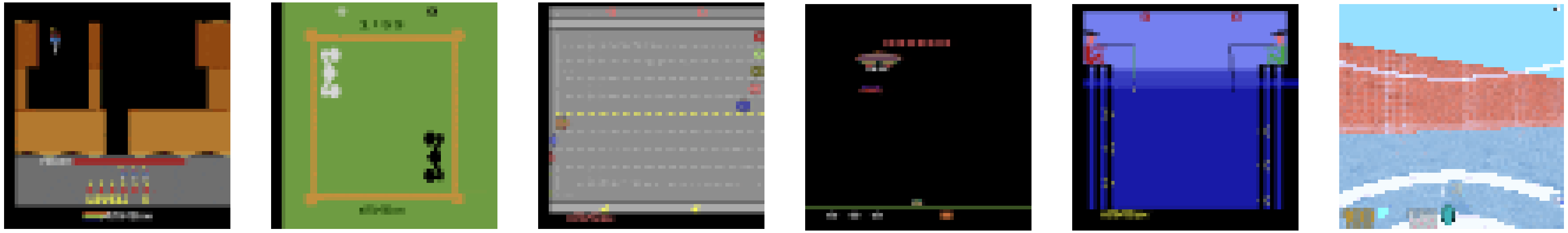
+ does not require an abstraction
- but does require an emulator (Stella for ALE)

M. Giacobbe et al., *Shielding Atari Games with Bounded Prescience*. AAMAS 2021.

Our Approach



Model-Based RL to the Rescue!



1. Learn a world model from the latent space
2. Learn a policy inside the world model
3. Collect data in the real environment using the learned policy
4. Use the model of the environment to keep the agent safe
5. Repeat until convergence

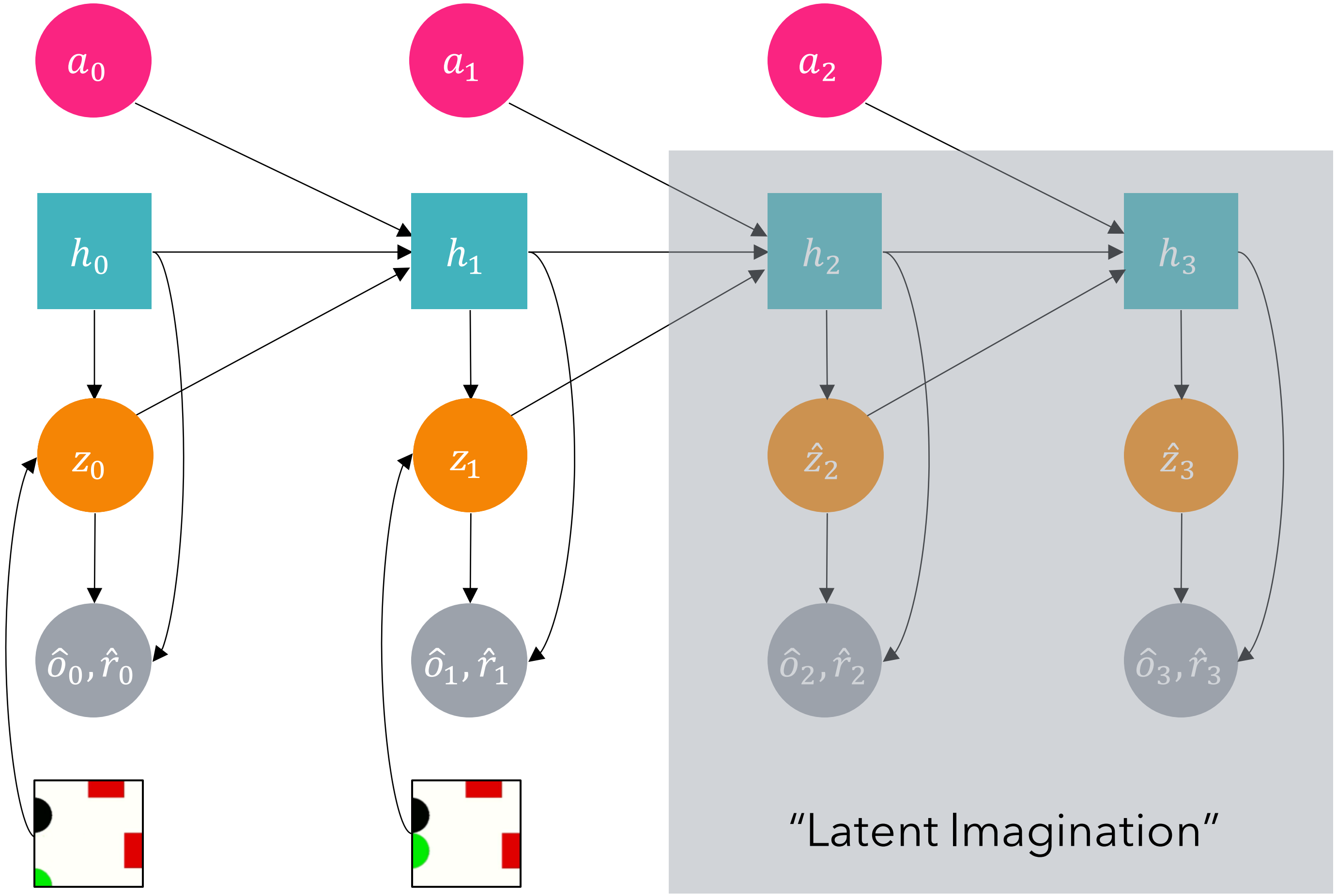
New!

D. Hafner, et al. *Mastering Atari with Discrete World Models*. 2021.
C. He, B. G. Leon & F. Belardinelli, *Do Androids Dream of Electric Fences?* SafeAI@AAAI2022.

World Models

- Predictive models of an environment maintained by the model-based agent.
- Learnt from experience.
- Used
  - as a substitute for the environment during training [Ha et al. 2018; Hafner et al. 2021]
  - for approximate shielding.
- We use **recurrent state-space models** (RSSM) [Hafner et al. 2019b].
- 3 key components:
  - dynamics model (recurrent, representation, and transition models)
  - reward model
  - observation model

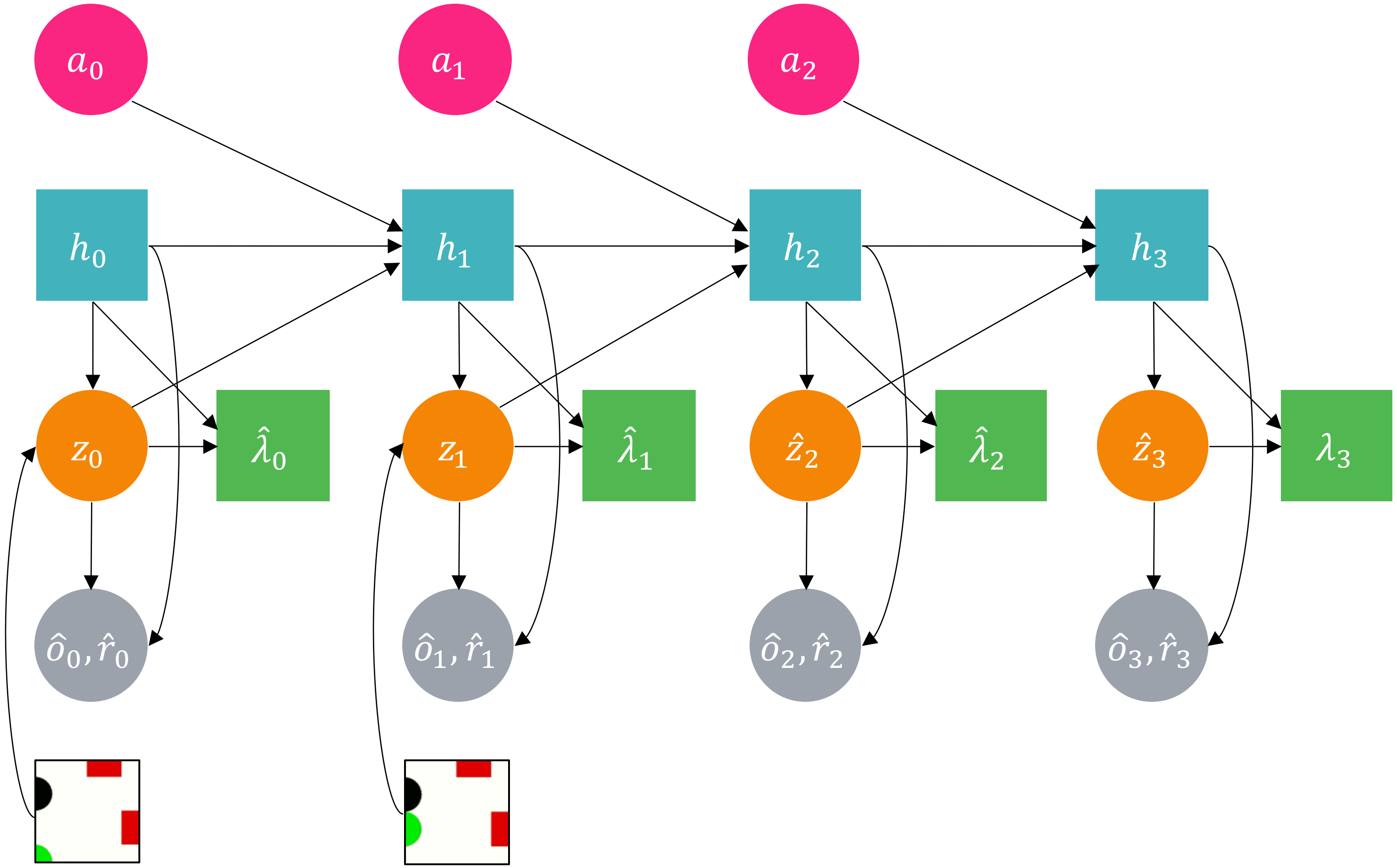
Learning World Models



Recurrent **S**tate **S**pace **M**odel

$$h_t = f(h_{t-1}, z_{t-1}, a_{t-1})$$
$$z_t \sim q(z_t | h_t, o_t)$$
$$\hat{z}_t \sim p(\hat{z}_t | h_t)$$
$$\hat{o}_t \sim p(\hat{o}_t | h_t, z_t)$$
$$\hat{r}_t \sim p(\hat{r}_t | h_t, z_t)$$

D. Hafner et al. *Learning Latent Dynamics for Planning from Pixels*. ICML 2019.

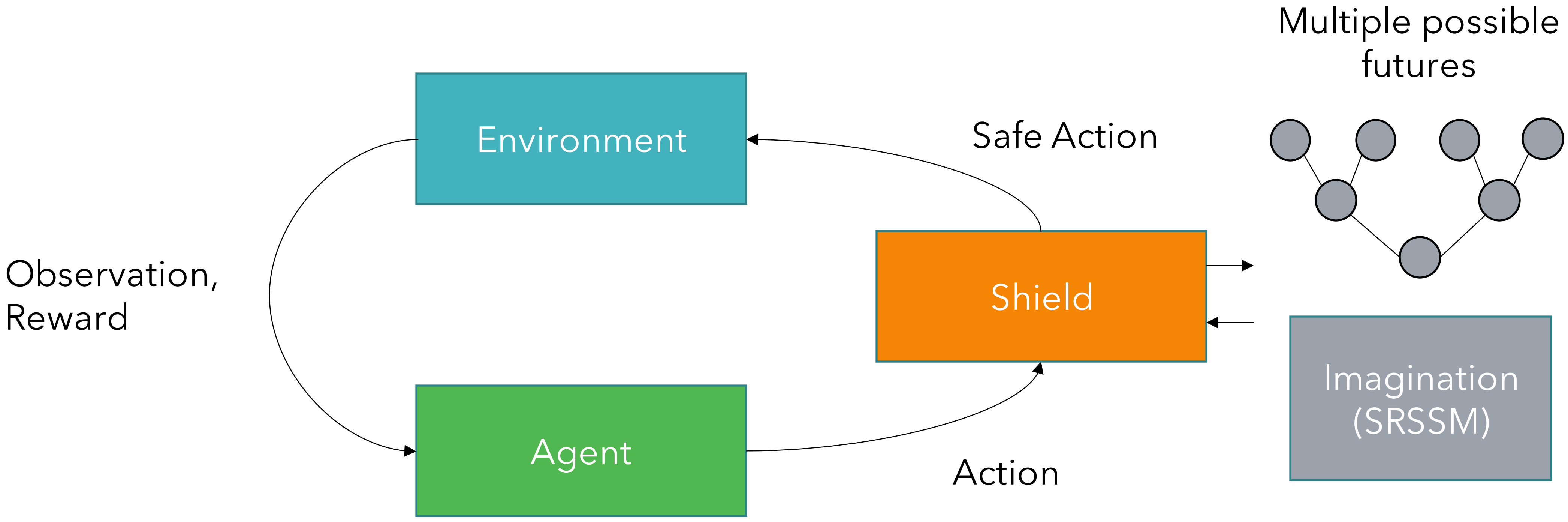


New!

Safety
Recurrent
State
Space
Model

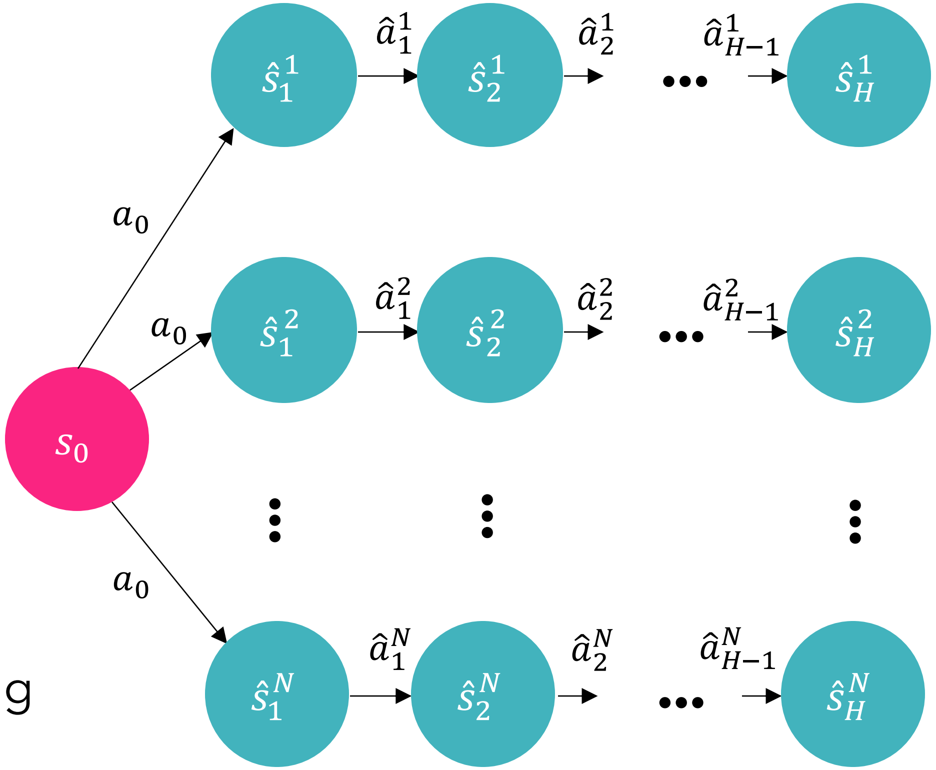
Violation prediction \hat{λ}ϕ: *S* → {*safe*, *unsafe*}

Latent Shielding



Approximate Bounded Prescience

New!



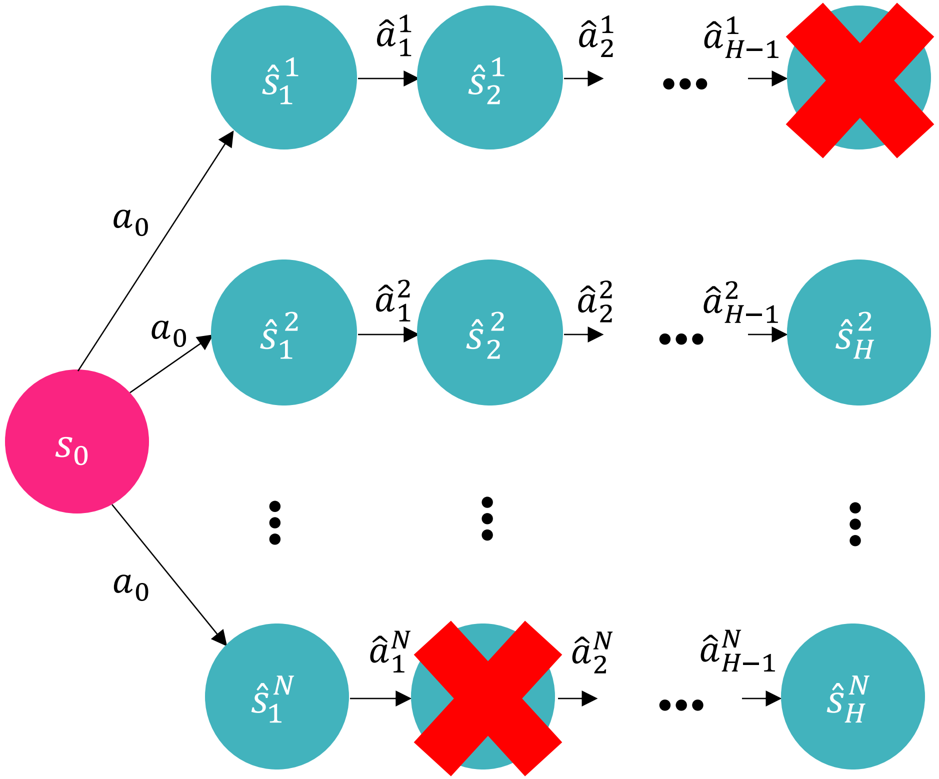
$$\hat{a}_h^n = \pi(\hat{s}_{h-1}^n) + \boxed{\eta}$$

Noise Term

Differences wrt BPS:

1. We approximate the labelling function through SRSSM
2. We sample a fixed number of future trajectories

ABP Shielding for Latent Trajectories



$$\hat{a}_h^n = \pi(\hat{s}_{h-1}^n) + \boxed{\eta}$$

Noise Term

ABP Shielding for Latent Trajectories

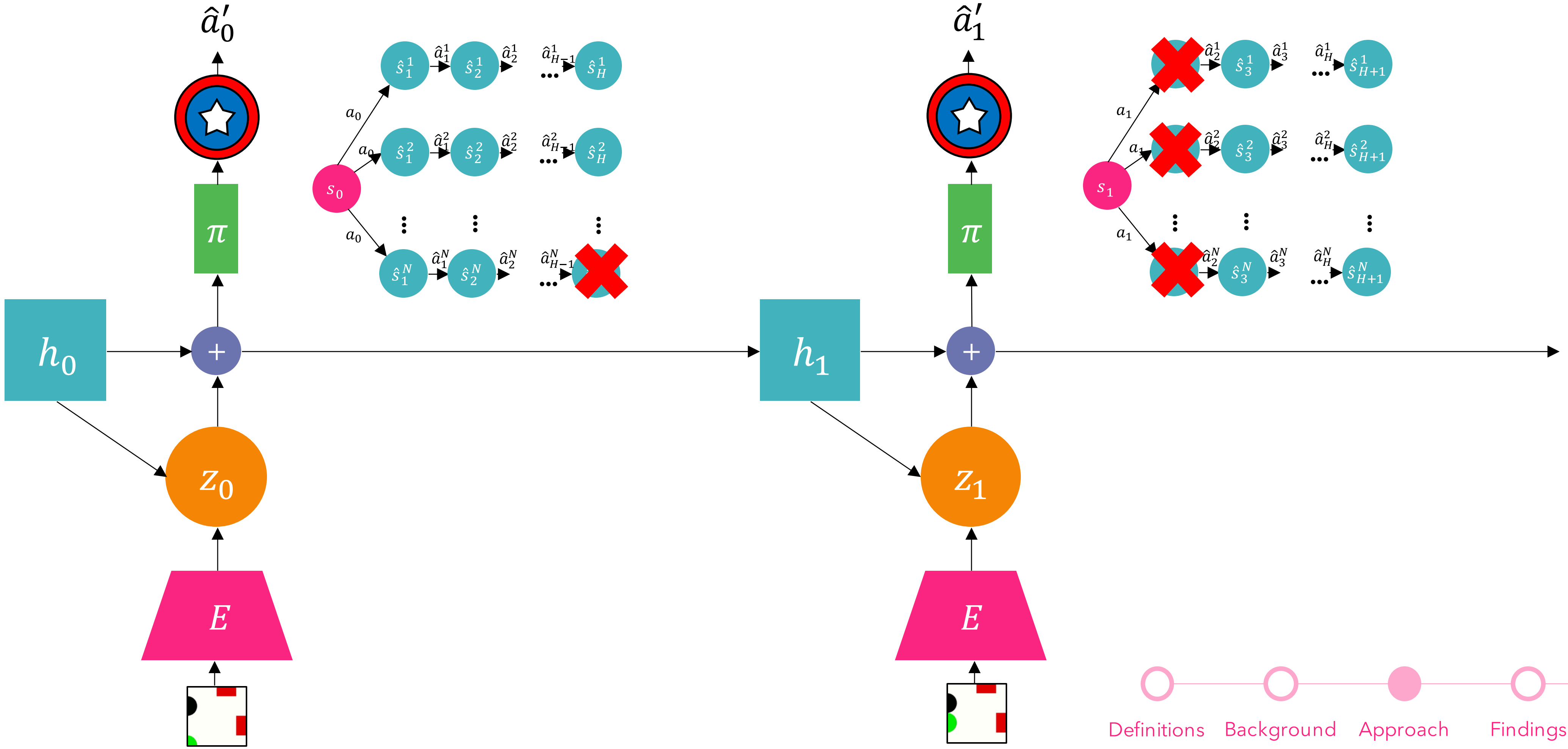
$$\pi'(s_t) = \begin{cases} \pi(s_t), & \text{if } P(s_{t+1}\ \mathit{is\ a\ violation} \mid \pi(s_t)) < \epsilon \\ \varsigma(s_t), & \text{otherwise} \end{cases}$$

Safety Threshold

Safe Alternative Policy

New!

ABP Shielding for Latent Trajectories



Training an Agent with Latent Shielding

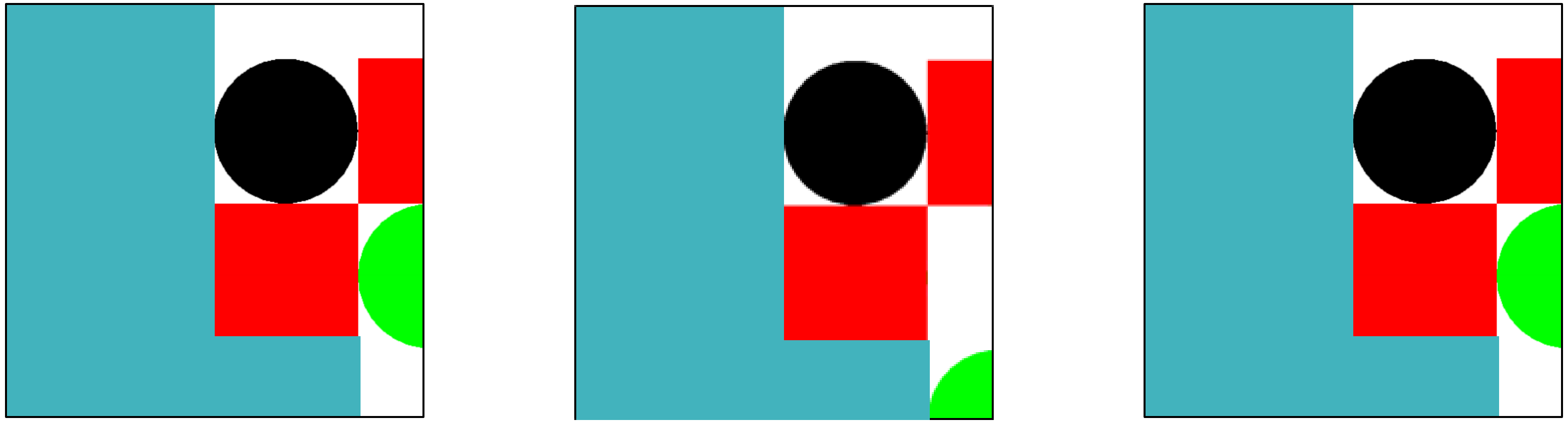
New!

1. Learn a **SRSSM** model of the environment.
2. Learn the policy using the model of the environment:
   - the agent imagines trajectories with actions chosen from its current policy.
   - the unshielded policy is updated **assigning a punishment for violations.**
3. Collect data in the real environment using the learned **shield policy.**
4. Repeat until convergence.

But It’s Not All Fun and Games...

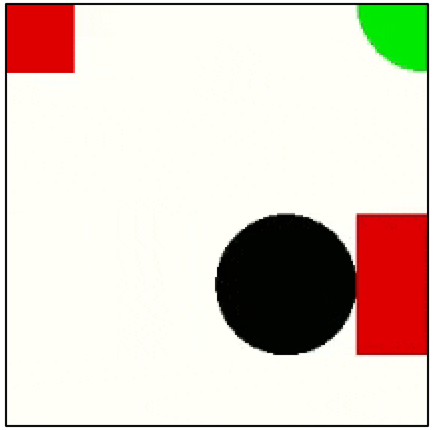
An inaccurate internal model of the environment can lead to the latent shield hindering exploration!

But It’s Not All Fun and Games…



An inaccurate internal model of the environment can lead to the latent shield hindering exploration

But It’s Not All Fun and Games...



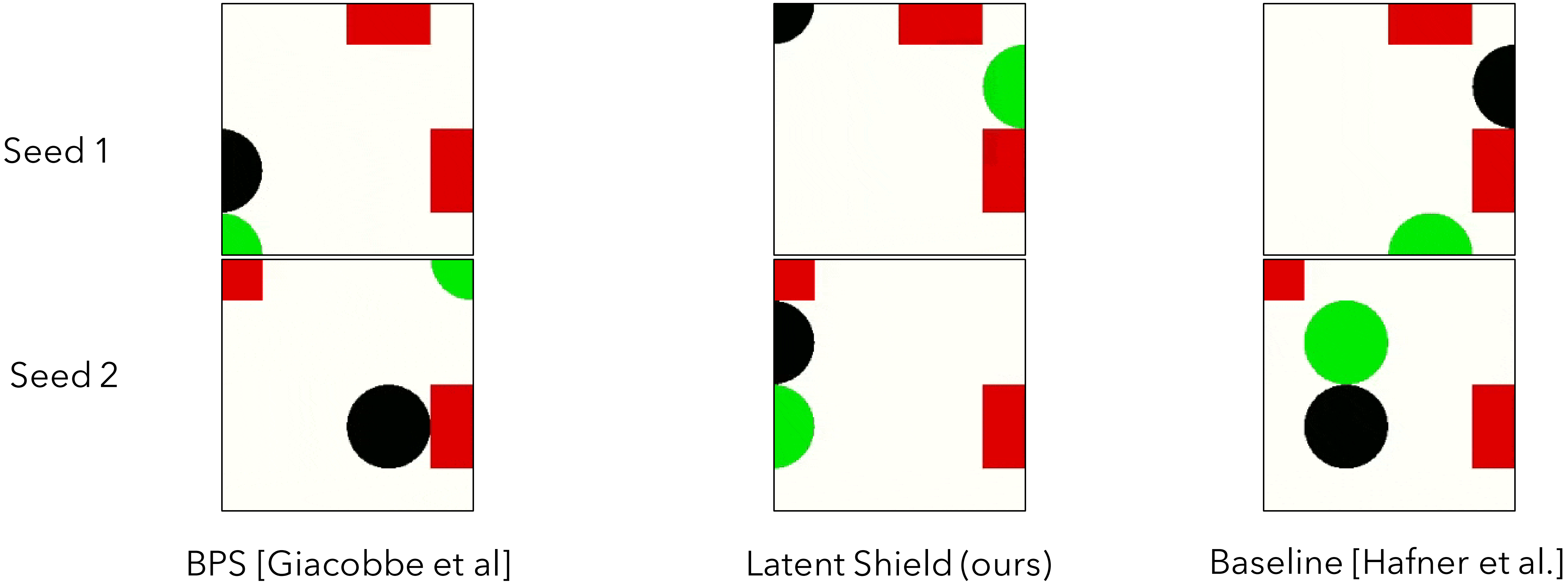
In fact, even classical shielding [Alshiekh et al., 2018] can hinder exploration

Shield Introduction Schedules

New!

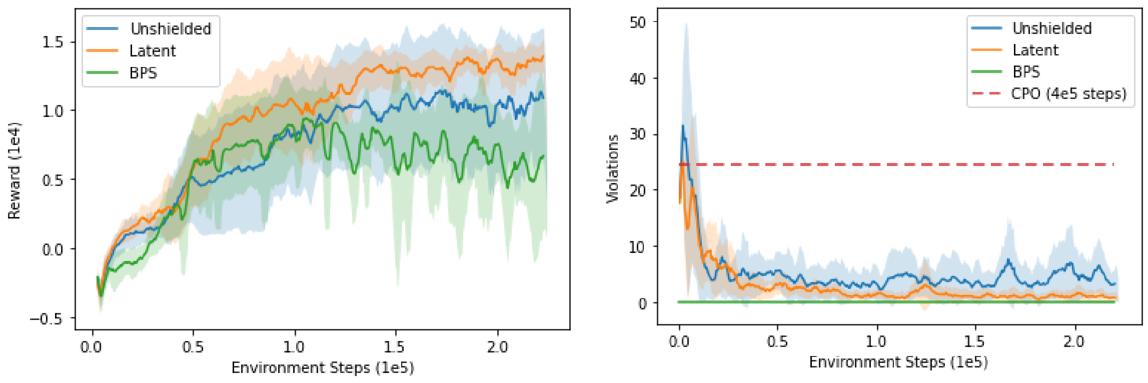
- A gradually decaying probability of disabling the shield with respect to time.
- Enabling shielding after a certain number of training episodes have been completed.

Performance Evaluation

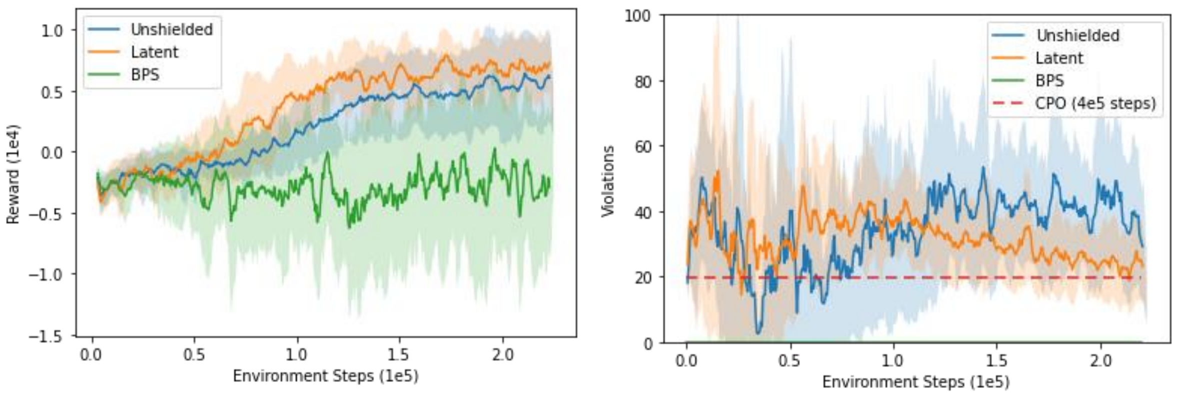


M. Giacobbe et al. *Shielding Atari Games with Bounded Prescience*. 2021.
D. Hafner, et al. *Dream to Control: Learning Behaviors by Latent Imagination*. 2020.
D. Hafner, et al. *Mastering Atari with Discrete World Models*. 2021.

Performance Evaluation



Fixed Gridworld

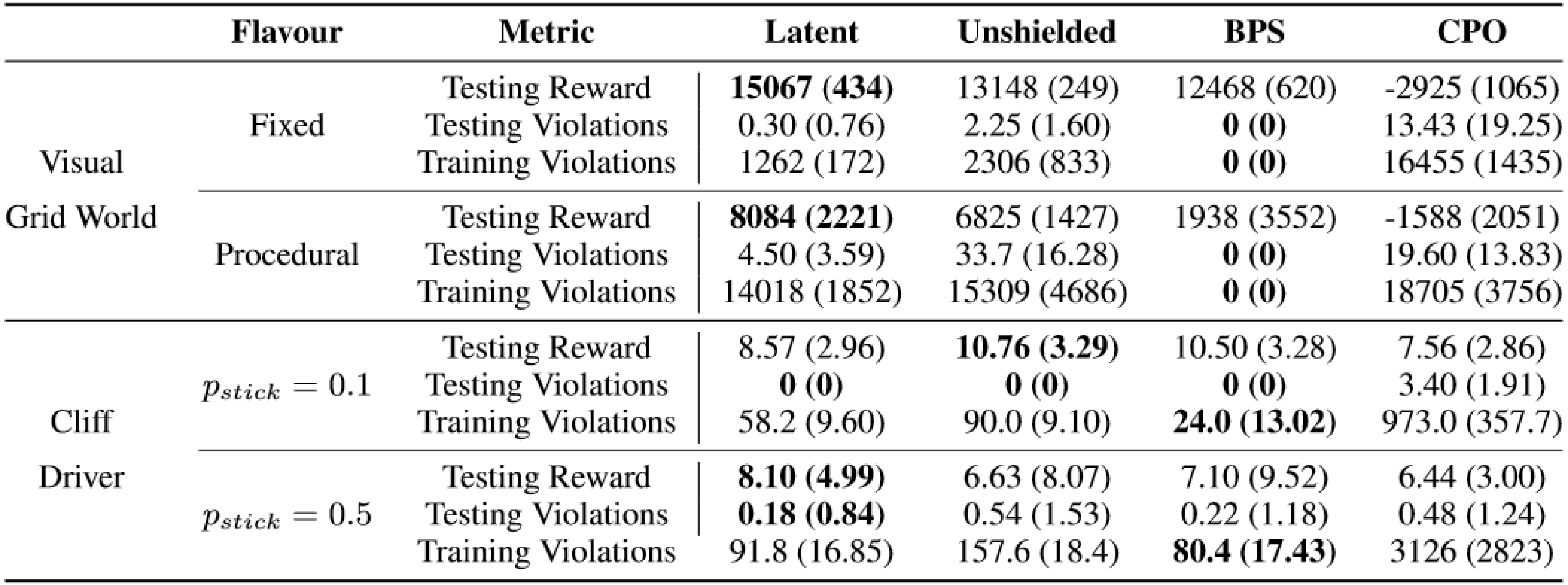


Procedurally Generated Gridworld

(see paper for MORE graphs)

M. Giacobbe et al. *Shielding Atari Games with Bounded Prescience.* 2021.
D. Hafner, et al. *Dream to Control: Learning Behaviors by Latent Imagination*. 2020.
D. Hafner, et al. *Mastering Atari with Discrete World Models*. 2021.
J. Achiam, et al. Constrained Policy Optimization. 2017.

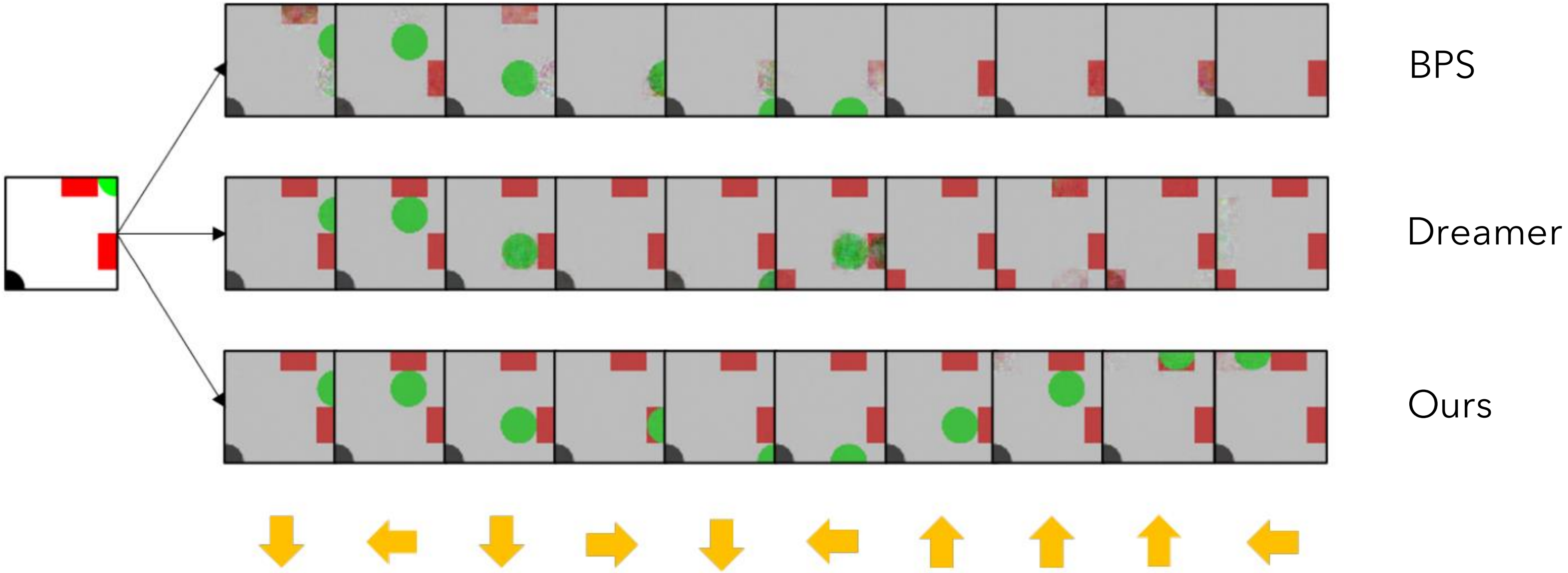
Performance Evaluation



	Flavour	Metric	Latent	Unshielded	BPS	CPO
Visual Grid World	Fixed	Testing Reward	**15067 (434)**	13148 (249)	12468 (620)	-2925 (1065)
		Testing Violations	0.30 (0.76)	2.25 (1.60)	**0 (0)**	13.43 (19.25)
		Training Violations	1262 (172)	2306 (833)	**0 (0)**	16455 (1435)
	Procedural	Testing Reward	**8084 (2221)**	6825 (1427)	1938 (3552)	-1588 (2051)
		Testing Violations	4.50 (3.59)	33.7 (16.28)	**0 (0)**	19.60 (13.83)
		Training Violations	14018 (1852)	15309 (4686)	**0 (0)**	18705 (3756)
Cliff Driver	$p_{stick} = 0.1$	Testing Reward	8.57 (2.96)	**10.76 (3.29)**	10.50 (3.28)	7.56 (2.86)
		Testing Violations	**0 (0)**	**0 (0)**	**0 (0)**	3.40 (1.91)
		Training Violations	58.2 (9.60)	90.0 (9.10)	**24.0 (13.02)**	973.0 (357.7)
	$p_{stick} = 0.5$	Testing Reward	**8.10 (4.99)**	6.63 (8.07)	7.10 (9.52)	6.44 (3.00)
		Testing Violations	**0.18 (0.84)**	0.54 (1.53)	0.22 (1.18)	0.48 (1.24)
		Training Violations	91.8 (16.85)	157.6 (18.4)	**80.4 (17.43)**	3126 (2823)

M. Giacobbe et al. *Shielding Atari Games with Bounded Prescience*. 2021.
D. Hafner, et al. *Dream to Control: Learning Behaviors by Latent Imagination*. 2020.
D. Hafner, et al. *Mastering Atari with Discrete World Models*. 2021.
J. Achiam, et al. Constrained Policy Optimization. 2017.

Examining Latent Dynamics



Drawbacks

- Rolling out the world model for long horizons leads to compounding model errors.
- Without safety critics approximate methods, **latent shielding** can only be used for relatively short horizons.
- Overestimation of violation hinders exploration.
- Approximate, but how much approximate?

Approximate Shielding of Atari Agents for Safe Exploration (with A. Goodall, ALA Workshop @AAMAS)

- Safety critics [Srinivasan et al. 2020] for further look-ahead abilities, which reduces the overestimation of expected costs.
- Approach grounded on Probabilistic Computation-tree Logic (PCTL).
- We derive PAC bounds on the probability of accurately detecting a safety violation in the near future.
- We empirically show our approach reduces the rate of safety violations on a small set of Atari games.

PCTL and ε-Bounded Safety

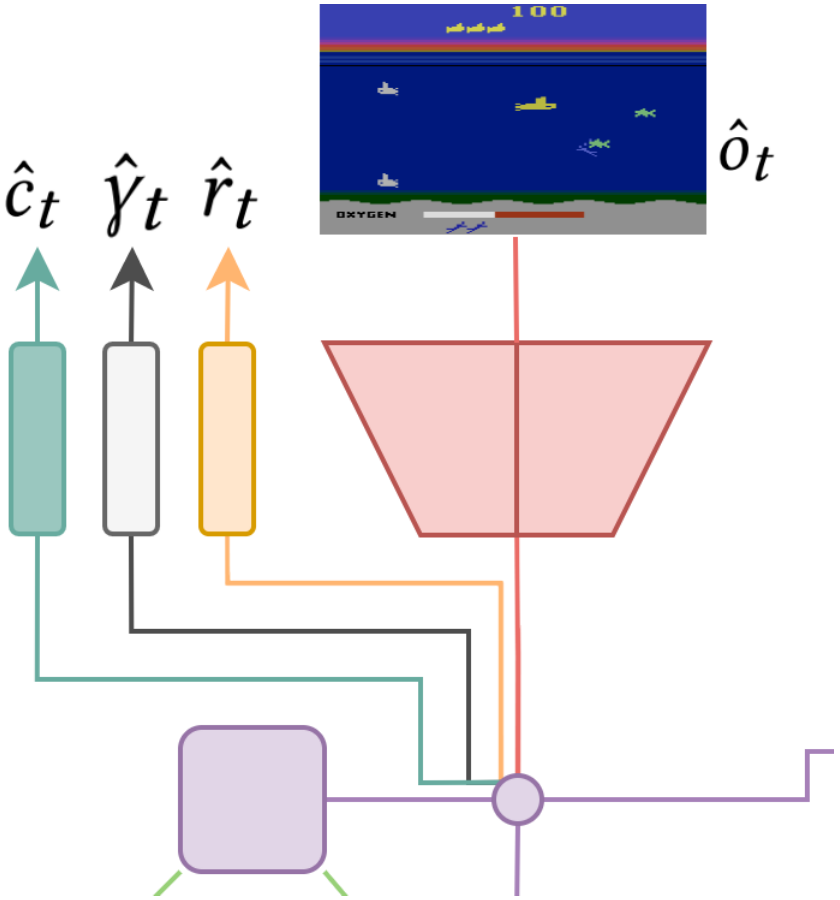
- Recall: we encode the safety constraint as some formula Φ.
- Consider some fixed (stochastic) policy π and a POMDP M.
- Together π and M define a (probabilistic) transition system T.
- **ε-Bounded Safety:** a state $s \in S$ is **ε-bounded safe** iff

$$s \models P_{1-\varepsilon}(\square_{\leq n}\Phi)$$

- We don't know what the "true" transition system T is, so we must learn it!

Cost Function

- For look-ahead shielding and safe behaviour learning we augment the RSSM from DreamerV2 [Hafner et al., 2020] with a cost predictor.
- Targets for the cost predictor are constructed as
  $c_t = \{0,$ if $s_t \models \Phi$
  C, otherwise
  where $C > 0$ is a hyperparameter.



Safety Critic Learning

- Safety critics estimate the expected costs under the task policy π_{task}.
- They give us an idea of how safe specific states are under the task policy state-distribution.
- Additionally, we can use them to bootstrap the end of 'imagined' trajectories for further look-ahead capabilities.
- To prevent overestimation of the expected costs we jointly train two safety critics v_1 and v_2 with a TD3-style algorithm [Fujimoto et al., 2018], and then take the minimum.

Approximate Shielding

- **Recall:** we are concerned with checking a formula of the form $P_{1-\varepsilon}(\square_{\leq n}\Phi)$.
- Let $\mu_{s\models\phi}$ be shorthand for $\mu_s(\{\tau \mid \tau[0] = s, \text{ for all } 0 \leq i \leq n, \tau[i] \models \Phi\})$
- We estimate $\hat{\mu}_{s\models\phi} \approx \mu_{s\models\phi}$ by Monte-Carlo sampling: roll-out the world model with π_{task} to generate a batch of *m* traces $\tau = \langle \hat{s}_1, \ldots, \hat{s}_H \rangle$ of compact latent states, where H is the look-ahead horizon.
- For each trace τ we compute the discounted cost as follows

$$\mathrm{cost}(\tau) = \sum_{t=1, H} (\hat{\gamma}_t)^{t-1} \cdot \hat{c}_t$$

Approximate Shielding (with Safety Critics)

- If we use safety critics we compute the bootstrapped costs instead,

$$\text{b-cost}(\tau(i)) = (\sum t=1, H-1\ (\hat{\gamma}(i)_t){}^\wedge t-1 \cdot \hat{c}(i)_t) + \min\{v_1(\hat{s}(i)_H), v_2(\hat{s}(i)_H)\}$$

- We can now estimate $\mu_{s|=\phi}$ with a greater look-ahead horizon $T > H$, as the safety critics capture the expected costs from $\hat{s}_H$ and beyond.

$$\hat{\mu}_{s|=\phi} = 1/m \sum i=1, m\ 1(\text{b-cost}(\tau\,(i)) < \gamma{}^\wedge T-1 \cdot C\ )$$

Environment Interaction

- To mitigate safety violations in the real environment we pick actions with the shielded policy,

$$\pi_{shield}(\cdot \mid s) = \begin{cases} \pi_{task}(\cdot \mid s) & \text{if } \hat{\mu}_{s\models\phi} \in [1 - \varepsilon + e, 1] \\ \pi_{safe}(\cdot \mid s) & \text{otherwise} \end{cases}$$

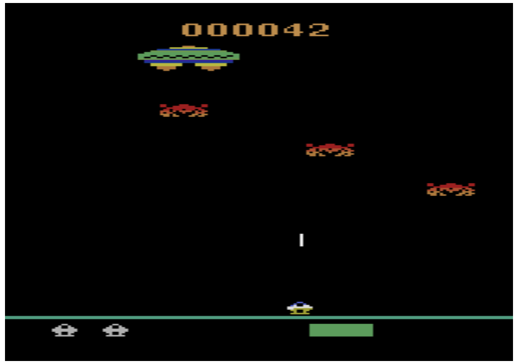
- We make the distinction here between the desired safety level ε and the approximation error e.

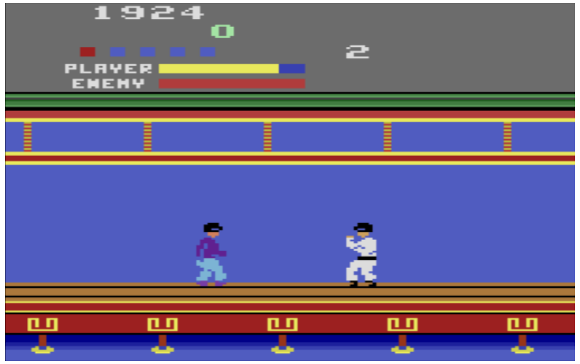
Probabilistic Guarantees

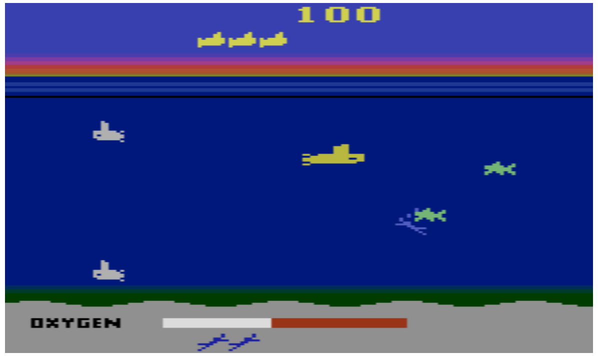
- Although the world model only gives us an approximate transition system $\tilde{T} \approx T$, we can obtain some probabilistic guarantees for the 'true' transition system T.
- **Proposition** Given access to the 'true' transition system T , with probability $1-\delta$ we can estimate the measure $\mu_{s|=\phi}$ up to some approximation error e, by sampling m traces $\tau \sim T$, provided,

$$m \geq (2/\varepsilon\text{^}2) \log (2/\delta)$$

Atari Games







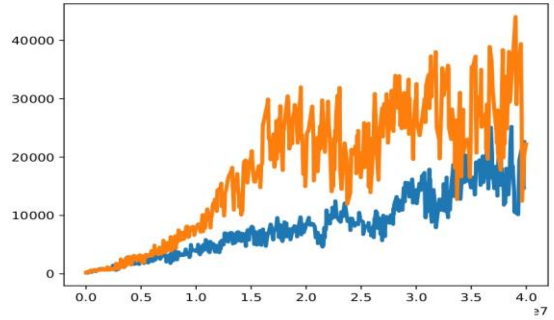
$\Phi=\neg\text{hit}\wedge\neg\text{overheat}$

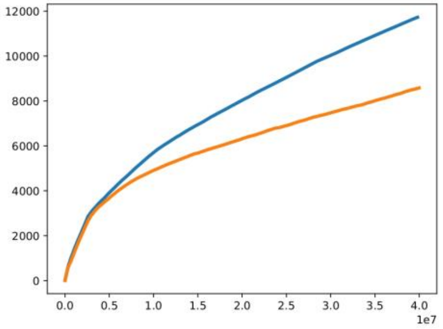
$\Phi=\neg\text{energy-loss}\wedge\neg\text{loose-life}$

$\Phi=(\text{surface}\Rightarrow((\text{diver}\wedge\text{low-oxygen})$
$\vee\text{very-low-oxygen}\vee\text{six-divers}))$
$\wedge\neg\text{out-of-oxygen}\wedge\neg\text{hit}$

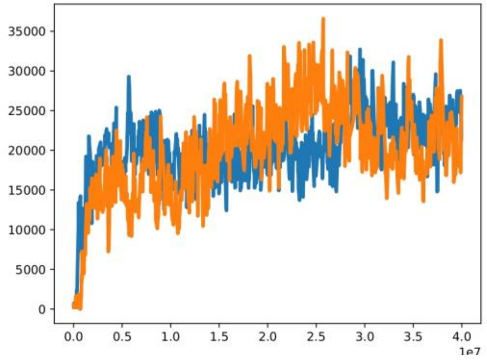
Results

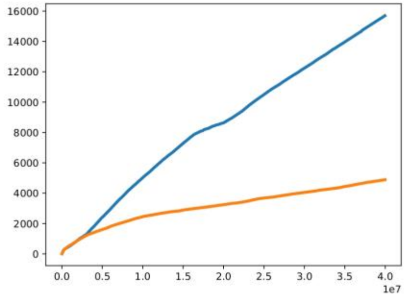
Assault



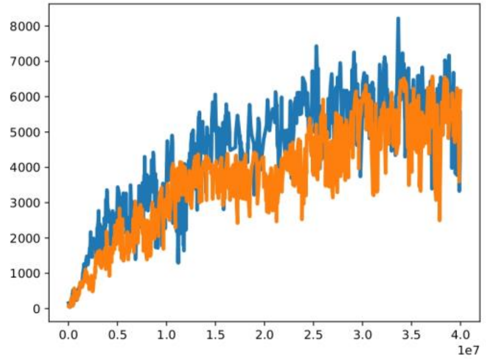


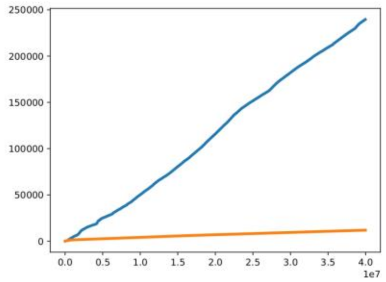
Kung-fu Master





Seaquest





Conclusions

- Built on latent shielding and BPS, we obtain a **general purpose algorithm for approximate shielding** that uses safety critics and policy roll-outs in the latent space of a world model to successfully shield Atari agents.
- In contrast to previous work our algorithms requires minimal hyperparameter tuning and no shield introduction schedules.
- Approximate shielding lacks the strict guarantees of classical shielding approaches, although we are able to get some **probabilistic guarantees**.
- The empirical results are promising and demonstrate that agents can benefit from shielding in safety-critical domains.

Future Work

- Conduct a more in-depth theoretical analysis of the algorithm to derive bounds on the approximate transition system.
- Linear-time properties beyond safety.
- Multi-agent environments.
- In this work we assume access to perfect state labels, in the real world sensor's are noisy and it would be interesting to investigate how we can deal with this.
- It may be important to show that our approach is model agnostic (we can use any world model architecture). We are currently integrating approximate shielding with DreamerV3 [Hafner et al., 2023] as a starting point.

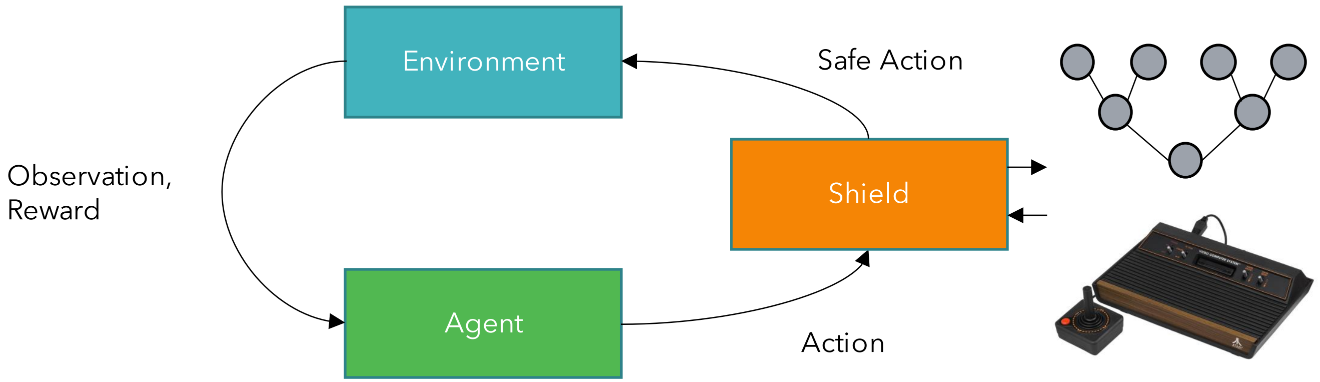
Takeaways

- **Latent shielding** allows to shield agents in high-dimensional environments without *a priori* knowledge of the dynamics.
- It does so by learning the environment model rather than having it be handcrafted.
- Shielding can harm model-based DRL algorithms - introduce the shield gently with a **Shield Introduction Schedule**.

Future Work

- Cost function instead of binary classifier.
- Actor-critic for path evaluation.
- Linear-time properties beyond safety.
- Probabilistic safety.
- Multi-agent environments.

BP Shielding



Environment

Safe Action

Shield

Observation, Reward

Agent

Action

Given some finite trace $\rho = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \ldots \xrightarrow{a_{n-1}} s_n$

And the set of all finite traces of length H from state s, $\varrho_H(s)$

A policy π is H-bounded safe iff.

There are no safe traces

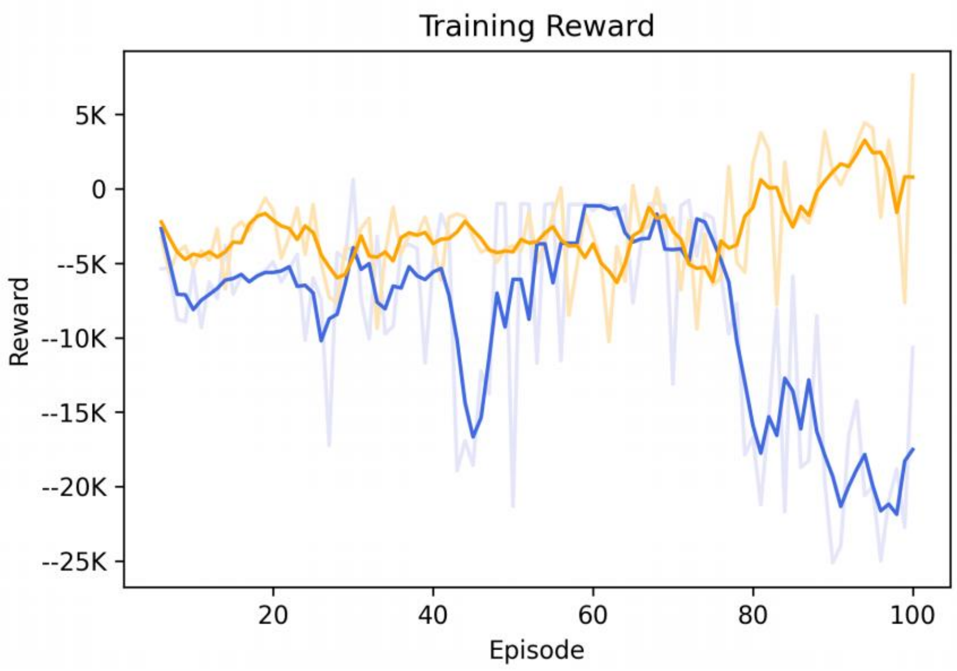
$$\forall s \in \mathcal{S}.(\exists \rho = (s,a) \in \varrho_H(s) S(\rho,\phi) \wedge \pi(s_0) = a_0) \vee \forall \rho \in \varrho_H(s).\neg S(\rho,\phi))$$

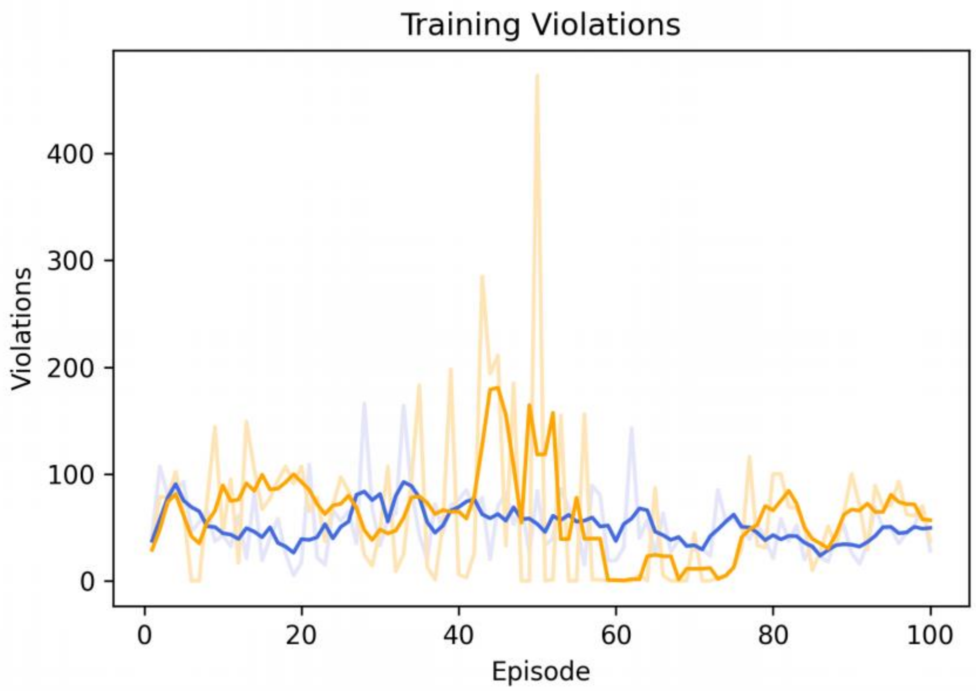
If there is a safe trace of length H, we take it

M. Giacobbe et al., *Shielding Atari Games with Bounded Prescience*. AAMAS 2021.

Do Shield Introduction Schedules Work?

Yes





Without SIS With SIS

Verifying Safe States

- We encode the safety constraint on the environment as some propositional formula Φ.

- **Goal:** find a policy π that maximises expected reward, while minimizing violations of the safety constraint Φ during training (here γ is the discount factor).

Safe Behaviour Learning

- The safe policy π_{safe} is used as the backup policy if we detect that the task policy π_{task} is likely to commit a safety violation in the next T steps.
- Since we have no access to an abstraction of the environment, we cannot synthesize a shield before training and so the safe policy must be learned.
- We train the safe policy π_{safe} with the same actor-critic algorithm used to train the task policy π_{task} (see [Hafner et al., 2020]), except we are only concerned with minimising expected costs.

Approximate Shielding

- We estimate the probability of committing a safety violation (in the next *n* steps) under the task policy state-distribution.
- If $P_{\pi_{task}}$ [violation] > ε, play with the backup policy π_{safe} else play with the task policy π_{task}.