LEARNING COARSER MARKOV STATE ABSTRACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

State abstractions in reinforcement learning create compact representations, and increase sample-efficiency, by keeping information in observations that is relevant for decision-making while discarding other regularities. Recently introduced Markov state abstractions (MSAs) define reward-free sufficient conditions under which the learned representation is Markov. However, current implementations of the MSA objectives satisfy a stronger (kinematic inseparability) condition, which results in over-refined abstractions that preserve all transition regularities in the observation. We show for the first time how to align the training objective with the original theoretical MSA conditions and thus build coarser abstractions. This objective drives the representation to satisfy sufficient Markov conditions, but does not, by itself, guarantee minimality. We therefore provide an additional heuristic to push the representation towards minimality. Our experiments in pixel-based domains show that the learned abstractions recover the underlying complexity of the problem, which improves downstream performance.

1 INTRODUCTION

Although deep reinforcement learning (Sutton & Barto, 2018) has shown tremendous progress over the last decade, it still requires impractically many interactions to solve tasks with high-dimensional observations. This performance penalty is especially vexing in domains with a *representation gap* between the agent's high-dimensional observations and the natural low-dimensional task state—common in general agents that must solve many tasks using the same observation space (Konidaris, 2019).

One approach to resolving this gap is learning a *state abstraction* (Li et al., 2006) that compresses the agent's high-dimensional observation into a much more compact, task-relevant state. The key question here is how to formulate a learning rule or loss function that (approximately) preserves some important property of the state space (Li et al., 2006; Abel et al., 2016), with the most common target being an abstraction that preserves the ability to accurately represent a model of the task. Such model-preserving abstractions can also support learning value functions and policies, and can be used across multiple tasks in the same environment.

Allen et al. (2021) recently defined sufficient conditions for a learned state abstraction to retain the Markov property (and therefore be model-preserving), and derived a corresponding loss function; the resulting algorithm learned compact state abstractions that support efficient learning in high-dimensional tasks. However, while those conditions are sufficient for a reward-free model-preserving abstraction, the derived loss function is sufficient to satisfy a stronger property—kinematic inseparability. The resulting representation therefore retains unnecessary information because every regularity in the environment will be preserved; for example, the agent will retain the information required to reproduce lighting conditions, or irrelevant objects in the background. These kinds of common high-dimensional distractors are exactly the information that state abstraction is intended to discard.

We therefore introduce a new objective function that more closely matches the original conditions in Allen et al. (2021), and stops once model-preserving conditions are met. We prove that there is a unique minimal solution among a specific abstraction class, and derive a simple objective that pushes the abstraction towards minimality. Our experiments in pixel-level environments show that the proposed objective learns abstractions that reflects the intrinsic complexity of the domains. These

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.

learned abstractions are robust to visual distractors—maintaining performance regardless of the level of visual distraction—whereas other baselines quickly lose performance.

2 BACKGROUND

We use the Markov decision process (MDP) formalism to formulate the reinforcement learning problem. An MDP is a 5-tuple (S, A, R, T, γ) where S is the state space, A is the action space, $R(s, a, s')$ is the reward that the agent gets when it arrives at state s' when executed action a in state s, dynamics of which is governed by the transition function $T(s' \mid s, a)$, and $\gamma \in [0, 1]$ is the discount factor controlling the decision-making horizon.

State abstractions State abstractions can be viewed from the lens of learning invariant representations where the invariance definitions are tied to the MDP language. Formally, a state abstraction is a mapping $\phi : S \to \overline{S}$ where states that satisfy a specific set of conditions are aggregated into the same abstract state $\bar{s} \in \overline{S}$. Types of abstractions range from finer (approximately) model-preserving abstractions (which are the least compact, but are suitable for multi-task learning) where we keep the transition dynamics of reward-relevant paths to coarser abstractions where we only (approximately) preserve the optimal policy (which are the most compact, but support only representing, and not necessarily learning, that policy) (Li et al., 2006; Abel et al., 2016).

Markov state abstractions An abstraction ϕ is said to be Markov if and only if knowing past actions and past abstract states does not change the belief about the current ground observation. More formally, for all $s \in S$, $a \in A$, and $k \geq 1$, the following equality holds:

$$B^{\pi(k)}_{\phi,t}(s \mid z_t, \{a_{t-i}, z_{t-i}\}_{i=1}^{k}) = B^{\pi}_{\phi,t}(s \mid z_t) \tag{1}$$

where $B^{\pi}_{\phi,t}$ and $B^{\pi(k)}_{\phi,t}$ are belief distributions induced by abstraction $\phi(x) = z$, the policy $\pi \in \Pi_\phi$, and its k-repeated applications. While this is a simple definition for checking the Markov conditions, it requires conditioning on arbitrarily past abstract states, which makes it impractical. Allen et al. (2021) introduced sufficient conditions for being Markov based only on state, action, and next state tuples, which allows one to turn them into objectives for learning a Markov state abstraction (MSA). If the following two conditions hold for any policy π in the class Π_ϕ and for every timestep t:

1. *Inverse model*: $I^{\pi}_{\phi,t}(a \mid s, s') = I^{\pi}_{\phi,t}(a \mid z, z')$ for all $s, s' \in S$, $a \in A$ where $z = \phi(s)$ and $z' = \phi(s')$,
2. *Density ratio*: $\frac{p^{\pi}_t(s'|z)}{p^{\pi}_t(s')} = \frac{p^{\pi}_t(z'|z)}{p^{\pi}_t(z')}$ for all $s, s' \in S$, $a \in A$ where $z = \phi(s)$, $z' = \phi(s')$, $p^{\pi}_t(s' \mid z) = \sum_{s \in S} B^{\pi}_{\phi,t}(s \mid z) p^{\pi}_t(s' \mid s)$, and $p^{\pi}_t(z') = \sum_{s' \in \phi^{-1}(z')} p^{\pi}_t(s')$,

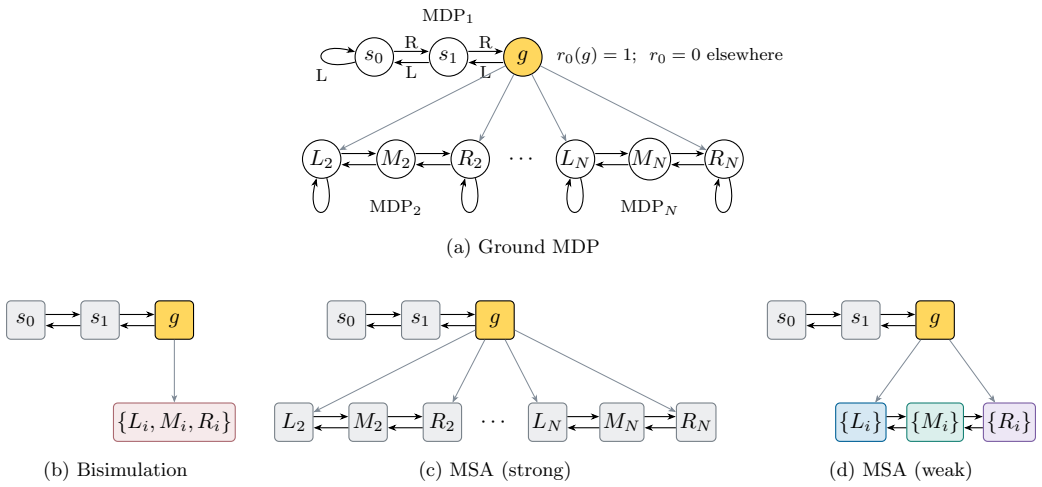
then ϕ is a Markov state abstraction.

Intuitively, these conditions dictate that (1) the executed action should be explained equally well either with ground or abstract states, and (2) the pointwise mutual information between s' and z should be equal to that of z' and z.

3 WEAK AND STRONG MSA CONDITIONS

Consider the simple scenario in Figure 1 showing a ground MDP together with three different abstractions. There is a simple chain-walk domain where the agent can move left or right, and enter one of the several sub-MDPs (MDP_2 to MDP_n) from state g with equal probability in Figure 1a. Sub-MDPs can be thought of as copies of the same underlying mechanics with different observations. For simplicity, we only considered three states in each sub-MDP, but this could be arbitrarily complex—e.g., a robot manipulating objects on the table in different lighting conditions or background.

Model-preserving abstractions such as bisimulation (Dean & Givan, 1997) require the reward model and the one-step transitions to abstract states to be the same for states with the same abstraction. Figure 1b shows the abstraction formed under this criteria. While this is a good—and minimal—abstraction for deriving a policy to reach g, if the task changes to reaching a state in one of the



(a) Ground MDP

(b) Bisimulation (c) MSA (strong) (d) MSA (weak)

Figure 1: Sufficient abstractions with each class. (a) The ground MDP with $3 \times N$ states in which the agent can move left or right between states and, from state g, can jump to one of the sub-MDPs below with equal probability. (b) Model-preserving conditions create four abstract states: three states for MDP$_1$ and another abstract state covering the rest. (c) MSA (strong) recovers the original MDP without aggregating behaviorally equivalent MDPs. (d) MSA (weak) aggregates MDP$_2$ to MDP$_n$ together, allowing a coarser abstraction by using our proposed objectives.

sub-MDPs below, the agent would have to construct a new abstraction. Learning abstractions using bisimulation is especially challenging under sparse rewards as the conditions are trivially satisfied with a single abstract state when the agent has not observed any reward during initial exploration. Learned bisimulation metrics (Zhang et al., 2021) typically require additional regularizing conditions to prevent early collapse (Kemertas & Aumentado-Armstrong, 2021).

Markov state abstractions defined in Section 2 address this issue by considering the reward-free, multi-task setting. The inverse action model prevents representation collapse and, when combined with the density ratio condition, defines sufficient conditions for a Markov representation. Nevertheless, objectives used in the implementation of MSA in Allen et al. (2021) target a stronger density ratio condition, which implies the stricter kinematic inseparability condition defined in Misra et al. (2020). Consider the two different density ratios both of which define sufficient conditions for a Markov state abstraction (when combined with the inverse model),

$$\frac{p_t^\pi(s' \mid s)}{p_t^\pi(s')} = \frac{p_t^\pi(z' \mid z)}{p_t^\pi(z')} \quad \text{(strong)}, \qquad \frac{p_t^\pi(s' \mid z)}{p_t^\pi(s')} = \frac{p_t^\pi(z' \mid z)}{p_t^\pi(z')} \quad \text{(weak)}. \tag{2}$$

where $z = \phi(s)$, $z' = \phi(s')$, and $\pi \in \Pi_\phi$. When the strong condition holds, the abstract ratio $p(z' \mid z)/p(z')$ matches the ground ratio $p(s' \mid s)/p(s')$, which encodes the pointwise mutual information between consecutive observations. As such, any abstraction that meets the strong condition would retain all the regularities in the observation, regardless of their relevance to the task. On the other hand, in the weak condition, because the ground ratio considers the abstract state, the equality can hold without explaining all the similarity between consecutive observations. In fact, the weak condition, when used in isolation, allows representation collapse with a single abstract state that subsumes all states.

We show abstractions learned with the strong and the weak conditions (in combination with the inverse model objective) in Figures 1c and 1d, respectively. MSA (strong) recovers the exact ground MDP without aggregating any states even though sub-MDPs 2 through N are copies of one another. MSA (weak) groups the three states, L_i, M_i, and R_i across copies, and can describe how the agent moves without determining the chain that it is currently in as the action dynamics do not differ between copies.

However, MSA (weak) sufficient conditions do not guarantee the minimal representation. For instance, the abstraction in Figure 1c also satisfies the weak condition. To find coarser abstractions,

we consider a specific class: abstractions that are finer than ϕ_{inv} (one that only satisfies the inverse model) and Markov.

Theorem 1 (Coarsest Markov Abstraction)**.** *Let ϕ_{inv} be a state abstraction that satisfies the inverse model condition, $I_t^{\pi}(a \mid s, s') = I_{\phi_{inv},t}^{\pi}(a \mid z, z')$, for all $\pi \in \Pi_{\phi_{inv}}$ and all t, $s, s' \in S$, $a \in A$ where $z = \phi_{inv}(s)$ and $z' = \phi_{inv}(s')$, $\succeq$ be the* finer than *relation defined in Li et al. (2006), and let*

$$\mathcal{H}(\phi_{inv}) := \{\phi \succeq \phi_{inv} : \phi \text{ is Markov over } \Pi_{\phi}\}.$$

Then $\mathcal{H}(\phi_{inv})$ has a unique coarsest Markov state abstraction ϕ^* such that $\phi_j \succeq \phi^*$ for all $\phi_j \in \mathcal{H}(\phi_{inv})$.

Corollary 1. *Let ρ be any distribution on S with full support, and for a deterministic abstraction ϕ let $I_\rho(S; \phi(S))$ denote the mutual information between $s \sim \rho$ and $z = \phi(s)$. Then ϕ^* is the unique minimizer of $I_\rho(S; \phi(S))$ over $\phi \in \mathcal{H}(\phi_{inv})$. Any Markov refinement of ϕ_{inv} other than ϕ^* keeps strictly more information about the ground state than is needed to be Markov.*

Proofs are provided in Appendix A. We should note that the minimality is over the abstractions that are finer than ϕ_{inv} and Markov, not minimality over all Markov abstractions. Also, these results are for deterministic abstractions in a finite state space while in the next section, we introduce a stochastic encoder to penalize $I(s; z)$, which relaxes the deterministic setting of the theorem. In summary, Theorem 1 states that there is a unique coarsest element in the defined set, and Corollary 1 gives us a practical metric that can be minimized to steer the representation to the minimal representation, which we show in the next section.

4 Learning Coarser Markov State Abstractions

We now introduce an objective for learning coarser abstractions using reward-free transitions (s, a, s'), approximating the MSA (weak) conditions with learned estimators, an inverse model, and an additional rate penalty that provides the preference for coarseness.

4.1 Training Objectives

Density-ratio matching. To target the weak density ratio condition, we train two binary classifiers: a ground-successor classifier $D_g(y \mid z, s')$ and an abstract-successor classifier $D_z(y \mid z, z')$, where both condition on the current abstract state z. Positive pairs ($y = 1$) are observed transitions, while the negative pairs ($y = 0$) are sampled from the empirical successor marginal to replace the successor. Each classifier is fit using binary cross-entropy with equal weight on two classes. With balanced classes, their Bayes-optimal posterior odds are density ratios (Cranmer et al., 2015), and equality of these posteriors therefore corresponds to equality of the two density ratios.

$$\frac{D_g^\star(1 \mid z, s')}{D_g^\star(0 \mid z, s')} = \frac{p(s' \mid z)}{p(s')}, \qquad \frac{D_z^\star(1 \mid z, z')}{D_z^\star(0 \mid z, z')} = \frac{p(z' \mid z)}{p(z')}. \tag{3}$$

The encoder then matches the abstract classifier to the ground classifier:

$$\mathcal{L}_{\text{match}} = \mathbb{E}_{(z,\, s',\, z') \sim \mathcal{M}_\phi} \Big[D_{\text{KL}}\big(\text{sg}[D_g(\cdot \mid z, s')] \,\big\|\, D_z(\cdot \mid z, z') \big) \Big]. \tag{4}$$

Inverse model. We retain the inverse prediction objective used by Allen et al. (2021). An inverse model $f_\psi(a \mid z, z')$ predicts the action that generated the transition and is trained by:

$$\mathcal{L}_{\text{inv}} = \mathbb{E}_{(s,a,s') \sim \mathcal{D},\; z \sim q_\phi(\cdot \mid s),\; z' \sim q_\phi(\cdot \mid s')} \big[-\log f_\psi(a \mid z, z') \big]. \tag{5}$$

Rate. By Corollary 1, we know that the coarsest abstraction has the minimal $I(S; Z)$ among abstractions satisfying the weak conditions. Therefore, we use a stochastic encoder $q_\phi(z \mid s) = \mathcal{N}(\phi(s), \sigma^2 I)$ with learned mean $\phi(s)$ and fixed $\sigma > 0$, together with a learned prior $r_\eta(z)$, and introduce a penalty on the rate:

$$\mathcal{L}_{\text{rate}} = \mathbb{E}_s \big[D_{\text{KL}}\big(q_\phi(z \mid s) \,\|\, r_\eta(z) \big) \big] = I_q(S; Z) + D_{\text{KL}}\big(q_\phi(z) \,\|\, r_\eta(z) \big) \;\geq\; I_q(S; Z), \tag{6}$$

which is a variational upper bound on $I(S; Z)$ (Alemi et al., 2017).

Training objective The encoder minimizes a weighted combination of the three terms:

$$\mathcal{L}_{\text{enc}} = \alpha\,\mathcal{L}_{\text{inv}} + \beta\,\mathcal{L}_{\text{match}} + \lambda\,\mathcal{L}_{\text{rate}}, \tag{7}$$

where α, β, and λ determine their relative contributions. Training alternates between two phases. First, we fit the estimators: the classifiers, the inverse model, and the prior with the encoder fixed. Then we update the encoder using Equation 7, with the estimator parameters fixed.

4.2 Learned Representations

We first evaluate whether the weak objective preserves the distinctions required by a Markov abstraction while eliminating regularities in a visual chain-walk domain (Figure 2). The domain has two chain types that differ only in their dynamics: the success probability of the chosen action. We denote a combination of the position in the chain and the chain type by a *cell*, and a Markov abstraction must retain all six cells. Each observation is a sampled 28×28 pixel MNIST digit, identifying the cell, rendered on one of the four backgrounds. The background could be thought of as a factored sub-MDP having its own dynamics, changing from one texture to another in a predictable manner, independently from the executed action. As such, background identity helps predicting the next image, but it does not provide additional information about the next cell.

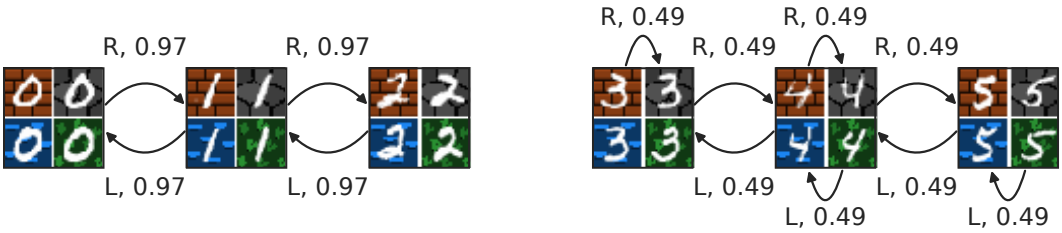


Figure 2: A chain-walk domain with six underlying states. Each chain has three states with different transition dynamics (types). Each state is rendered with four different backgrounds. With probability 0.05, the next position is sampled uniformly within the current chain. Otherwise, moves always succeed in type 1 and succeed with probability 0.5 in type 2.

This persistent distraction separates the two ratio conditions. The weak condition is suitable for merging the four backgrounds of a cell, since $p(s' \mid z)/p(s')$ equals $p(z' \mid z)/p(z')$ when backgrounds are averaged over and represented by z. Conversely, the same merge violates the strong condition because it conditions on a single observation s, and the background of s determines the background of s'. Therefore, $p(s' \mid s)$ is nonzero only for successors that share the background of s. Consequently, the six-cell abstraction satisfies the inverse and weak density-ratio conditions, while the strong density-ratio condition requires all cell-background combinations.

Figure 3 illustrates the two-dimensional abstract states learned by MSA (strong) and MSA (weak) at successive training steps. Observations of the same cell are separated into sub-clusters by background, and this is persistent under MSA (strong). Under MSA (weak), observations of the same cell collapse as training proceeds, with background-specific groups increasingly overlapping. The weak objective thus allows abstraction to aggregate visual variants of the same cell, while preserving the distinctions between positions and chain types, as shown in Figure 4a.

We quantify this behavior using 5-nearest-neighbor probes on held-out observations (Figure 4b). The cell probe predicts the six position-type combinations, while the background probe distinguishes the four backgrounds within each cell. Cell accuracy increases under both objectives, but with MSA (weak), this improvement is accompanied by a decrease in background accuracy toward the chance level. Under MSA (strong), the background accuracy instead approaches 100%. Collectively, these trends show that MSA (weak) preserves cell distinctions needed to predict transitions between cells, while making it difficult to recover background identity, which does not affect these transitions.

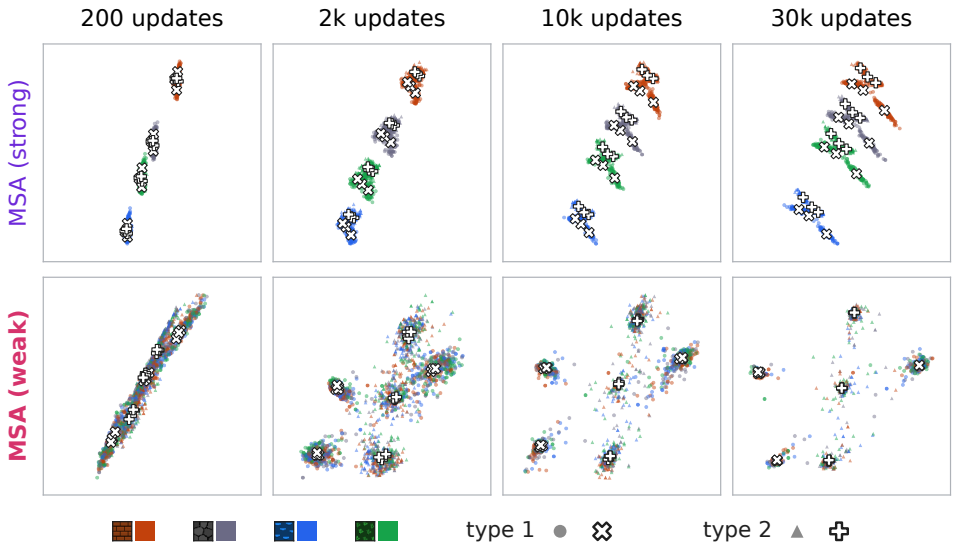


Figure 3: Visualization of abstractions learned by MSA (strong) and MSA (weak) at consecutive training updates. Colors correspond to the background type. Circles and triangles show type 1 and type 2 observations, respectively, and markers show the median of each cell-background group. MSA (weak) merges the backgrounds of a cell, whereas MSA (strong) keeps a separate cluster for each background.

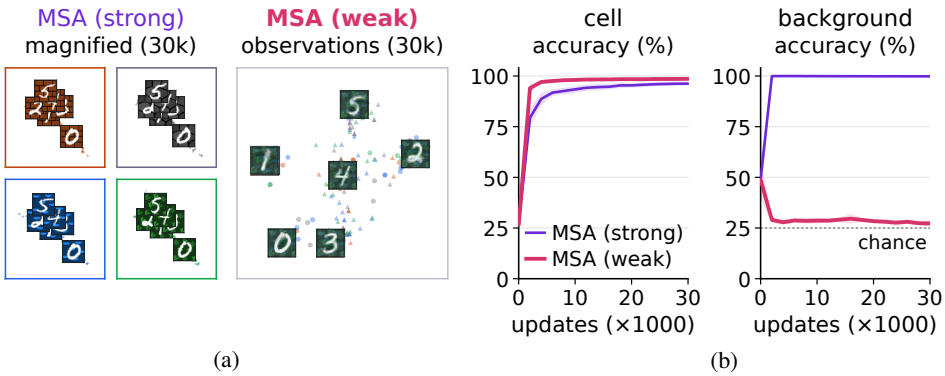


Figure 4: (a) Final representations from Figure 3. Each sub-cluster of MSA (strong) is magnified separately for better visualization. Observations are placed in their cell-background medians. (b) Balanced 5-nearest-neighbor accuracy for the cell identity and for the background identity within each cell (mean $\pm$ standard deviation over 10 seeds). Both objectives yield higher cell accuracy, while the background accuracy decreases under MSA (weak) and approaches 100% under MSA (strong). The opposing trends show that MSA (weak) keeps cells detectable while abstracting away the background.

5 Using Coarser MSAs in Deep Reinforcement Learning

We now evaluate the weak objective in two online settings, in which the underlying decision problem is fixed while task-irrelevant visual variation increases. To compare the abstraction objectives, we fix the RL agent, including its architecture, hyperparameters, and interaction budget in each experiment.

5.1 Online abstraction learning for visual navigation

We first evaluate our approach on a goal-conditioned navigation task in a 3×3 maze, where the cells are separated by walls, and the agent has four movement actions. The agent observes images of its current cell and the goal. To create a task-irrelevant visual structure, we make visually distinct copies of the maze with identical dynamics and reward. The background texture is specific to each copy and fixed within the episode. The background color also changes through a four-phase cycle with every action the agent takes, and both factors are predictable from the current observation.

We use goal-conditioned Double DQN Van Hasselt et al. (2016) as the base algorithm, and all methods share the same architecture. We learn abstraction online alongside RL, and the encoder is trained on the corresponding abstraction objectives for MSA (weak) and MSA (strong). We measure the final success rate of each method with a varying number of copy mazes, as in Figure 5. For a single maze, all methods perform near ceiling. As the number of maze copies increases, the success rates of MSA (strong) and DQN decline substantially, whereas MSA (weak) maintains a high success rate. Figure 6 shows the average return over 100,000 interactions. We see that DQN can only learn in the single-copy scenario and fails to show any progress when the distractors increase.

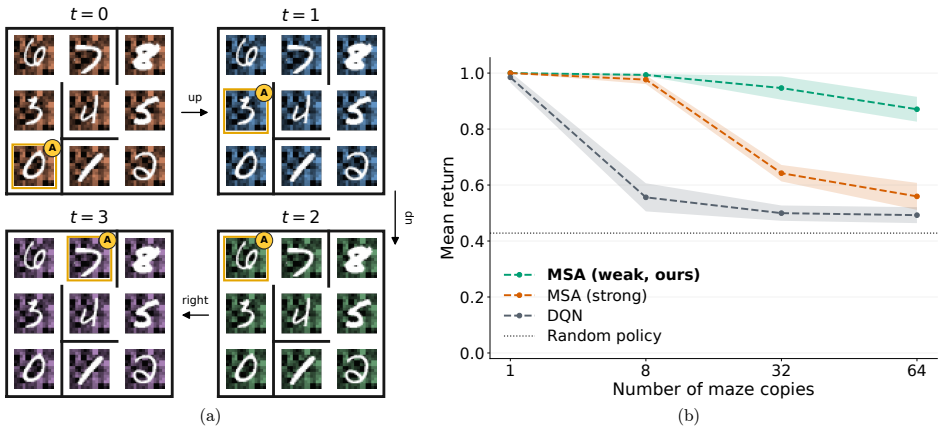


Figure 5: (a) Example trajectory through four color phases (A: agent). (b) Final test success rate versus the number of maze copies (5 seeds; points and shaded bands denote the mean ± 2 standard errors (SE))

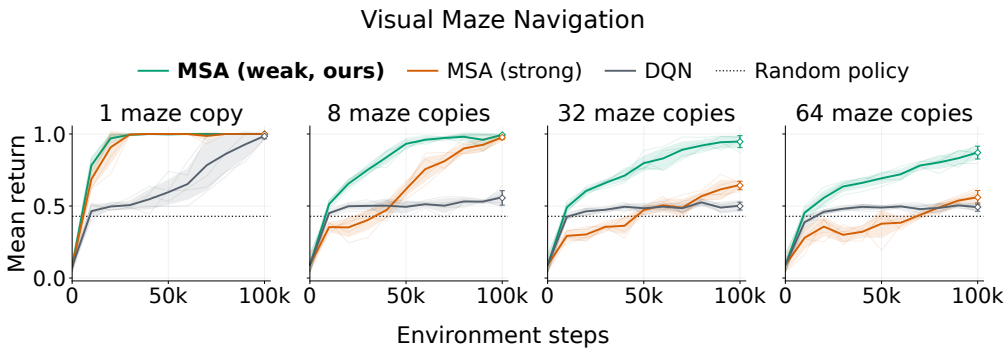


Figure 6: Learning curves for visual maze navigation with different numbers of maze copies (Figure 5). Lines and shaded bands show mean return ± 2 SE across 5 seeds.

5.2 Online abstraction learning for continuous control

Next, we evaluate our approach on image-based continuous control tasks: Walker Walk and Cheetah Run from the DeepMind Control Suite (Tassa et al., 2018). We follow the color distractors of the

Distracting Control Suite (Stone et al., 2021) (shown in Figure 7) to introduce task-irrelevant visual variation. The physics, dynamics, and rewards are unchanged, while the color deviates from its original value between episodes. In this experiment, we use RAD (Laskin et al., 2020b), a Soft Actor-Critic Haarnoja et al. (2018) trained on pixel inputs with random data augmentation, as an RL agent, and we compare abstraction objectives by attaching each to the same RAD agent.

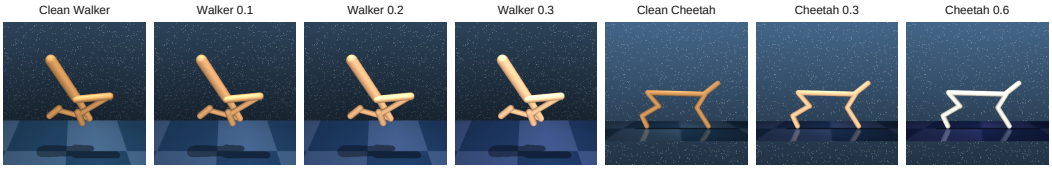


Figure 7: Example observations from Distracting Control Suite (Stone et al., 2021) with increasing levels of color distractors. Although looking perceptually similar, methods quickly degrade under increasing distraction.

In Figure 8, we compare the final return of each method as the distractor strength increases. All methods have similar performance under the clean setting, but as the distractor strength increases, other baselines drop more quickly than MSA (weak).

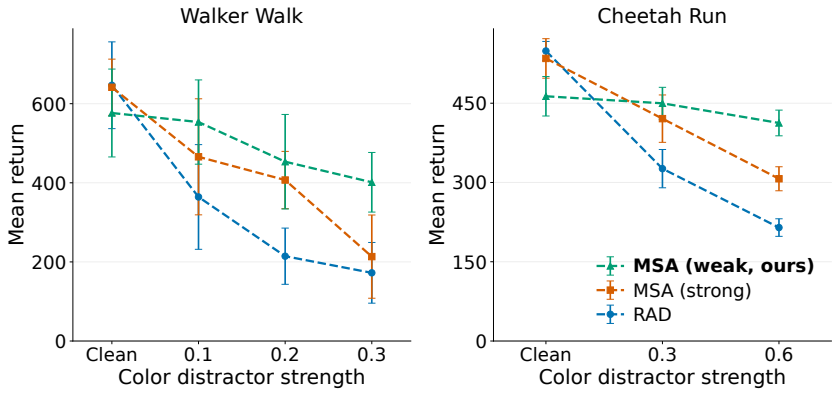


Figure 8: Final return versus color-distractor strength for continuous control tasks: Walker Walk (left) and Cheetah Run (right), (10 seeds; points and error bars denote the mean ± 2 SE).

6 Related Work

State abstraction and bisimulation. State abstractions in reinforcement learning have a long history from exact model minimization using bisimulation (Dean & Givan, 1997; Givan et al., 2003) to the taxonomy of abstraction types (Li et al., 2006) and their value-loss guarantees (Abel et al., 2016). Bisimulation metrics (Ferns et al., 2011; Castro, 2020) relax the equivalence to a distance, and DBC (Zhang et al., 2021) learns an encoder that approximately follows such a metric from pixels. Bisimulation metrics struggle under sparse rewards as the representation can quickly collapse under no reward. Additional regularization techniques are used to prevent collapse under bisimulation (Kemertas & Aumentado-Armstrong, 2021).

Discarding distractors. Several lines of work aim to remove observation components that are irrelevant to control. In the Block-MDP setting with exogenous noise, multi-step inverse models provably recover the agent-controllable state and filter distractors with their own dynamics (Efroni et al., 2022; Lamb et al., 2022). Levine et al. (2025) extend this to learning the endogenous state and its dynamics from a single trajectory, under a fast-mixing assumption on the exogenous component. Our weak condition removes the same kind of persistent distractor with a one-step objective. Contrastive and predictive objectives (Oord et al., 2018; Laskin et al., 2020a; Stooke et al., 2021;

Schwarzer et al., 2021) maximize agreement between consecutive latents, and therefore, like the strong MSA objective, retain any predictable regularity. On the other hand, inverse-dynamics objectives (Agrawal et al., 2016; Pathak et al., 2017) retain only what is needed to identify actions and are part of our objective as well.

Latent world models. Reconstruction-based world models (Hafner et al., 2019; 2020) must encode every observation factor, so distractors are removed only with reward (Fu et al., 2021), a factorized noise model (Wang et al., 2022), or by replacing reconstruction (Deng et al., 2022). Latent-prediction objectives (Schwarzer et al., 2021; Hansen et al., 2024) avoid reconstruction but need anti-collapse regularizers that reward additional information, so a predictable distractor is kept because it is cheap to predict—the same behavior as the strong MSA condition. DeepMDP (Gelada et al., 2019) learns a latent transition model together with a reward model and shows that the latent distances bound the bisimulation metric; the reward loss is what anchors the representation, so, like bisimulation, it is task-specific and uninformative under sparse reward. Our objective differs in both respects: the inverse model, not a variance term, prevents collapse, and the weak condition is backward-looking, asking only that the abstract successor retain what the ground successor says about the abstract predecessor, which is why persistent but action-irrelevant structure can be discarded.

Prior approaches remove distractors using reward, a structural noise model, or the Ex-BMDP assumption; ours does so from reward-free one-step conditions, and Theorem 1 gives the resulting abstraction a well-defined minimality target.

7 Conclusion

We showed that the objective used in Markov state abstractions targets a stricter condition, and how this could lead to over-refined abstractions that not only keep the task-relevant information but any regularity between successive states. We focused on the original weak density ratio condition and derived objectives that satisfy this criteria. While inverse model and the weak density ratio conditions are sufficient for a Markov representation, they are not the necessary conditions, as such, minimality is not guaranteed. Nevertheless, solutions that are finer than the abstraction that only preserves the inverse model provided us a practical target for finding coarse Markov state abstractions. Our experiments on pixel-based domains and continuous control tasks show that MSA (weak) (1) learns coarser abstractions, (2) which results in sample efficiency and faster learning, and (3) is more robust to distractors.

AI use statement

In this work, we used generative AI tools to

1. implement methods,
2. formulate mathematical claims,
3. provide critical ingredients for proving mathematical claims,
4. assist in the writing of proofs,
5. develop theoretical models or conceptual frameworks.

We have not used generative AI tools to

1. generate synthetic data sets,
2. propose or refine hypotheses,
3. design or provide feedback on research methodology or experiments,
4. clean and reformat dataset,
5. assist with translation,
6. support qualitative and thematic data analysis,
7. interpret results.

Additionally, we used generative AI tools to

1. create or modify scientific figures or images,
2. create or edit software code,
3. discover research topics or identify gaps,
4. brainstorming,
5. summarize or analyse existing literature,
6. draft parts of a research paper,
7. edit a research paper to improve readability
8. identify relevant literature.

We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

D. Abel, D.E. Hershkowitz, and M.L. Littman. Near optimal behavior via approximate state abstraction. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 2915–2923, 2016.

Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. *Advances in Neural Information Processing Systems*, 29, 2016.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

C. Allen, N. Parikh, O. Gottesman, and G.D. Konidaris. Learning Markov state abstractions for deep reinforcement learning. In *Neural Information Processing Systems 34*, pp. 8229–8241, 2021.

Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10069–10076, 2020.

Kyle Cranmer, Juan Pavez, and Gilles Louppe. Approximating likelihood ratios with calibrated discriminative classifiers. *arXiv preprint arXiv:1506.02169*, 2015.

Thomas Dean and Robert Givan. Model minimization in markov decision processes. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pp. 106–111, 1997.

Fei Deng, Ingook Jang, and Sungjin Ahn. Dreamerpro: Reconstruction-free model-based reinforcement learning with prototypical representations. In *International conference on machine learning*, pp. 4956–4975. PMLR, 2022.

Yonathan Efroni, Dylan J Foster, Dipendra Misra, Akshay Krishnamurthy, and John Langford. Sample-efficient reinforcement learning in the presence of exogenous information. In *Proceedings of Thirty Fifth Conference on Learning Theory*, pp. 5062–5127, 2022.

Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi Jaakkola. Learning task informed abstractions. In *International Conference on Machine Learning*, pp. 3480–3491, 2021.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International conference on machine learning*, pp. 2170–2179. PMLR, 2019.

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. *Artificial intelligence*, 147(1-2):163–223, 2003.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 10–15 Jul 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2555–2565, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Nick Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, volume 2024, pp. 47376–47405, 2024.

Mete Kemertas and Tristan Aumentado-Armstrong. Towards robust bisimulation metric learning. *Advances in Neural Information Processing Systems*, 34:4764–4777, 2021.

G. Konidaris. On the necessity of abstraction. *Current Opinion in Behavioral Sciences*, 29:1–7, 2019.

Alex Lamb, Riashat Islam, Yonathan Efroni, Aniket Didolkar, Dipendra Misra, Dylan Foster, Lekan Molu, Rajan Chari, Akshay Krishnamurthy, and John Langford. Guaranteed discovery of controllable latent states with multi-step inverse models. *arXiv preprint arXiv:2207.08229*, 2022.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*, pp. 5639–5650. PMLR, 2020a.

Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19884–19895, 2020b.

Alexander Levine, Peter Stone, and Amy Zhang. Learning a fast mixing exogenous block mdp using a single trajectory. In *International Conference on Learning Representations*, volume 2025, pp. 60545–60591, 2025.

L. Li, T.J. Walsh, and M.L. Littman. Towards a unified theory of state abstraction for MDPs. In *Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics*, 2006.

Dipendra Misra, Mikael Henaff, Akshay Krishnamurthy, and John Langford. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In *Proceedings of the Thirty-Seventh International Conference on Machine Learning*, pp. 6961–6971, 2020.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 2778–2787, 2017.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021.

Austin Stone, Oscar Ramirez, Kurt Konolige, and Rico Jonschkowski. The distracting control suite–a challenging benchmark for reinforcement learning from pixels. *arXiv preprint arXiv:2101.02722*, 2021.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International conference on machine learning*, pp. 9870–9879. PMLR, 2021.

R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2018.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Tongzhou Wang, Simon Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian. Denoised MDPs: Learning world models better than the world itself. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 22591–22612, 2022.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021.

A PROOFS

Let us first restate Markov state abstraction conditions defined in Allen et al. (2021) (and also mentioned in Section 2).

Condition 1 (Inverse model)**.** *$I_t^\pi(a \mid s, s') = I_{\phi,t}^\pi(a \mid z, z')$ for all $\pi \in \Pi_\phi$ and all t, and $s, s' \in S$, $a \in A$ where $z = \phi(s)$ and $z' = \phi(s')$.*

Condition 2 (Density ratio)**.** *$\frac{p_t^\pi(s' \mid z)}{p_t^\pi(s')} = \frac{p_t^\pi(z' \mid z)}{p_t^\pi(z')}$ for all $\pi \in \Pi_\phi$ and all t, and $s, s' \in S$, where $z = \phi(s)$, $z' = \phi(s')$, $p_t^\pi(s' \mid z) = \sum_{s \in S} B_{\phi,t}^\pi(s \mid z) p_t^\pi(s' \mid s)$, and $p_t^\pi(z') = \sum_{s' \in \phi^{-1}(z')} p_t^\pi(s')$.*

If ϕ is a state abstraction that satisfies these two conditions, then ϕ is a Markov state abstraction. Conversely, if ϕ is a Markov state abstraction satisfying Condition 1, it satisfies Condition 2. Both proofs are given in Allen et al. (2021).

Let us now turn our attention to proving that among refinements of ϕ_{inv}, there exists a unique coarsest state abstraction ϕ^*. Throughout the proof, we assume $p_t^\pi(s) > 0$ for all s, t and all policies considered, or otherwise restrict S to the support.

Lemma 1 (Inverse-model holds in refinements)**.** *If ϕ_{inv} satisfies Condition 1, then any $\phi \succeq \phi_{inv}$ satisfies Condition 1 for all $\pi \in \Pi_\phi$.*

Proof. Consider the ground and abstract inverse models:

$$I_t^\pi(a \mid s, s') = \frac{T(s' \mid a, s)\pi_t(a \mid s)}{\sum_{\tilde{a}} T(s' \mid \tilde{a}, s)\pi_t(\tilde{a} \mid s)}, \quad I_{\phi,t}^\pi(a \mid z, z') = \frac{T_{\phi,t}^\pi(z' \mid a, z)\pi_t(a \mid z)}{\sum_{\tilde{a}} T_{\phi,t}^\pi(z' \mid \tilde{a}, z)\pi_t(\tilde{a} \mid z)}$$

where, for $\pi \in \Pi_\phi$, the abstract transition probability is defined as

$$T_{\phi,t}^\pi(z' \mid a, z) := \sum_{s \in \phi^{-1}(z)} \sum_{s' \in \phi^{-1}(z')} T(s' \mid a, s) \frac{\pi_t(a \mid s) B_{\phi,t}^\pi(s \mid z)}{\sum_{\tilde{s} \in S} \pi_t(a \mid \tilde{s}) B_{\phi,t}^\pi(\tilde{s} \mid z)} \tag{8}$$

and $\phi^{-1}(z) := \{s \in S : \phi(s) = z\}$ is the grounding set of z. Call a ground pair (s, s') *reachable* if $c_{s,s'} := \sum_a T(s' \mid a, s) > 0$.

A policy-free form of Condition 1. Let π^u be the uniform policy, $\pi_t^u(a \mid s) = 1/|A|$ for all t, s, which is in $\Pi_{\phi_{\text{inv}}}$. Under π^u the policy factors cancel and

$$I_t^{\pi^u}(a \mid s, s') = \frac{T(s' \mid a, s)}{c_{s,s'}}, \quad I_{\phi_{\text{inv}},t}^{\pi^u}(a \mid z, z') = \frac{T_{\phi_{\text{inv}},t}^{\pi^u}(z' \mid a, z)}{\sum_{\tilde{a}} T_{\phi_{\text{inv}},t}^{\pi^u}(z' \mid \tilde{a}, z)} =: v_{z,z'}(a).$$

Note that the right-hand side depends on abstract states (z, z') (and on t), not on ground states (s, s') inside each abstract state's grounding set. As ϕ_{inv} satisfies Condition 1, for every reachable $(s, s') \in \phi_{\text{inv}}^{-1}(z) \times \phi_{\text{inv}}^{-1}(z')$,

$$\frac{T(s' \mid a, s)}{c_{s,s'}} = v_{z,z'}(a) \quad \text{for all } a \in A. \tag{9}$$

Thus, the normalized likelihood $T(s' \mid \cdot, s)/c_{s,s'}$, although defined per ground pair, takes the single value $v_{z,z'}$ on all reachable pairs of the block pair (z, z').

Restriction to the blocks of ϕ. Let (w, w') be a pair of abstract states in ϕ. Since $\phi \succeq \phi_{\text{inv}}$, there are abstract states z, z' in ϕ_{inv} with $\phi^{-1}(w) \subseteq \phi_{\text{inv}}^{-1}(z)$ and $\phi^{-1}(w') \subseteq \phi_{\text{inv}}^{-1}(z')$. Hence Equation 9 holds for every reachable $(s, s') \in \phi^{-1}(w) \times \phi^{-1}(w')$. Let $v := v_{z,z'}$ for the common vector. Equivalently, for every $(s, s') \in \phi^{-1}(w) \times \phi^{-1}(w')$, reachable or not,

$$T(s' \mid a, s) = c_{s,s'} v(a) \quad \text{for all } a \in A. \tag{10}$$

since both sides vanish for unreachable pairs.

The abstract likelihood of ϕ is proportional to v. Notice that for any $\pi \in \Pi_\phi$ and t, $\pi_t(a \mid s) = \pi_t(a \mid w)$ for every $s \in \phi^{-1}(w)$, so the posterior weight in Equation 8 collapses to the belief:

$$\frac{\pi_t(a \mid s) B_{\phi,t}^\pi(s \mid w)}{\sum_{\tilde{s}} \pi_t(a \mid \tilde{s}) B_{\phi,t}^\pi(\tilde{s} \mid w)} = \frac{\pi_t(a \mid w) B_{\phi,t}^\pi(s \mid w)}{\pi_t(a \mid w) \underbrace{\sum_{\tilde{s}} B_{\phi,t}^\pi(\tilde{s} \mid w)}_{=1}} = B_{\phi,t}^\pi(s \mid w). \tag{11}$$

Combining Equation 10 with the definition of the abstract transition function in Equation 8,

$$\begin{aligned} T^{\pi}_{\phi,t}(w' \mid a, w) &= \sum_{s \in \phi^{-1}(w)} \sum_{s' \in \phi^{-1}(w')} T(s' \mid a, s) B^{\pi}_{\phi,t}(s \mid w) \\ &= \sum_{s \in \phi^{-1}(w)} \sum_{s' \in \phi^{-1}(w')} c_{s,s'} v(a) B^{\pi}_{\phi,t}(s \mid w) = K v(a), \\ K &:= \sum_{s \in \phi^{-1}(w)} \sum_{s' \in \phi^{-1}(w')} c_{s,s'} B^{\pi}_{\phi,t}(s \mid w). \end{aligned}$$

K depends on π, t, w, w' but not on a, and $K > 0$ whenever the abstract pair (w, w') has positive probability, which is exactly when $I^{\pi}_{\phi,t}(\cdot \mid w, w')$ is defined.

Inverse models are equal. For the same $\pi \in \Pi_\phi$ and t, and any reachable $(s, s') \in \phi^{-1}(w) \times \phi^{-1}(w')$, using Equation 10 and $\pi_t(\cdot \mid s) = \pi_t(\cdot \mid w)$ on the ground side and the previous step on the abstract side,

$$\begin{aligned} I^{\pi}_t(a \mid s, s') &= \frac{c_{s,s'} v(a) \pi_t(a \mid w)}{\sum_{\tilde{a}} c_{s,s'} v(\tilde{a}) \pi_t(\tilde{a} \mid w)} \\ &= \frac{v(a) \pi_t(a \mid w)}{\sum_{\tilde{a}} v(\tilde{a}) \pi_t(\tilde{a} \mid w)} \\ &= \frac{K v(a) \pi_t(a \mid w)}{\sum_{\tilde{a}} K v(\tilde{a}) \pi_t(\tilde{a} \mid w)} \\ &= I^{\pi}_{\phi,t}(a \mid w, w'). \end{aligned}$$

Since $\pi \in \Pi_\phi, t, (w, w')$ and the reachable pair (s, s') were arbitrary, ϕ satisfies Condition 1 for all $\pi \in \Pi_\phi$. □

Given Lemma 1 and Theorems 1 and 2 of Allen et al. (2021), a refinement $\phi \succeq \phi_{\text{inv}}$ is a Markov state abstraction if and only if it satisfies Condition 2 for all $\pi \in \Pi_\phi$ and t. Let $\mathcal{H}(\phi_{\text{inv}})$ be the set of such refinements.

For abstractions ϕ_1, ϕ_2, let $\phi_1 \vee \phi_2$ (the *join*) be the finest abstraction coarser than both: its grounding sets are the connected components of the graph on S joining s, s'' whenever they share a ϕ_1-block or a ϕ_2-block. Hence, (a) each grounding set of the join is a disjoint union of grounding sets of ϕ_1 (and of ϕ_2), and (b) states in a common join-block are linked by a chain of single ϕ_1- or ϕ_2-steps; (c) $\Pi_{\phi_1 \vee \phi_2} = \Pi_{\phi_1} \cap \Pi_{\phi_2}$; and (d) the join of two refinements of ϕ_{inv} refines ϕ_{inv}.

Lemma 2 (Joins preserve Condition 2)**.** *Fix π and t. If ϕ_1, ϕ_2 satisfy Condition 2 for (π, t), so does $\phi := \phi_1 \vee \phi_2$.*

Proof. Multiplying Condition 2 by $p^{\pi}_t(z)$ gives $p^{\pi}_t(s', z)/p^{\pi}_t(s') = p^{\pi}_t(z', z)/p^{\pi}_t(z')$, i.e. $p^{\pi}_t(z \mid s') = p^{\pi}_t(z \mid z')$ for all z and all $s' \in \phi^{-1}(z')$. Since $p^{\pi}_t(z \mid z')$ is a convex combination of the values $p^{\pi}_t(z \mid s')$, $s' \in \phi^{-1}(z')$, this is equivalent to

$$\phi(s'_1) = \phi(s'_2) \implies \sum_{s \in \phi^{-1}(z)} p^{\pi}_t(s \mid s'_1) = \sum_{s \in \phi^{-1}(z)} p^{\pi}_t(s \mid s'_2) \quad \forall s'_1, s'_2 \in S \text{ and } \forall z \in Z_\phi, \tag{12}$$

where Z_ϕ is the abstract state space induced by ϕ. It thus suffices to show Equation 12 for ϕ, given it for ϕ_1 and ϕ_2.

Let $\mathcal{G} = \phi^{-1}(z)$ for some $z \in Z_\phi$ and suppose $\phi_1(s'_1) = \phi_1(s'_2)$. By (a), $\mathcal{G} = \phi_1^{-1}(z_1) \sqcup \cdots \sqcup \phi_1^{-1}(z_m)$ for some $z_1, \ldots, z_m \in Z_{\phi_1}$, so by Equation 12 for ϕ_1,

$$\sum_{s \in \mathcal{G}} p^{\pi}_t(s \mid s'_1) = \sum_{j=1}^{m} \sum_{s \in \phi_1^{-1}(z_j)} p^{\pi}_t(s \mid s'_1) = \sum_{j=1}^{m} \sum_{s \in \phi_1^{-1}(z_j)} p^{\pi}_t(s \mid s'_2) = \sum_{s \in \mathcal{G}} p^{\pi}_t(s \mid s'_2);$$

likewise for a ϕ_2-step. If $\phi(s'_1) = \phi(s'_2)$, apply this along the chain of (b). □

Proof of Theorem 1.

Proof. $\mathcal{H}(\phi_{\text{inv}})$ is non-empty since the ground abstraction ϕ_0 is a valid Markov state abstraction. It is closed under join: for $\phi_1, \phi_2 \in \mathcal{H}(\phi_{\text{inv}})$ and $\phi = \phi_1 \vee \phi_2$, (d) gives $\phi \succeq \phi_{\text{inv}}$, and for any $\pi \in \Pi_\phi$ and t, (c) gives $\pi \in \Pi_{\phi_1} \cap \Pi_{\phi_2}$, so Lemma 2 applies. As $\mathcal{H}(\phi_{\text{inv}})$ is finite, $\phi^* := \bigvee_{\phi \in \mathcal{H}(\phi_{\text{inv}})} \phi$ lies in it and is refined by every member; two such elements would refine each other, so ϕ^* is unique. Being a refinement of ϕ_{inv} in $\mathcal{H}(\phi_{\text{inv}})$, ϕ^* satisfies Condition 1 by Lemma 1 and Condition 2 by membership for all $\pi \in \Pi_{\phi^*}$, hence is a Markov state abstraction by Theorem 1 of Allen et al. (2021). □

Proof of Corollary 1

Proof. For deterministic ϕ, $I_\rho(S; \phi(S)) = H_\rho(\phi(S))$. Let $\phi \in \mathcal{H}(\phi_{\text{inv}})$ with $\phi \neq \phi^*$. By Theorem 1, $\phi \succeq \phi^*$ and $\phi \neq \phi^*$, so some grounding set of ϕ^* is split by ϕ into two or more grounding sets, each of positive ρ-mass by full support. By the grouping property of entropy,

$$H_\rho(\phi(S)) = H_\rho(\phi^*(S)) + \sum_{z \in Z_{\phi^*}} \rho\big(\phi^{*-1}(z)\big)\, H_\rho\big(\phi(S) \mid \phi^*(S) = z\big),$$

and the conditional entropy is strictly positive on any split grounding set. Hence $H_\rho(\phi(S)) > H_\rho(\phi^*(S))$. □

B Learning Curves

Figures 9 and 10 show learning curves corresponding to Figure 8. Lines and shaded bands show mean return ±2 SE across 10 seeds, after a five-point moving average within each seed.

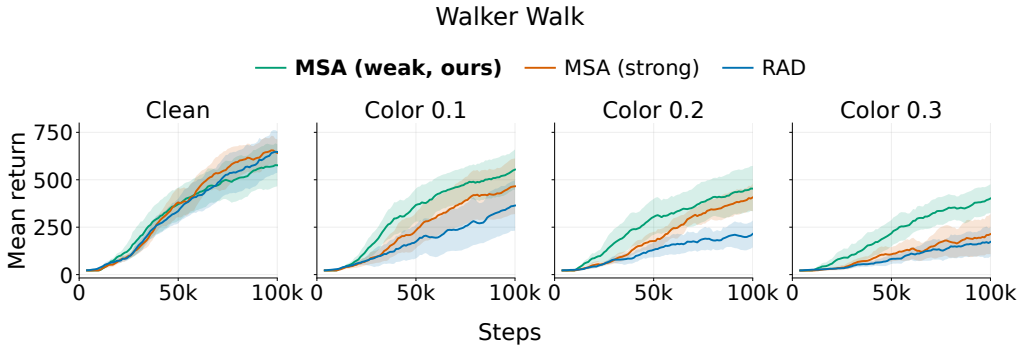


Figure 9: Learning curves for the Walker Walk task.

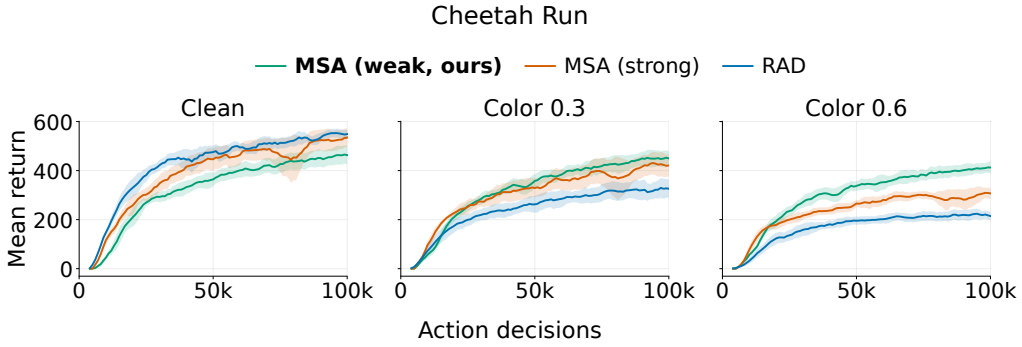


Figure 10: Learning curves for the Cheetah Run task.

C ADDITIONAL REPRESENTATION VISUALIZATIONS

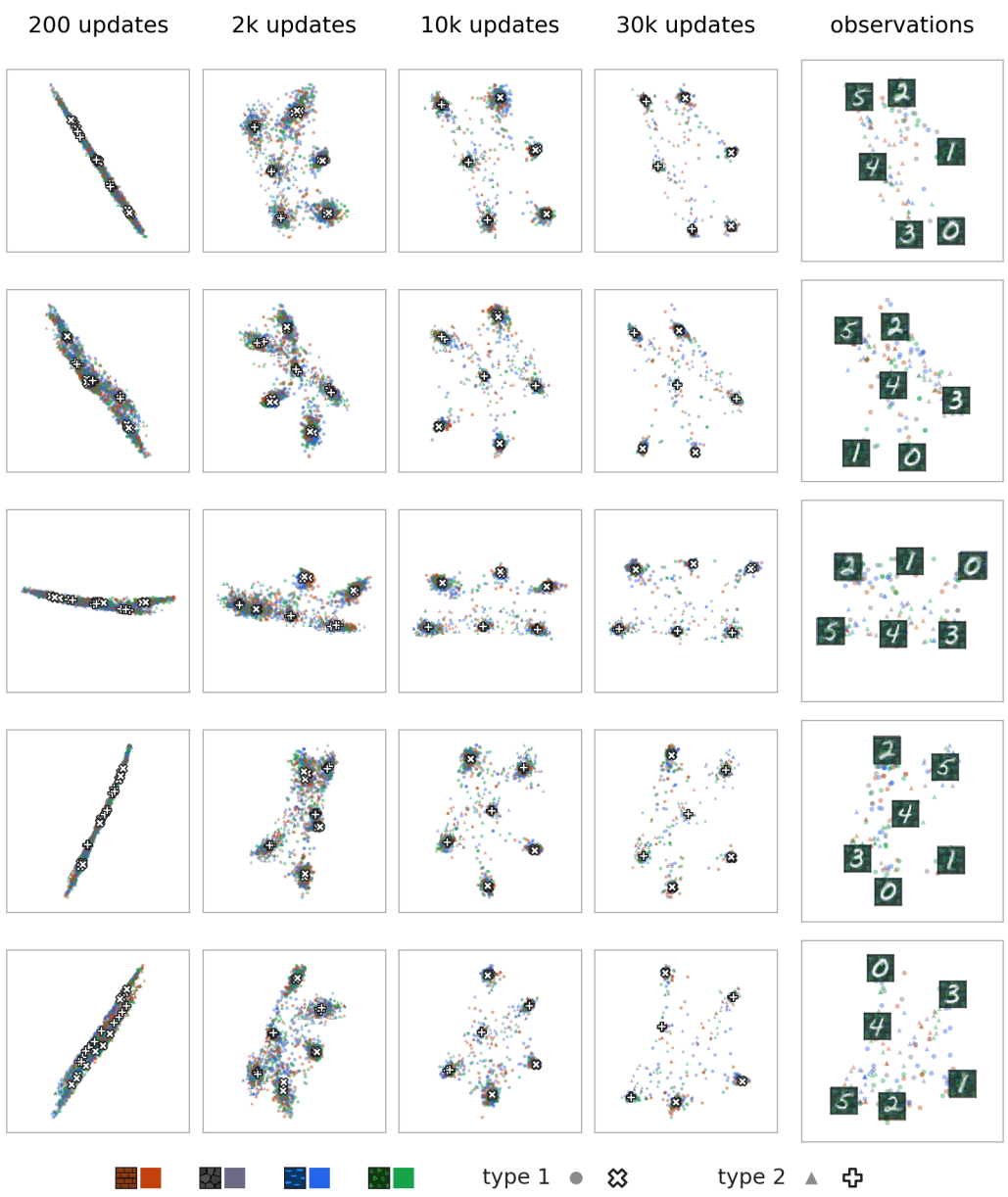


Figure 11: MSA (weak) representations.

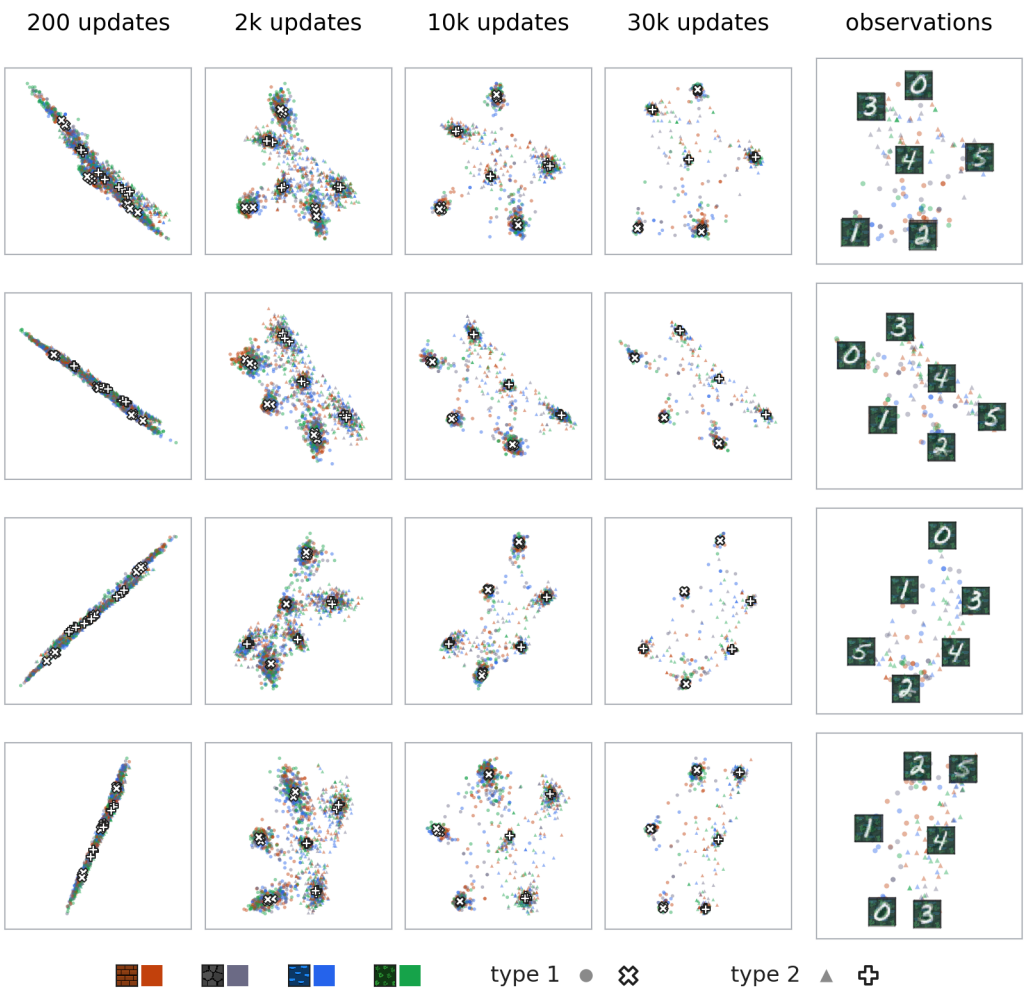


Figure 12: MSA (weak) representations (continued).



Figure 13: MSA (strong) representations.

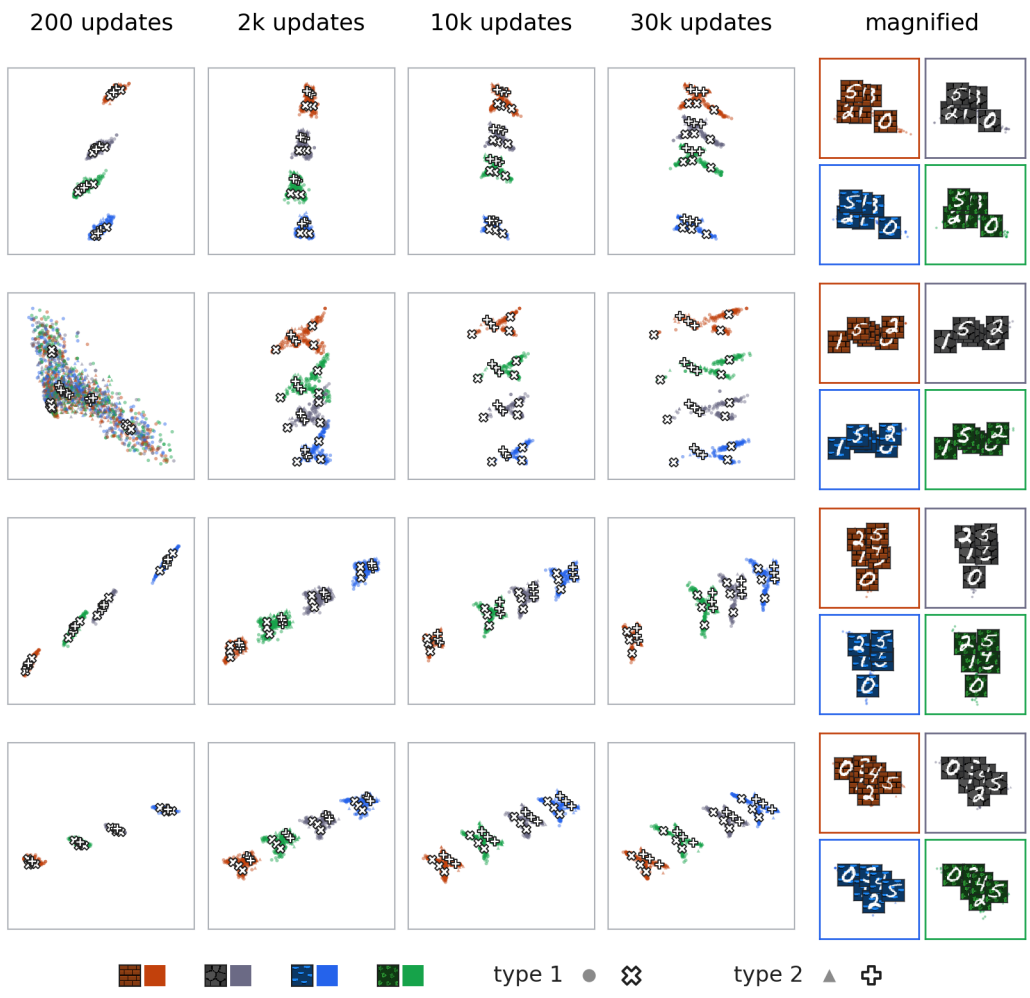


Figure 14: MSA (strong) representations (continued).

D Experimental Details

MSA (weak).

Setting	Chain	Maze	Control
Inverse coefficient α	1	1	10
Matching coefficient β	1	1	10
Rate coefficient λ	0.1	0.01	0.01
Gaussian-mixture components	8	8	16
Negatives per transition	10	10	10
Rate samples per observation	4	4	4
Auxiliary fits per encoder update	1	1	1
Inverse loss	Cross-entropy	Cross-entropy	Gaussian NLL

Visual chain.

Setting	Value
Encoder hidden layers	Two 128-unit layers with ReLU
Pixel preprocessing	Flattened; scaled to $[0, 1]$
Training transitions	10,000, balanced across states
Data-collection actions	Alternating left/right at the middle; inward at boundaries
Optimizer / learning rate	AdamW / 10^{-4}
Weight decay / batch size	10^{-4} / 128
Gradient clipping	Norm 1
MSA (Strong) loss weights	Inverse 1; contrastive 1; smoothness 0

Visual maze.

Setting	Value
Start and goal	Uniformly sampled distinct cells
Reward / episode limit	1 at the goal, 0 otherwise / 20 actions
Blocked actions	Leave the cell unchanged
Image preprocessing	28×28 RGB; flattened; scaled to $[-0.5, 0.5]$
Encoder / Q-head widths	$2352 \to 256 \to 128 \to 8$ / $16 \to 256 \to 4$
Hidden activation	ReLU
Optimizer / learning rate	AdamW / 10^{-4}
Weight decay / batch size	10^{-4} / 128
Replay capacity / discount	100,000 / 0.99
MSA (strong) loss weights	Inverse 1; contrastive 1; smoothness 1
Evaluation	Greedy; every 10,000 steps; all distinct start–goal pairs, copies, and initial phases

Continuous control.

Setting	Walker Walk	Cheetah Run
Action repeat	2	4
Encoder image size	84×84	108×108
Replay augmentation	Random crop	Random translation
Actor / critic / encoder / auxiliary learning rate	10^{-3}	2×10^{-4}
MSA initial random actions	10,000	5,000
MSA catch-up updates	9,000	4,000
MSA (strong) smoothness coefficient	10	30

Shared setting	Value
Frame stack	Three RGB frames
Encoder	Four convolutional layers; 32 filters; 3×3 kernels; strides $(2, 1, 1, 1)$; 50-dimensional output
Actor / critic hidden layers	Two 1,024-unit ReLU layers each
Optimizer / batch size	Adam / 128
Replay capacity / discount	100,000 / 0.99
RAD initial random actions	1,000
Evaluation	10 episodes every 1,000 action selections
Final return	Average of the last five evaluations (96,000–100,000 action selections) within each seed