Stable Planning through Aligned Representations in Model-Based Reinforcement Learning

Misagh Soltani Forest Agostinelli

msoltani@email.sc.edu foresta@cse.sc.edu

Dept. of Computer Science & Engineering
University of South Carolina

The third Reinforcement Learning Conference (RLC)

Presenter: Forest Agostinelli

August 17, 2026

Molinaroli College of Engineering and Computing
UNIVERSITY OF SOUTH CAROLINA

Code on GitHub

Download on the App Store

Long-Horizon Planning

1. Hand-code a simulator

Accurate, but a new simulator must be specified manually for every system and every environment.

2. Write a symbolic model

Plan over symbols, as in PDDL, then hard-code how each symbol grounds to the actions the agent executes.

3. Learn the world model

Learn the world model from data obtained by interacting with the environment. A problem instance is then given as a start image and a goal image (e.g., DeepCubeAI [1]).

[1] Forest Agostinelli and Misagh Soltani. "Learning discrete world models for heuristic search". In: *Reinforcement Learning Conference*. 2024

Long-Horizon Planning

1. Hand-code a simulator

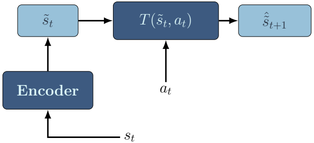
Accurate, but a new simulator must be specified manually for every system and every environment.

2. Write a symbolic model

Plan over symbols, as in PDDL, then hard-code how each symbol grounds to the actions the agent executes.

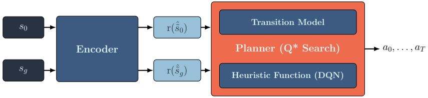
3. Learn the world model

Learn the world model from data obtained by interacting with the environment. A problem instance is then given as a start image and a goal image (e.g., DeepCubeAI [1]).



(a) Train a world model with a **discrete latent space** on clean states [1].

(b) Train heuristic function with Q-learning, in the discrete latent space, on data generated by the world model.



(c) Perform search in the discrete latent space, using the learned world model and heuristic function.

[1] Forest Agostinelli and Misagh Soltani. "Learning discrete world models for heuristic search". In: *Reinforcement Learning Conference*. 2024

Problem: Noisy Observations Break the Learned World Model

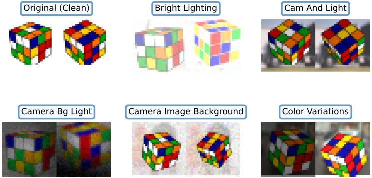
Assumption: observations are clean. A world model trained on clean offline images does not generalize to noise.

Option 1. Re-train for every new kind of noise

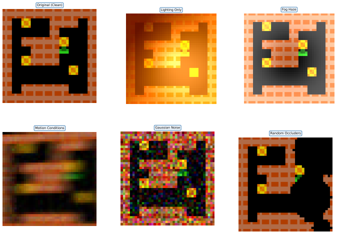
- Computationally expensive.
- The analysis has to be redone, because both the transition model and the heuristic function change.

Option 2. Train the world model on noisy observations

- Noise introduces stochasticity. Noise at t differs from noise at $t+1$, so transitions in a deterministic MDP appear stochastic.
- The latent encoding of the goal image changes with the noise, which breaks exact goal testing and state re-identification.
- In our experiments, the resulting world model predicts the next discrete latent state with only $\approx 50\%$ accuracy on held-out data as opposed to $\approx 100\%$ for the clean data.

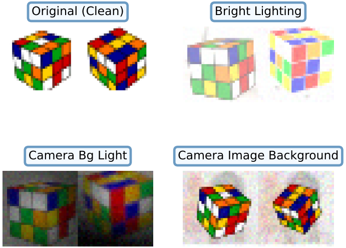


Rubik's Cube

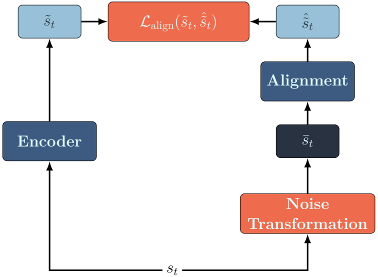


Sokoban

SPAR: Aligning Noisy States to a Fixed Discrete World Model



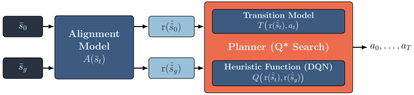
A clean state and examples of the noise transformations applied to it.

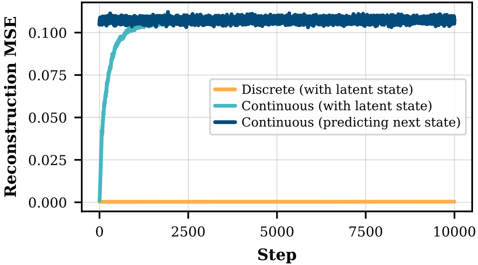


Replace the encoder with an **alignment model**, and keep the world model and the heuristic function **fixed**.

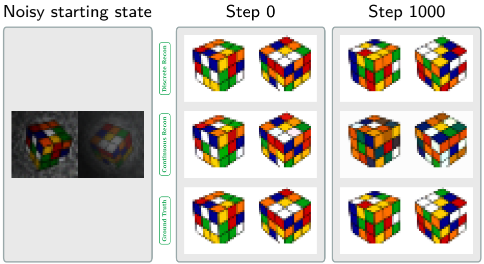
> Train an **alignment model** that maps noisy states to the same discrete latent states as their corresponding clean states. A new kind of noise then only requires re-training the alignment model, while keeping the world model and the heuristic function fixed.

Model Performance: Discrete versus Continuous Latent Space





Reconstruction MSE averaged across the Rubik's Cube visual transformations (100 sequences of 10,000 steps each).



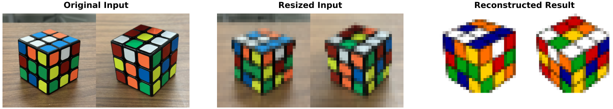
Reconstructed output of the alignment model after taking 1,000 steps.

Continuous alignment is possible. **Rounding** prevents error accumulation over long rollouts and recovers discrete latent state of the clean data. It also enables exact goal testing and state re-identification, which are crucial for planning.

Planning Performance and Generalization to Real Images

	Observations	SPAR	DeepCubeAI	Greedy Policy
Rubik's Cube	Clean	100%	100%	0%
	Noisy, 16 variants	**89.92%**	0%	0%
	Real-world, 10 instances	**50%**	0%	0%
Sokoban	Clean	100%	100%	41.9%
	Noisy, 18 variants	**96.31%**	0%	0%

Percentage of problem instances solved using Q^* search [2].



Two phone photos of the Rubik's Cube, resized to 32×32 and concatenated, and the decoded prediction of the alignment model.

No real-world images were seen during training.

SPAR generalizes to noise not seen during training, including real-world pictures of the Rubik's Cube.

[2] Forest Agostinelli et al. "Q* search: Heuristic search with deep q-networks". In: *ICAPS Workshop on Bridging the Gap Between AI Planning and Reinforcement Learning workshop*. 2024

Thank you for your attention!

Any questions?

Takeaway. Our results show that **both** the alignment model and the discrete latent space are needed.

github.com/misaghsoltani/SPAR

Download on the App Store

Future work

- Use formal logic or natural language to specify goals, instead of goal images.
- Treat particularly noisy examples as partially observable MDPs (POMDPs).
- Learn the world model and the heuristic function when the offline data is not clean, so that observations of the same underlying state still map to the same discrete latent state.

This material is based upon work supported by the National Science Foundation under Award No. 2426622. The authors gratefully acknowledge the computational resources provided by the Theia high performance computing cluster at the University of South Carolina, which is supported by National Science Foundation Grant No. 2320292. We also acknowledge the technical assistance and resources provided by Research Computing at the University of South Carolina (RRID:SCR_027488).

References

[1] Forest Agostinelli and Misagh Soltani. "Learning discrete world models for heuristic search". In: *Reinforcement Learning Conference*. 2024.

[2] Forest Agostinelli et al. "Q* search: Heuristic search with deep q-networks". In: *ICAPS Workshop on Bridging the Gap Between AI Planning and Reinforcement Learning workshop*. 2024.

Supplementary slides

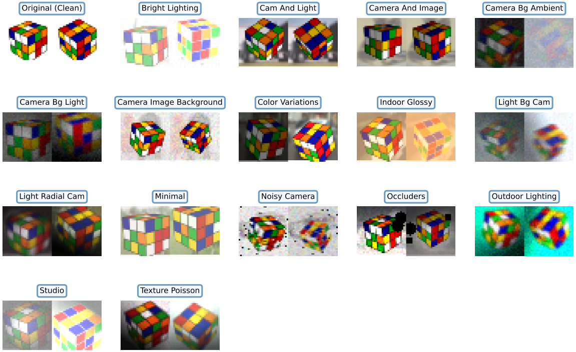
Why Not Just Train the World Model on Noisy Observations?

- The same state produces **different latent encodings** under different noise, which breaks exact state re-identification and goal testing.
- Noise at t differs from noise at $t+1$, so transitions in a deterministic MDP **appear stochastic** during training.
- Every new kind of noise forces **re-training** of the world model and of any function trained with data generated from it.

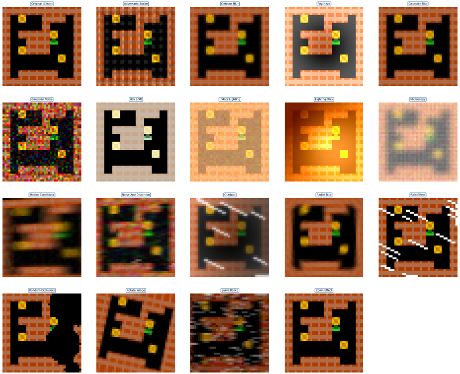
Accuracy of the predicted discrete latent state on held-out data	
World model trained on noisy observations	$\approx 50\%$
Encoder trained on clean observations	100%
Alignment model on held-out noisy data	$\approx 90\%$

The encoder reaches **100%** accuracy with a simpler neural network, and rounding removes every error below 0.5, so no error accumulates across thousands of timesteps.

Visual Transformations



Rubik's Cube, 16 variants



Sokoban, 18 variants

Transformations to which the cost-to-go is invariant are applied before rendering (background), during rendering (lighting, color, geometry), and after rendering, and are then combined into the variants used to train the alignment model. Backgrounds for the Rubik's Cube include randomly selected images from the CIFAR dataset.

Training Objectives

Discrete world model

$$\mathcal{L}(\theta) = (1-\omega)\,\mathcal{L}_{\mathrm{rec}}(\theta) + \omega\,\mathcal{L}_{\mathrm{m}}(\theta)$$

- $\mathcal{L}_{\mathrm{rec}}$ is the reconstruction error of the encoder and decoder, and $\mathcal{L}_{\mathrm{m}}$ is a symmetric transition loss with a stop-gradient operator on one side of each term.
- ω starts small, weighting reconstruction higher, and is increased until both terms are weighted equally.
- The encoder output passes through a logistic activation and is rounded to $\{0,1\}$, with a straight-through estimator enabling gradient-based training.

Alignment model

$$\mathcal{L}_{\mathrm{align}}(\phi) = \frac{1}{N}\sum_{i=1}^{N}\left\|A(\bar{s}_t^{(i)}) - \tilde{s}_t^{(i)}\right\|_2^2, \qquad \bar{s}_t = \mathcal{N}(s_t), \quad \tilde{s}_t = E(s_t)$$

- At inference the outputs of A are rounded element-wise to $\{0,1\}$.

Planning Performance

DOMAIN	OBSERVATION	SOLVER	LEN	NODES	SECS	NODES/SEC	SOLVED	95% CI
Rubik's Cube	Clean	SPAR	22.85	2.00E+05	6.21	3.22E+04	100%	[99.6, 100.0]
		DeepCubeAI	22.85	2.00E+05	6.21	3.22E+04	100%	[99.6, 100.0]
		Greedy Policy	–	–	–	–	0%	[0.0, 0.4]
	Noisy	SPAR	22.85	2.00E+05	6.21	3.22E+04	**89.92%**	[89.4, 90.4]
		DeepCubeAI	–	–	–	–	0%	[0.00, 0.02]
		Greedy Policy	–	–	–	–	0%	[0.00, 0.02]
	Real-World	SPAR	23.61	1.90E+05	6.21	3.06E+04	**50%**	[23.7, 76.3]
		DeepCubeAI	–	–	–	–	0%	[0.0, 27.8]
		Greedy Policy	–	–	–	–	0%	[0.0, 27.8]
Sokoban	Clean	SPAR	32.32	2.39E+04	20.58	1.16E+03	100%	[99.6, 100.0]
		DeepCubeAI	32.32	2.39E+04	20.58	1.16E+03	100%	[99.6, 100.0]
		Greedy Policy	–	–	–	–	41.9%	[38.9, 45.0]
	Noisy	SPAR	32.30	2.39E+04	20.55	1.16E+03	**96.31%**	[96.0, 96.6]
		DeepCubeAI	–	–	–	–	0%	[0.00, 0.02]
		Greedy Policy	–	–	–	–	0%	[0.00, 0.02]

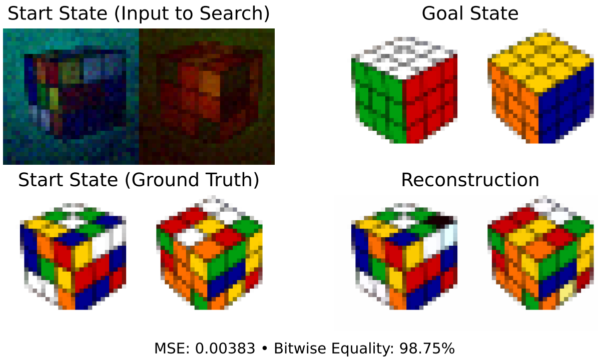
Each visual variant is evaluated on 1,000 test instances, so the noisy rows aggregate 16,000 instances for the Rubik's Cube and 18,000 for Sokoban. The clean rows use 1,000 instances per domain and the real-world row uses 10. DeepCubeAI was not designed for noisy observations and shows how a latent-space planner behaves without alignment.

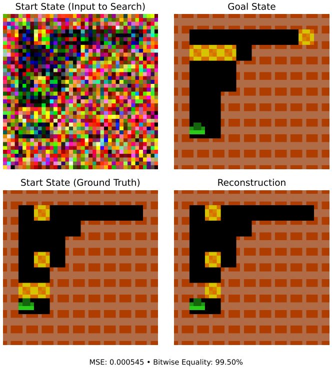
Solve Rate per Visual Transformation

Rubik's Cube		Sokoban	
Camera and Lighting	97.6%	Rotate Image	100.0%
Minimal	97.0%	HSV Shift	100.0%
Bright Lighting	96.8%	Gaussian Blur	100.0%
Camera and Image Background	95.4%	Fog Haze	100.0%
Indoor and Glossy	94.6%	Defocus Blur	100.0%
Occluders	94.1%	Indoor Lighting	99.9%
Camera and Image	93.9%	Adversarial Noise	99.9%
Color Variations	93.7%	Radial Blur	99.8%
Texture and Poisson Noise	91.3%	Zoom Effect	99.8%
Studio	88.2%	Rain Effect	99.7%
Camera and Background and Lighting	84.9%	Outdoor	99.3%
Lighting and Background and Camera	84.3%	Surveillance	98.9%
Lighting and Radial Blur and Camera	84.0%	Noise and Distortion	98.4%
Camera and Background and Ambient Light	82.0%	Microscopy	98.1%
Noisy Camera	80.7%	Gaussian Noise	97.6%
Outdoor Lighting	80.2%	Lighting Only	97.6%
		Motion Conditions	92.4%
		Random Occluders	52.2%

1,000 test instances per variant.

Failure Cases





- In many failure cases, information relevant to the clean state was occluded by noise, and a lack of information may make it impossible to find a path with SPAR.