Temporal Abstraction Through Precision-Gated Latent Dynamics

Anonymous authors
Paper under double-blind review

Abstract

Temporal abstraction is usually modeled through explicit options or skills. We ask whether recurrent latent dynamics can satisfy option-like criteria without explicit option identities, option-specific policies, or learned termination functions. We study a precision-gated recurrent action-value model in a controlled navigation setting and evaluate its latent dynamics by operational criteria: persistence, behavioral influence, input-dependent initiation, modulation or interruption, reuse under known symmetry, and absence of option labels. The model uses input-driven latent dynamics with prediction-error correction, allowing us to test whether effective duration and correction change under temporal cost and observation-reliability manipulations. The paper provides a scoped mechanistic account of option-like temporal abstraction as a property of learned latent dynamics rather than as a discrete architectural object.

1 Introduction

Agents often need actions whose consequences extend beyond a single primitive step. The options framework formalizes this problem by letting an agent select a temporally extended policy rather than only primitive actions Sutton et al. (1999), and later work has learned option policies or discovered option-like behaviors from value and graph structure Bacon et al. (2017); Machado et al. (2017). This makes temporal abstraction explicit, but it also casts extended behavior as the execution of discrete policy objects. We ask whether recurrent latent dynamics can satisfy analogous criteria without explicit option identities, option-specific policies, or learned termination functions.

We propose precision-gated latent dynamics as one mechanism for this form of temporal abstraction. Current inputs drive recurrent latent modes with learned timescales; those modes persist, mix, and influence primitive action values. Linear-modal analyses are useful here because they make spectra, decay rates, and perturbations directly measurable Williams et al. (2015); Brunton et al. (2022). Prediction-error gates control how strongly residual signals update the latent state. In model-level terms, this is a scoped use of precision: the weighted quantity is a prediction-error update, not a general active-inference implementation Friston (2010); Limanowski et al. (2024).

We evaluate the mechanism by criteria rather than by label. This places the work near predictive-representation accounts that learn reusable forecasts of future state or occupancy Dayan (1993); Stachenfeld et al. (2017), but our tests focus on whether latent dynamics influence action over time. We test the criteria in Four Rooms, where interpretable geometry and known C_4 symmetry support persistence, perturbation, input-forcing, gate, and canonical-frame reuse analyses. Group-symmetry approaches to reinforcement learning motivate treating this as a constrained reuse test rather than arbitrary out-of-distribution generalization van der Pol et al. (2020); Wang et al. (2022).

The paper makes three contributions. First, it defines option-like latent dynamics operationally. Second, it studies an input-driven recurrent architecture with learned latent timescales and precision-gated prediction-error updates. Third, it tests the resulting criteria in a controlled navigation setting. The evidence supports a scoped claim: temporally

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Operational criteria for option-like latent dynamics.

Criterion	What must be shown	Option analogue
Persistence	Latent influence lasts beyond one primitive step	temporally extended action
Behavioral influence	Latent perturbations change action values, policy logits, or selected actions	intra-option policy
Input-dependent initiation	Task inputs or task phase drive latent dynamics	initiation set
Modulation / interruption	Later input or prediction-error gates reduce, redirect, or correct latent influence	termination or interruption
Reuse	The same learned policy and dynamics remain useful after canonical frame remapping	reusable extended control
No option labels	The architecture has no option IDs, option-specific policies, or termination classifiers	contrast to explicit options

extended control can arise from how recurrent systems initiate, sustain, correct, and interrupt latent dynamics, rather than from selecting a discrete library of option policies.

2 Conceptual Setup and Operational Criteria

2.1 Option interface and latent-dynamics criteria

The options framework defines a temporally extended action as

$$o = \langle I_o, \pi_o, \beta_o \rangle, \tag{1}$$

where I_o is the initiation set, π_o is the intra-option policy, and β_o is the termination rule Sutton et al. (1999). We use this tuple as an interface: availability, action selection while active, and stopping. A recurrent state h_t is only a candidate carrier for that interface. It may store recent information without creating temporally extended control. We therefore treat latent dynamics as option-like only when they satisfy the criteria in Table 1.

2.2 Precision and reuse scope

Active-inference accounts use precision to describe the confidence or gain assigned to sensory evidence, prediction errors, policies, transitions, or actions Friston (2010); Limanowski et al. (2024). This breadth makes the term useful but easy to overuse. A claim about precision is interpretable only after the weighted quantity is specified.

Here the weighted quantity is a residual update to the recurrent state. In abstract form, the update has the structure

$$h_{t+1} = F(h_t, x_t) + g_t e_t, \tag{2}$$

where e_t is a prediction error or residual signal and g_t is a learned gate. Larger g_t gives the residual more influence over the next latent state; smaller g_t lets the current latent trajectory dominate. The empirical question is whether changing this weight changes the duration or behavioral effect of latent dynamics.

A known task symmetry gives a controlled way to test reuse without claiming broad out-of-distribution generalization. Let g be a rotation in C_4. If a state and goal can be rotated into a canonical frame, a policy can act in that frame and then rotate the selected action back:

$$\pi_g(x) = g\, \pi_0(g^{-1} x). \tag{3}$$

Both equations are scope definitions. The gate is a model-level prediction-error weight, not a full active-inference generative model or neural precision variable. Canonical-frame evaluation tests reuse under a supplied symmetry; it does not show that the model discovers the symmetry group or generalizes to arbitrary environments.

3 Model: Precision-Gated Latent Dynamics

The model receives the current navigation state, updates a recurrent latent state, and reads primitive action values from that state. It has no option identifiers, option-specific policies, or learned termination classifiers.

3.1 Content pathways and input forcing

At each timestep, the model observes normalized position, remaining time, and fixed wall information from the Four Rooms map. Let $x_t \in [0,1]^2$ denote position, let $m_t \in [0,1]$ denote remaining time, and let $b(x_t) \in \{0,1\}^4$ indicate whether each primitive movement would hit a wall. Two encoders first produce unconstrained content vectors and then project them to unit length,

$$p_t = f_\theta([x_t, b(x_t)]), \quad \hat{p}_t = \frac{p_t}{\|p_t\|_2}, \qquad \tau_t = g_\theta(m_t), \quad \hat{\tau}_t = \frac{\tau_t}{\|\tau_t\|_2}, \tag{4}$$

where $\hat{p}_t$ carries position and wall content and $\hat{\tau}_t$ carries remaining-time content. This normalization makes pathway direction, rather than pathway magnitude, the quantity supplied to the forcing matrices. The model can then rotate different input directions into different latent modes, while the effect of the B matrices is not conflated with arbitrary encoder-output scale.

A third pathway represents the interaction between position and time,

$$\iota_t = (W_p \hat{p}_t) \odot (W_\tau \hat{\tau}_t), \qquad \hat{\iota}_t = \frac{\iota_t}{\|\iota_t\|_2}, \tag{5}$$

where $\odot$ denotes elementwise multiplication. The interaction pathway allows the model to drive different latent dynamics for the same position at different remaining times, or for the same remaining time in different parts of the maze. The base forcing term is

$$u_t^{\text{base}} = B_p \hat{p}_t + B_\tau \hat{\tau}_t + B_\iota \hat{\iota}_t. \tag{6}$$

The matrices B_p, B_τ, and B_ι map pathway directions into latent coordinates, so pathway-to-mode coupling can be analyzed without conflating encoder scale with forcing strength.

3.2 Recurrent latent dynamics

The recurrent state $h_t \in \mathbf{R}^{32}$ evolves through learned, input-independent linear dynamics plus input-driven forcing:

$$h_t = A h_{t-1} + u_t. \tag{7}$$

The matrix A is block diagonal, with each two-dimensional block parameterized as

$$A_i = \begin{pmatrix} a_i & -b_i \\ b_i & a_i \end{pmatrix}. \tag{8}$$

Each block has eigenvalues $a_i \pm i b_i$. The magnitude $\sqrt{a_i^2 + b_i^2}$ determines how slowly activity in that block decays, and the phase determines rotation within the block. This modal parameterization provides an analyzable linear substrate for measuring persistence and perturbation effects, without requiring the broader claim that the learned recurrent state is a complete Koopman representation Williams et al. (2015); Brunton et al. (2022). Because A is shared across inputs, the model cannot choose a different transition matrix for each state. Instead, task inputs select which latent modes receive forcing.

3.3 Prediction-error correction and action values

The model also includes a prediction-error correction pathway. A prediction head estimates the current position from the previous latent state and previous action,

$$\tilde{x}_t = q_\phi(h_{t-1}, a_{t-1}), \qquad e_t = x_t - \tilde{x}_t. \tag{9}$$

In the experiments reported here, q_ϕ is not trained by a supervised position-prediction loss. Its parameters receive gradient only through the temporal-difference objective, along the path from $\tilde{x}_t$ to the residual, correction signal, latent state, and action values. Thus e_t should be read as a learned correction signal that the control objective shapes, not as a calibrated one-step forecast error.

The residual e_t is combined multiplicatively with position and time content to form a correction signal,

$$z_t = (W_e e_t) \odot (W_p^c \hat{p}_t) \odot (W_\tau^c \hat{\tau}_t), \qquad c_t = \frac{z_t}{\|z_t\|_2}, \tag{10}$$

and this signal supplies an additional latent-state update,

$$u_t = u_t^{\text{base}} + B_c c_t. \tag{11}$$

This pathway is the model's prediction-error gate: it gives residual errors a context-dependent route into the recurrent state.

Primitive action values are read directly from the recurrent state:

$$Q_t = W_Q h_t, \tag{12}$$

with one value for each primitive movement action and the stay action. This readout makes behavioral influence directly testable.

4 Experimental Setting and Analyses

This section defines the task, precision-demand manipulations, performance metrics, and analyses used to test the operational criteria. Results are reported by criterion, not by training job.

4.1 Task and precision-demand manipulations

We evaluate the model in Four Rooms, a grid-navigation task with fixed walls, doorway bottlenecks, and known C_4 rotational symmetry. At each step the agent observes normalized position, remaining time, and wall indicators, and chooses among four movement actions and a stay action. Episodes end at goal arrival or horizon expiry. We report success rate, return, episode length, and excess path length, defined as episode length minus shortest path length on successful episodes.

We manipulate precision demand along two axes. First, we vary the per-step cost of movement across magnitudes $0.04, 0.08, 0.12, 0.16, 0.20, 0.30, 0.40$, and 0.50, implemented as negative rewards. Higher temporal cost makes delayed or misdirected action more expensive. Second, we vary the emission probability of the position observation. Lower emission probability makes position observations less reliable, so residual signals are not uniformly trustworthy even though the agent may still need to use them when precise correction matters.

Together, these manipulations test whether option-like latent dynamics have a fixed effective duration or whether persistence, action influence, and correction change with temporal cost and observation reliability. They are mechanistic precision-demand analyses of one trained architecture, not population-level seed-variance estimates. A performance change alone would not establish flexible duration; the relevant evidence is whether latent persistence, action influence, and correction-pathway activity change with these precision-demand manipulations.

4.2 Latent-dynamics analyses

We analyze the trained model with probes and interventions matched to the operational criteria. Table 2 states, for each criterion, the measurement and the precision-demand test where applicable.

The precision-demand manipulations are interpreted as tests of flexible effective duration, not as a claim that higher cost or lower emission probability must shorten dynamics. The

Table 2: Analyses used to test each option-like criterion.

Criterion	Measurement	Precision-demand test
Persistence	eigenvalues, impulse responses, effective decay scales	effective duration changes with temporal cost or observation reliability
Behavioral influence	latent perturbations or ablations over time	action-value influence lengthens, shortens, or redirects with precision demand
Input-dependent initiation	pathway forcing maps over position, time, and task phase	task variables drive specific latent modes across phases
Modulation / interruption	residual, correction signal, and correction forcing diagnostics	correction pathway changes latent updates or action values with temporal cost or emission probability
Reuse	canonical frame evaluation under supplied C_4 rotations	same policy/dynamics remain useful without new option labels or retraining

correction-pathway diagnostics are learned control signals, not calibrated physical forecast errors, because the prediction head is trained through the control objective. The reuse test is likewise scoped: it tests reuse under a supplied symmetry transformation, not learned symmetry discovery or arbitrary novel-context generalization.

4.3 Contrast cases and claim limits

The analyses distinguish option-like latent dynamics from generic recurrent memory by requiring all criteria in Table 2. We include only contrast cases that were actually run or directly inspected. Failed or partial variants are qualitative context, not statistical baselines, unless they share the same evaluation unit and metrics.

5 Results

We present results in the order of the operational criteria rather than in the order analyses were run. The goal is to test whether the model's latent dynamics persist, influence action, begin under task inputs, change with correction signals and precision-demand manipulations, and remain reusable under supplied task symmetry.

5.1 Task competence and flexible effective duration

The model must first solve the navigation task well enough for its latent dynamics to be interpretable as useful control structure. Figure 1 reports task competence across precision-demand conditions and pairs those behavioral metrics with duration and correction diagnostics. The cost manipulation increases the consequence of slow or misdirected action, while the emission-probability manipulation changes observation reliability. The key question is not whether performance changes, but whether effective duration, action influence, or correction-pathway activity changes with these demands.

5.2 Latent modes persist and influence action values

Figure 2 tests the two requirements that make latent dynamics more than memory. First, latent activity should persist beyond one primitive step. Second, perturbing or ablating latent modes should change action values or selected actions over subsequent timesteps. The perturbation panel is essential: eigenvalues and impulse responses show persistence, but only action-linked perturbations establish behavioral influence.

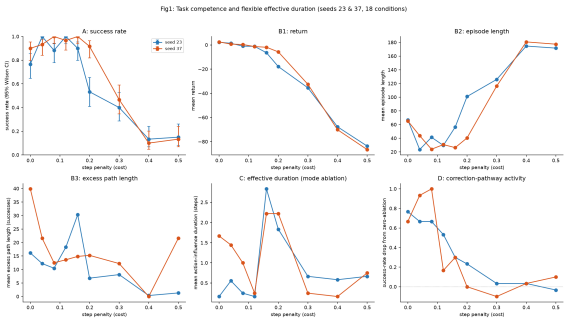


Figure 1: Task competence and flexible effective duration under precision-demand manipulations. Panels A–B summarize held-out episode outcomes from the cost sweep; panel C reports effective duration from teacher-forced mode ablations; panel D reports correction-pathway importance from zeroed or shuffled correction rollouts. Performance changes alone do not establish flexible option-like duration; the evidence-bearing panels are the effective-duration and correction-pathway trends.

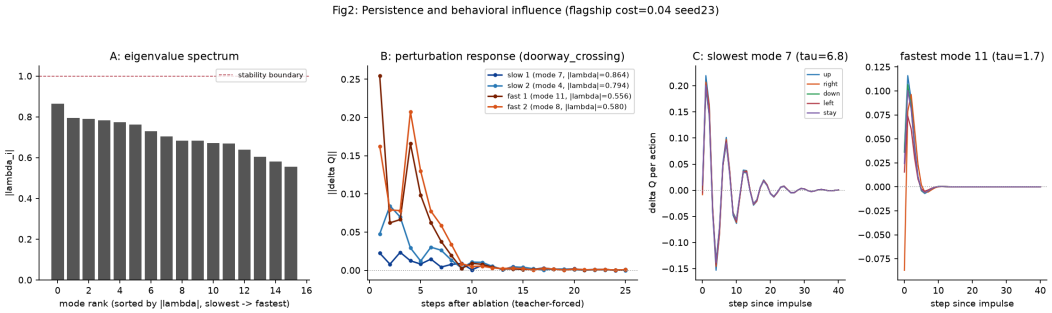


Figure 2: Persistence and behavioral influence of latent modes. Panel A shows that the learned modal spectrum is stable but spans multiple decay rates. Panel B compares teacher-forced intact and modal-ablated branches after a doorway-crossing perturbation; because subsequent observations are matched, changes in Q are attributable to the latent perturbation itself. Panel C shows representative per-action Q impulse responses for the slowest and fastest modes, illustrating that slower modes carry longer-lived action-value influence. This figure establishes persistence with behavioral influence, while Figure 3 shows where such influence is initiated and modulated across the task.

5.3 Inputs initiate latent dynamics and correction signals modulate them

Figure 3 is the central mechanistic result. It tests whether task variables drive different latent dynamics and whether the correction pathway modulates latent updates. The spatial maps report effective time constants: the number of steps for the action-value-space response to a pathway impulse to decay. These quantities measure the duration of each control pathway's effect on action values. In this sense they estimate the timescale of option-like influence: where a control input affects action for longer, the corresponding latent dynamic behaves more like a temporally extended control mode. The bottom row adds two complementary views of persistence: how time-forcing persistence changes over remaining time, and where the hidden state remains linearly decodable as position after autonomous evolution.

5.4 Latent dynamics are reusable under supplied C4 symmetry

Figure 4 tests the reuse criterion under the known rotational symmetry of Four Rooms. The same learned policy and latent dynamics are evaluated after mapping a symmetry-related goal setting into the canonical frame and rotating the selected action back. This is

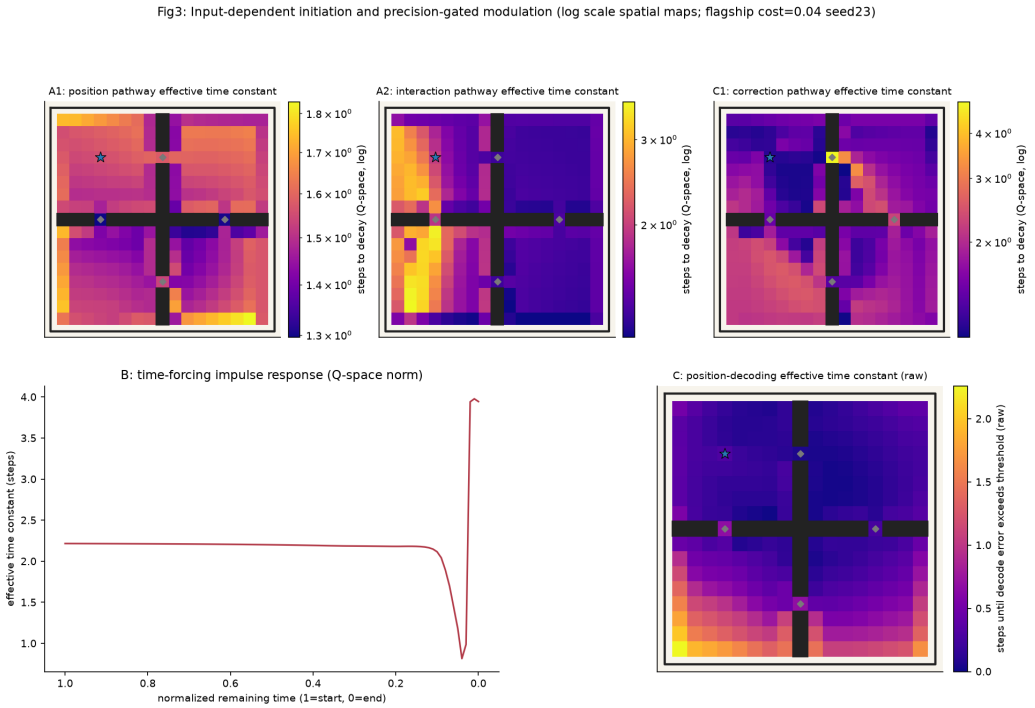


Figure 3: Input-dependent initiation and precision-gated modulation. The top row maps pathway-specific effective time constants across Four Rooms, showing that the duration of action-value influence is spatially structured rather than globally fixed. Position, interaction, and correction pathways emphasize different regions, so task inputs do not merely inject a uniform latent update; they select where different pathways produce longer- or shorter-lived effects on action. Panel B shows the Q-space effective time constant of the time-forcing pathway as remaining time changes, separating temporal content from spatial pathway effects. Panel C maps position-decoding persistence: after a real reset-style forward step at each cell, the hidden state is allowed to evolve autonomously and the plotted value is the number of steps until linear position-decoding error exceeds threshold. Together these panels show where control effects last long enough to serve as option-like temporal abstraction, and where they remain short-lived local corrections.

a constrained reuse test: it does not show learned symmetry discovery or arbitrary novel-context generalization.

5.5 Criteria-to-evidence summary

Table 3 summarizes how the Results support each operational criterion. The current evidence assignment treats each criterion as strongly supported by the corresponding main-text analysis.

6 Conclusion

We introduced an operational account of option-like temporal abstraction in recurrent latent dynamics. Rather than asking whether an agent selects an explicit option identity, we asked whether learned latent dynamics satisfy measurable criteria associated with temporally extended control: persistence, behavioral influence, input-dependent initiation, modulation or interruption, reuse, and absence of explicit option labels. This framing makes the claim falsifiable: each criterion requires a corresponding main-text analysis, and unsupported criteria should be reported directly rather than hidden in appendix diagnostics.

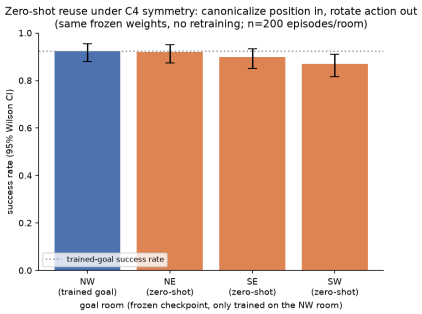


Figure 4: Reuse under supplied C_4 symmetry. The final figure should report rotated-goal performance or action agreement using the same learned checkpoint, with no new option labels, policy heads, or retraining. The result supports constrained reuse under known symmetry, not learned symmetry discovery.

Table 3: Criteria-to-evidence summary.

Criterion	Main evidence	Result summary	Claim strength
Persistence	Fig. 2A–B / Fig. 1C	Modal decay and ablation responses persist across multiple steps	Strong
Behavioral influence	Fig. 2C	Modal ablations alter action values under matched observations	Strong
Input-dependent initiation	Fig. 3A1–C1 / B	Pathway maps and time-forcing response show spatially and temporally varying persistence	Strong
Modulation / interruption	Fig. 1D / Fig. 3C1	Correction-pathway persistence and correction ablations change action-relevant dynamics	Strong
Flexible effective duration	Fig. 1C–D	Duration and correction metrics vary across precision-demand conditions	Strong
Reuse	Fig. 4	Canonical rotation reuses the same policy and dynamics without retraining	Strong
No option labels	Model section	No option IDs, option policies, or termination heads are present	Strong

Within Four Rooms, the model provides a controlled test bed for this account. Its latent dynamics are input-driven, its correction pathway gives residual signals a context-dependent route into latent state, and its reuse test is scoped to a supplied C_4 symmetry. The strongest version of the evidence is not task competence alone, but the conjunction of action-linked perturbations, flexible duration under temporal cost or observation reliability, correction-pathway controls, and reuse under symmetry. These results support a scoped mechanistic

claim: recurrent systems can implement reusable, temporally extended control through latent dynamics without selecting from an explicit library of options.

The scope is intentionally limited. The results do not establish literal Sutton options, learned symmetry discovery, biological precision, arbitrary novel-context generalization, or benchmark superiority over generic recurrent agents. They instead provide a diagnostic framework for asking when latent dynamics play the functional role of temporal abstraction. Future work should test the same criteria in additional navigation geometries, compare how different recurrent architectures satisfy them, and examine whether similar duration and correction metrics predict behavior in less controlled environments.

References

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 31, 2017.

Steven L. Brunton, Marko Budisic, Eurika Kaiser, and J. Nathan Kutz. Modern Koopman theory for dynamical systems. SIAM Review, 64(2):229–340, 2022. doi: 10.1137/21M1401243.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. Neural Computation, 5(4):613–624, 1993. doi: 10.1162/neco.1993.5.4.613.

Karl Friston. The free-energy principle: A unified brain theory? Nature Reviews Neuroscience, 11(2):127–138, 2010. doi: 10.1038/nrn2787.

Jakub Limanowski, Rick A. Adams, James Kilner, and Thomas Parr. The many roles of precision in action. Entropy, 26(9):790, 2024. doi: 10.3390/e26090790.

Marlos C. Machado, Marc G. Bellemare, and Michael Bowling. A laplacian framework for option discovery in reinforcement learning. In Proceedings of the 34th International Conference on Machine Learning, pp. 2295–2304. PMLR, 2017.

Kimberly L. Stachenfeld, Matthew M. Botvinick, and Samuel J. Gershman. The hippocampus as a predictive map. Nature Neuroscience, 20(11):1643–1653, 2017. doi: 10.1038/nn.4650.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence, 112 (1–2):181–211, 1999. doi: 10.1016/S0004-3702(99)00052-1.

Elise van der Pol, Daniel E. Worrall, Herke van Hoof, Frans A. Oliehoek, and Max Welling. MDP homomorphic networks: Group symmetries in reinforcement learning. In Advances in Neural Information Processing Systems, volume 33, 2020.

Dian Wang, Robin Walters, and Robert Platt. SO(2)-equivariant reinforcement learning. In International Conference on Learning Representations, 2022.

Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. Journal of Nonlinear Science, 25(6):1307–1346, 2015. doi: 10.1007/s00332-015-9258-5.