Université de Montréal

A Survey of State Representation Learning for Deep Reinforcement Learning

par

Ayoub Echchahed

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Discipline

December 23, 2024

© Ayoub Echchahed, 2024

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

A Survey of State Representation Learning for Deep Reinforcement Learning

présenté par

Ayoub Echchahed

a été évalué par un jury composé des personnes suivantes :

Liam Paull
(président-rapporteur)

Pablo Samuel Castro
(directeur de recherche)

Gauthier Gidel
(membre du jury)

Résumé

Sommaire: Les méthodes d'apprentissage de représentations sont essentielles pour traiter les défis liés aux espaces d'observations complexes dans les problèmes de prise de décision séquentielle. Récemment, plusieurs approches ont été développées pour apprendre des représentations en apprentissage par renforcement visuel, améliorant ainsi l'efficacité, la généralisation et les performances. Ce travail de recherche propose une analyse des principales méthodes utilisées, en examinant leurs mécanismes, avantages et limites. Six classes principales sont identifiées, analysées et comparées en fonction de leurs spécificités. La taxonomie présentée a pour objectif de clarifier et de structurer les approches existantes dans le domaine, tout en servant de guide pour les chercheurs. Les méthodes d'évaluation de la qualité des représentations sont également discutées, et des directions de recherche pertinentes pour élargir l'applicabilité des méthodes à différents contextes sont explorées.

Mots clés: Apprentissage par Renforcement, Apprentissage de Représentations, Apprentissage Profond.

Abstract

Summary: Representation learning methods are an important tool for addressing the challenges posed by complex observations spaces in sequential decision making problems. Recently, many methods have used a wide variety of types of approaches for learning meaningful state representations in visual reinforcement learning, allowing better sample efficiency, generalization, and performance. This survey aims to provide a broad categorization of these methods within a model-free online setting, exploring how they tackle the learning of state representations differently. We categorize the methods into six main classes, detailing their mechanisms, benefits, and limitations. Through this taxonomy, our aim is to enhance the understanding of this field and provide a guide for new researchers. We also discuss techniques for assessing the quality of representations, and detail relevant future directions.

Keywords: Reinforcement Learning, Representation Learning, Deep Learning.

Contents

List of tables

List of figures

List of Abbreviations

SRL	State Representation Learning
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
SSL	Self-Supervised Learning
MDP	Markov Decision Process
CDP	Contextual Decision Processes
BMDP	Block Markov Decision Process
POMDP	Partially Observable Markov Decision Process
FDM	Forward Dynamics Model
IDM	Inverse Dynamics Model

VLM	Vision-Language Model
LLM	Large Language Model
IB	Information Bottleneck
MI	Mutual Information
DA	Data Augmentation

Acknowledgments

I thank my supervisor, Pablo Samuel Castro, for his guidance and feedback.

Prologue

What is State Representation Learning? Why is it useful?

State representation learning (SRL) consists of learning to extract meaningful, task-relevant information from raw observations in decision-making systems. Consider a simulated autonomous vehicle navigating a busy urban environment, encountering diverse stimuli like traffic conditions, pedestrians, and changing weather. The role of an SRL algorithm within a decision-making system consists on distilling complex sensory inputs into compact, structured representations, prioritizing critical features while filtering out irrelevant noise.

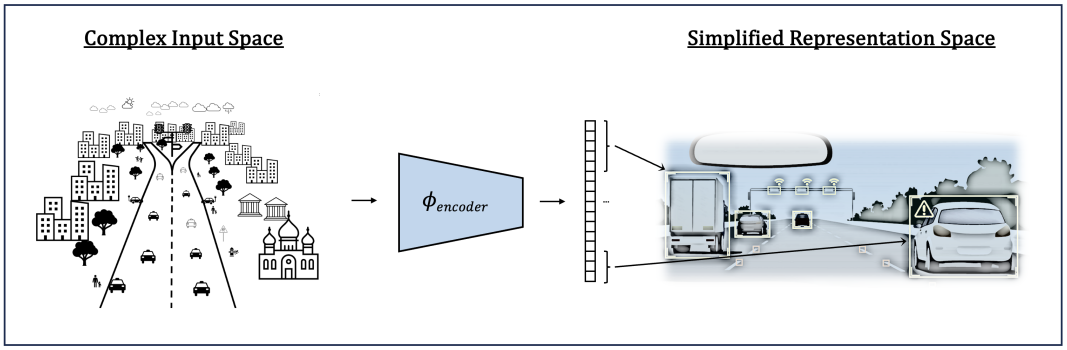

Fig. 1. SRL in action: Raw sensory inputs from a busy environment are distilled into compressed, task-relevant representations, enabling improved decision-making.

For example, while the color of buildings or the type of roadside trees might be detectable, these details are irrelevant to navigation. Instead, elements such as the positions and velocities of vehicles, traffic light statuses, road signs, and pedestrian movements are essential for effective decision-making. SRL ensures that these crucial variables are emphasized in the learned representation, enabling the policy to focus on what truly matters.

By reducing the complexity of the input space, SRL enhances learning efficiency, improves generalization, and increases robustness to environmental variations, such as altered street layouts or weather conditions. Despite these advantages, identifying and encoding relevant features is a complex task, often still performed manually in real-world applications like robotics and autonomous driving. SRL aims to automate this process, making it a cornerstone of scalable, efficient, and adaptive decision-making systems.

1. Introduction

The use of deep reinforcement learning (DRL) for complex control environments has several challenges, including the processing of large high-dimensional observation spaces. This problem, commonly referred to as the "state-space explosion", imposes severe limitations on the efficacy of traditional end-to-end RL approaches, which learn actions or value functions directly from raw sensory inputs, such as pixel observations. As environments grow increasingly complex, these end-to-end methods demonstrate progressively worse data efficiency and generalization, even in response to minor environmental changes. Overcoming these limitations is therefore crucial for addressing real-world problems with RL.

In response to these challenges, recent research has focused on decoupling representation learning from policy learning, treating them as two distinct problems. This strategy has proven useful for managing complex observations, enabling more efficient learning and improving the generalization of policies across various task settings. Specifically, state representation learning (SRL) techniques aim to transform raw, complex observations into structured, simplified representations that retain essential information for decision-making while discarding irrelevant details. This approach not only increases the learning efficiency but also enhances the robustness and adaptability of DRL agents to diverse environments.

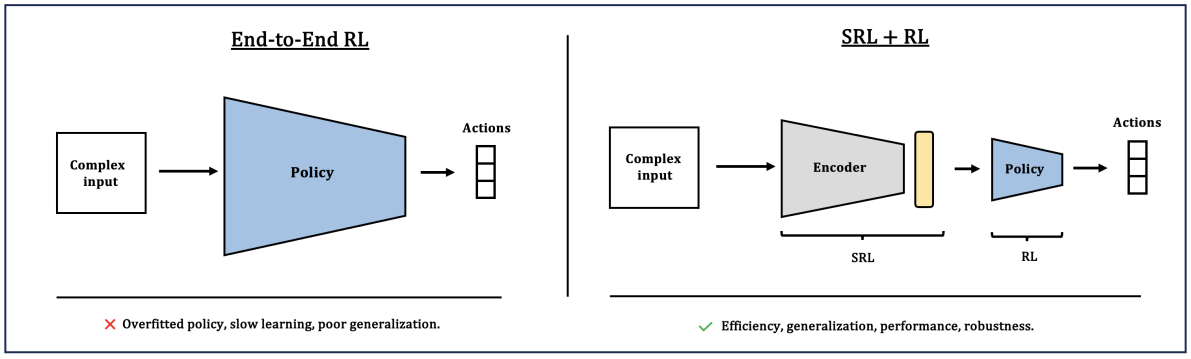

Fig. 2. Comparison of End-to-End RL (left) and SRL+RL (right). End-to-end directly maps high-dimensional inputs to actions, while SRL separates representation learning and policy learning.

Motivation. In recent years, there has been a growth in methods that integrate improved representation learning into deep RL, using various approaches. However, many works present inconsistent structuring and categorization of these approaches in their related work sections, making it challenging to obtain a clear and comprehensive understanding of the field. To our knowledge, existing surveys can provide valuable information on the topic but either do not cover the latest developments in the field or focus exclusively on specific classes of SRL methods (Lesort et al., 2018) (Ni et al., 2024) (Böhmer et al., 2015) (de Bruin et al., 2018) (Botteghi et al., 2024).

This survey builds on those works by providing an updated and structured analysis of the different approaches in state representation learning for deep RL, organizing them based on their principles and effectiveness in different scenarios. Through a detailed taxonomy, we analyze the inner-workings of these classes, highlighting their potential to improve the performance, generalization, and sample efficiency of deep-RL agents. We also explore ways of evaluating the quality of learned state representations, and discuss promising directions for the field. Overall, this survey can serve as a good resource for researchers and practitioners looking to familiarize themselves with this field.

Thesis Organization. This manuscript is structured as follows: Section 2 introduces the foundational concepts of state representation learning (SRL) within the deep reinforcement learning (DRL) framework. It defines the problem, objectives, and the characteristics of effective state representations, providing a formal basis for understanding subsequent sections. Section 3 presents the core taxonomy of SRL methods, categorizing them into six primary classes, while elaborating on their mechanisms and highlighting notable work from the literature. Section 4 addresses the critical aspect of evaluation, discussing benchmarks and metrics used to assess the quality and effectiveness of state representations, including their impact on sample efficiency, generalization, and robustness. Lastly, Section 5 explores promising directions for advancing SRL in DRL, such as multi-task learning, leveraging pre-trained visual models, and integrating multi-modal inputs.

2. Problem Definition

2.1 Formalism

Reinforcement Learning (RL) is typically modeled as a Markov Decision Process (MDP), characterized by the tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$. Here, $\mathcal{S}$ denotes the state space, and $\mathcal{A}$ denotes the action space. The transition probability function $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ defines the probability $P(s'|s, a)$ of transitioning from state s to state s' given action a, representing the environment dynamics. The reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ specifies the immediate reward $R(s, a)$ received after taking action a from state s, providing feedback on the action taken.

The objective of an RL agent is to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the expected cumulative discounted reward. Using π, the agent progressively generates experiences (s, a, r, s'), which can be organized into a trajectory τ. For each trajectory τ, the return G_t represents the total accumulated reward from time step t onwards. It is expressed as $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$, where $\gamma \in [0, 1)$ is the discount factor that prioritizes immediate rewards over future ones.

To evaluate how good a particular state or state-action pair is, we define value functions. The state value function $V^\pi(s)$ under policy π is the expected return starting from state s and following policy π, given by $V^\pi(s) = \mathbb{E}[G_t|s_t = s]$. Similarly, the action-value function $Q^\pi(s, a)$ represents the expected return starting from state s, taking action a, and subsequently following policy π, defined as $Q^\pi(s, a) = \mathbb{E}[G_t|s_t = s, a_t = a]$. Therefore, the objective of the agent can now be expressed as finding an optimal policy π^* that maximizes $Q^\pi(s, a)$.

2.2 Partial Observability

In many RL settings, full observability is rare. For example, in robotics, sensors might not capture all relevant state factors for optimal decision making in one time-step of data. A POMDP, or partially observable MDP, generalizes the notion of a MDP by accounting for situations where the agent does not have direct access to the full state $s \in \mathcal{S}$ of the environment, hence needing to rely on past observations to infer the current state.

Recurrent neural networks (RNNs) are commonly employed to address this partial observability issue, leveraging their hidden state to retain and process information from previous time steps. Another way to handle this is by concatenating the last n observations $(o_t, o_{t-1}, ..., o_{t-n+1})$ to approximate a sufficient statistic for decision-making, thus mitigating the effects of partial observability. For example, agents trained on the ALE benchmark (Bellemare et al., 2013) often employ this technique, known as 'frame stacking'.

In this survey, a POMDP framework is adopted and defined as $\mathcal{M} = \langle \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{O}$ represents the observation space. The other components mirror those of an MDP defined above, except that the agent now operates on observations $\mathcal{O}$ instead of states $\mathcal{S}$. An end-to-end RL policy $\pi : \mathcal{O} \to \Delta(\mathcal{A})$ will now map observations to action distributions. This framework is chosen over MDPs to address practical concerns: in real-world scenarios, agents rarely have access to the full environment state and instead rely on partial, high-dimensional observations that may fail to uniquely identify states. Unlike MDPs that assume full observability, POMDPs account for partial observability, making them often more realistic.

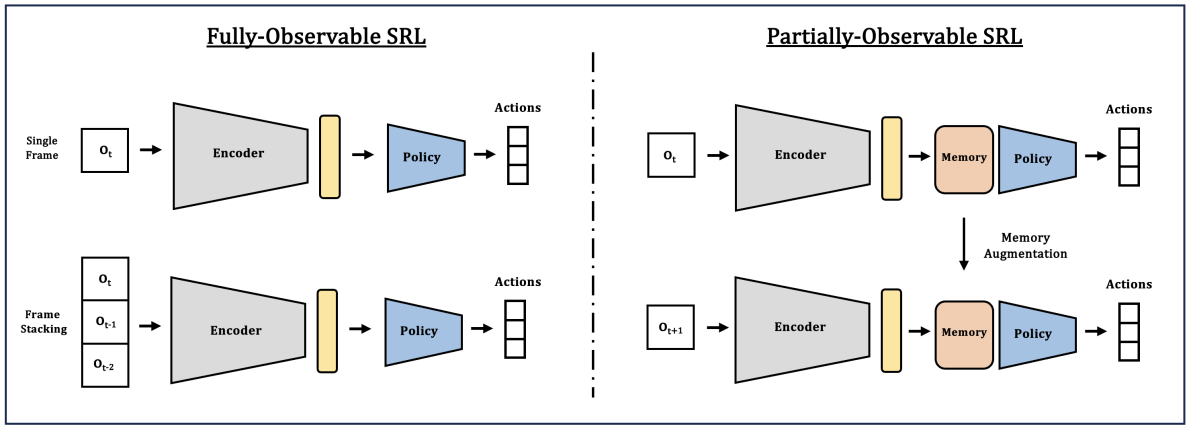

Fig. 3. Fully Observable SRL processes a single or stacked frames with all the information required for optimal decision-making, while Partially Observable SRL addresses missing information using memory modules (e.g., RNNs) or history compressors integrating past observations and actions.

In addition to POMDPs, two other decision-making frameworks not addressed here can be relevant to the subject: Contextual Decision Processes (CDPs) (Krishnamurthy et al., 2016) (Jiang et al., 2017) and Block MDPs (BMDPs) (Du et al., 2019). CDPs extend MDPs and POMDPs as a unified framework for reinforcement learning with rich observations, where agents make decisions based on rich features (context) for optimizing long-term rewards. Block MDPs differ by assuming that each observation corresponds uniquely to a latent state, ensuring that observations maintain Markovian properties. This allows for optimal representations to be inferred directly from observations without requiring additional memory or history, assuming the right encoder.

2.3 Deep Reinforcement Learning

Deep reinforcement learning (DRL) differs from traditional RL by utilizing deep neural networks to approximate value functions (or policies), either from high-dimensional inputs, or from encoded latent states. This becomes desirable when the state space is large or continuous, which is not well-suited for methods that rely on representing state-action pairs individually in a lookup table, known as tabular reinforcement learning.

DRL methods can be broadly categorized into three kind of approaches: value-based, policy-based, and actor-critic methods. Value-based methods, such as Deep Q-Networks (DQN) (Mnih et al., 2013), use a neural network to approximate the action-value function $Q(s, a)$. Policy-based methods, such as REINFORCE (Williams, 1992), directly parameterize the policy $\pi(a|s;\theta)$ and optimize it using gradient ascent on the expected cumulative reward. Actor-Critic methods combine the strengths of value-based and policy-based approaches as they maintain two networks: the actor, which updates the policy $\pi(a|s;\theta)$, and the critic, which evaluates a desired value function $Q(s, a)$ or $V(s)$.

2.4 State Representation Learning

The traditional end-to-end approach, which directly maps observations to actions, led to impressive results (Mnih et al., 2013). However, this approach becomes increasingly challenging as the complexity of the environment increases, which is why more efforts are directed towards learning better representations. Representation learning by itself can be defined as the process of automatically discovering features from raw data that are most useful for a task (Bengio et al., 2012). Although representation learning for deep-RL can be divided into state and action representation learning, the former will be the focus of this survey.

Problem: We define the objective of state representation learning (SRL) for reinforcement learning (RL) as learning a representation function $\phi^n : \mathcal{O}_0 \times \mathcal{O}_1 \times \cdots \times \mathcal{O}_n \to \mathcal{X}$, parameterized by θ_ϕ, which maps n-step observation sequences to a representation space $\mathcal{X}$, thereby allowing us to define policies $\Pi_\mathcal{X}$ over this reduced space. This encoder enables either a policy network ψ_π to compute actions $a_t = \psi_\pi(x_t)$, or a value network ψ_V to compute values $v_t = \psi_V(x_t)$, based on the representation $x_t = \phi(o_t)$, instead of directly using high-dimensional observations. The representation x_t is a vector in $\mathbb{R}^d$, where d is the chosen dimension of the representation space $\mathcal{X}$.

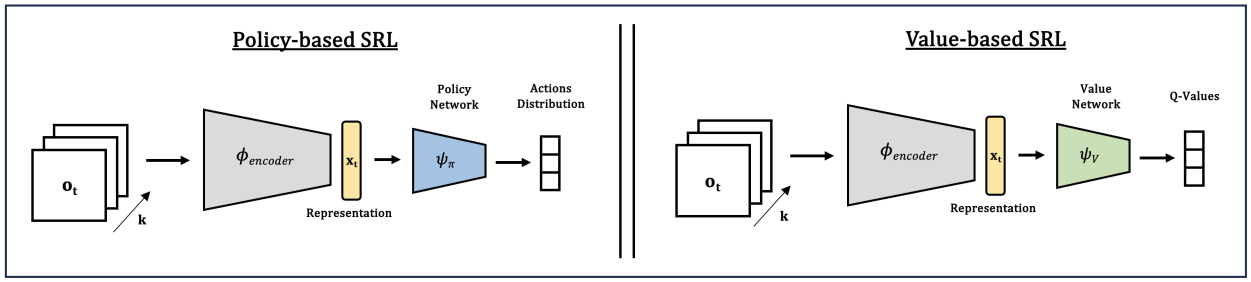

Fig. 4. Illustration of State Representation Learning (SRL) for RL, where a parametrized transformation ϕ is learned, mapping sequences of observations to representations. Two configurations are presented depending if a value-based RL approach is used (right) or a policy-based one (left).

However, not all representation functions are useful to obtain; the goal is to learn an encoder ϕ that captures the essential characteristics of effective state representations, as reviewed in the next section. Learning a good encoder simplifies the input space into a compact and relevant representation x_t, thereby (1) improving sample efficiency and performance by facilitating the function approximation process performed by the policy/value network; (2) enhancing generalization as learning a policy/value network from representations avoids the overfitting issues seen with high dimensional, unstructured, and noisy observation spaces.

In the presented taxonomy, the focus will be mostly on methods that learn state representations within a model-free online setting, where agents learn representations and policies in real-time through interactions with the environment without using an explicit model of the environment for taking actions. This differs from model-based RL, which involves learning a model of the environment's dynamics that is used for planning, enabling higher sample-efficiency at the cost of higher complexity. The offline pre-training of representations is also explored in section 5, where agents learn representations from fixed experience datasets.

Setting		Description
Pre-trained	Joint-training	Representations are learned either before reinforcement learning begins or simultaneously with RL training objectives.
Online	Offline	Learning occurs in real-time through interactions with the environment or from pre-collected datasets.
Coupled	Decoupled	Encoder parameters are optimized jointly with policy/value objectives or independently of them.
Reward-based	Reward-free	Representations are influenced by task rewards or focus on environment dynamics and visual features.
Single-task	Multi-task	Representations are learned for a specific task or shared across multiple tasks to capture common structures.
Model-free	Model-based	Representations are directly used for decision-making or integrated into a model of the environment for planning.

Table 1. Overview of the different settings for State Representation Learning in RL.

2.5 Defining Optimal Representations

A good starting point is to clearly define the ideal objectives pursued by the state representation learning methods studied in this survey. Specifically, we begin by asking: What are the characteristics that constitute effective state representations?

An optimal representation space can be defined by its ability to efficiently support policy learning for a set of downstream tasks. The learned manifold should be constrained to a low dimensionality, while remaining sufficiently informative to enable the learning of an optimal policy (or value function) using limited-capacity function approximators. If a manifold has too much information, it can slow down the learning process and hinder convergence to the optimal policy. Alternatively, a manifold with insufficient information will prevent convergence to the optimal policy (Abel, 2022). Therefore, the ideal latent manifold strikes a balance between information capacity and simplicity.

Structure: The learned representation space should be structured to encode task-relevant information while remaining invariant to noise and distractions. This means that points within a neighborhood around a representation x_t should exhibit a high degree of task-relevant similarity, which gradually diminishes as the distance from this point increases. These similarities can be encoded in the representation space using information or distances derived from observation features, environment dynamics, rewards, etc. Additionally, the encoder should remain invariant to noise, distractions, or geometric transformations that do not alter the true underlying state of the decision process.

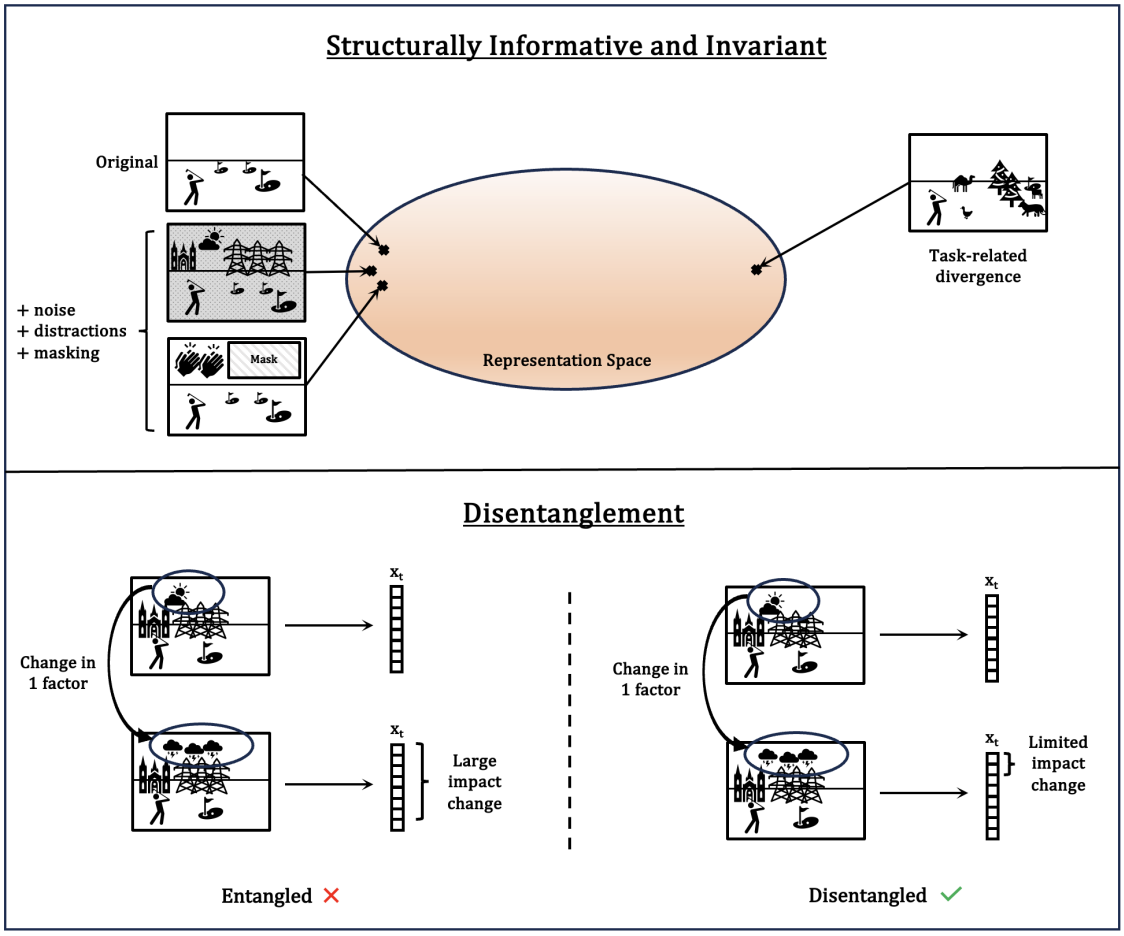

Fig. 5. Illustration of some optimal state representation properties. The top section demonstrates invariance to noise, distractions, and masking in the representation space, preserving task-relevant information. The bottom section illustrates disentanglement, where changes in individual factors ideally lead to localized latent impacts, ensuring robust and interpretable representations.

Continuity: A good latent structure should ideally ensure strong (Lipschitz) continuity of the value function within nearby representations—an important property that governs how smoothly the value function changes across the representation space (Le Lan et al., 2021). Points that are close in this space should correspond to relatively similar optimal value predictions, even if they are far apart in the input space. Strong continuity enhances learning efficiency by simplifying the function approximation process performed by the value network ψ_V. Furthermore, it promotes better generalization of the value function to unseen but nearby states. This concept of continuity also extends to the policy network ψ_π, ensuring that changes in the action distribution occur smoothly within the representation space.

Sparsity: Sparse representation learning for deep-RL refers to ways of acquiring representations from observations where only a small subset of the latent neurons are active at any given time. Enforcing sparsity constraints on the representations can allow the identification of the most relevant aspects of high-dimensional observations as it encourages the input to be well-described by a small subset of features. This enhances computational efficiency by reducing the number of active features, leading to simpler representations. Also, this helps avoid overfitting by focusing on the most relevant features, promoting better generalization. Sparse representations also improve interpretability by making it easier to understand which features are driving the agent's decisions.

Disentanglement: Acquiring disentangled state representations is useful for avoiding learning spurious correlations that can mislead RL agents. Disentanglement approaches separate the factors of variation in observations, ensuring independent and robust representations. This improves the agent's ability to generalize and adapt to new environments as a change in one observation factor only affects a subset of features in the representation, allowing the remaining features to stay stable and be used for decision-making. More on disentangled representation learning methods can be found in section (3.3).

Previous works have sought to define characteristics of effective representations and abstractions for RL. According to Wang et al. (2024b), optimal representations should exhibit high capacity, efficiency, and robustness. Abel (2022) identifies three essential criteria for state abstractions in RL: efficient decision-making, solution quality preservation, and ease of construction. These criteria stress the importance of balancing compression with performance to facilitate effective learning and planning in complex environments. Other relevant works that discuss the characteristics of good state representations for RL include (Böhmer et al., 2015), (Lesort et al., 2018), and (Botteghi et al., 2024). Similarly, definitions of optimal representations in the broader context of self-supervised learning (SSL) can often overlap with those needed for control, making research in that area valuable for RL as well.

3. Taxonomy of Methods

3.1 Overview

We categorize the representation learning methods into six distinct classes, which are presented in table 2. For each class, we provide a definition, details, benefits, limitations, and some examples of methods. While there are likely other methods in each class, the goal is not to be exhaustive, but rather to focus on the classes themselves. Additionally, some methods may be classified as hybrid, combining techniques from multiple classes.

Class	**Description**
Metric-based	Shape the representation space through a task-relevant distance metric between embeddings. They enhance generalization and efficiency by abstracting states with similar information, reducing complexity.
Auxiliary Tasks	Enhance the primary RL task with other simultaneous predictions that indirectly shape representations. These require additional parameters, but provide accelerated learning on the main task.
Augmentation	Leverage data augmentation for learning invariances to geometric and photometric transformations of observations. They do not directly learn representations, but enhance efficiency and generalization.
Contrastive	Shape the representation space by learning separate representations for different observations, and similar ones for related observations. Temporal proximity and/or transformations are used for establishing similarities.
Non-Contrastive	Construct their representation space by only minimizing the distance between the representations of similar observations. Unlike related contrastive approaches, no negative pairs are used during training.
Attention-based	Learn attention masks (Bahdanau et al., 2015) for computing scores that highlight important features of the input, helping agents disregard irrelevant details and increase the interpretability of decision-making.

Table 2. Overview of the classes presented in the taxonomy.

3.2 Metric-based Methods

Definition: Metric-based methods aim to structure the embedding space by using a metric that captures task-relevant similarities between state representations. By mapping functionally equivalent states to similar points in the latent space, these methods can enhance sample efficiency and improve policy learning. For instance, if two different visual observations in a game lead to the same downstream behavior and yield the same reward, they can be mapped to the same latent state. The environment's reward structure often plays a critical role in determining the task-relevance of the similarity metric used.

Details: The observation encoder, denoted as $\phi_\theta : \mathcal{O} \to \mathbb{R}^n$ with parameters θ, maps observations to an embedding space $\mathcal{X}$ where distances $\hat{d}(\phi_\theta(o_i), \phi_\theta(o_j))$ reflect some task-relevant similarities. For example, the distance metric $\hat{d}$ could be the L_2 norm, and the metric could be bisimulation (Ferns et al., 2012), which is introduced below. The representation learning objective can then be formalized as minimizing the expected squared difference between the embedding space distance $\hat{d}(\phi_\theta(o_i), \phi_\theta(o_j))$ and a metric $d^\pi(o_i, o_j)$ defined over observations (Chen & Pan, 2022). Therefore, the loss can be formally written as follow:

$$L(\phi_\theta) = \mathbb{E}\left[\left(\hat{d}(\phi_\theta(o_i), \phi_\theta(o_j)) - d^\pi(o_i, o_j)\right)^2\right]. \tag{1}$$

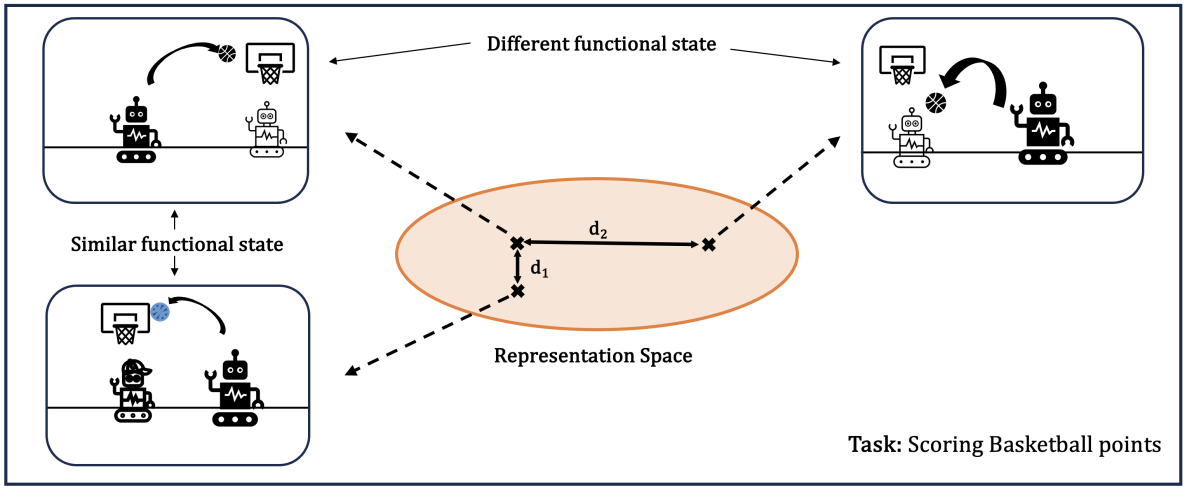

Fig. 6. Metric-based methods shape the representation space to capture task-relevant information. Representations with similar functional outcomes (e.g., shooting a basketball in the right trajectory) have minimal distance d_1, while representations with different outcomes (e.g., shooting a basketball in the wrong trajectory) are separated by larger distances d_2, hence $d_1 \ll d_2$.

Benefits: Metric-based methods offer strong theoretical guarantees by bounding the differences in value function outputs for pairs of embedded states, ensuring that states close in the metric space exhibit similar optimal behaviors. This is formalized as $|V^*(s_i) - V^*(s_j)| \leq d(s_i, s_j)$, which is key to improving sample efficiency and generalization, as it allows the agent to treat behaviorally equivalent states similarly. Additionally, these methods leverage task-relevant MDP information, such as rewards and transition dynamics, to shape the latent state space, making them particularly effective in abstracting away irrelevant visual distractions in more complex environments (Zhang et al., 2020). Also, some metric-based methods can avoid the need for training additional parameters, offering a computationally efficient approach (Castro et al., 2021).

Limitations: The operations involved in certain metrics, such as the Wasserstein distance used in bisimulation, are known to be computationally challenging (Castro, 2020a). This can lead to the need for approximations or relaxations, which can weaken the original theoretical guarantees (Chen & Pan, 2022). Furthermore, these methods typically require access to task-specific MDP information, which may not always be readily available or easy to obtain in real-world settings. In fact, even if rewards are available, real-world settings are often characterized by sparse reward structures, which can create latent instability or even embedding collapses in metric-based methods. Embedding explosion is another issue that can affect these methods (Kemertas & Aumentado-Armstrong, 2021). Finally, these methods are impacted by the non-stationary nature of the policy during training, which causes continuous updates to the embedding space and metrics, therefore favoring latent instabilities and hindering consistent performance compared to some other classes.

Categorization: Various metrics can be defined to quantify the similarity between states, each influencing how state representations are learned and aggregated.

a) Bisimulation Metrics

Bisimulation metrics, originally introduced for MDPs by Ferns et al. (2012), offer a way to quantify behavioral similarity between states. By measuring distances between states based on differences in both their rewards and transition dynamics, it allows state aggregation while preserving crucial information needed for effective policy learning. Formally, the bisimulation metric $d(x_i, x_j)$ between latent states x_i and x_j is updated using the following recursive rule:

$$T_k(d)(x_i, x_j) = \max_{a \in A} \left[(1 - c) \cdot |R(x_i, a) - R(x_j, a)| + c \cdot W_d(P(\cdot|x_i, a), P(\cdot|x_j, a)) \right]. \quad (2)$$

In this formulation, $T_k(d)$ represents an operator that updates the distance function $d(x_i, x_j)$, where $c \in [0, 1]$ is a parameter controlling the balance between the importance of reward differences and transition dynamics. The term $W_d(P(\cdot|x_i, a), P(\cdot|x_j, a))$ represents the Wasserstein distance (or Kantorovich distance) between the next-state distributions induced by the transitions from states x_i and x_j under action a.

Intuitively, W_d can be seen as quantifying the distance between two probability distributions, which corresponds in this case to the next-state distributions of (x_i, x_j). More precisely, the Wasserstein distance is known to measure the cost of transporting one probability distribution to another, and is formalized as finding an optimal coupling between two probability distributions that minimises a notion of transport cost associated with the base metric d (Villani, 2008). By iteratively applying the operator $T_k(d)$, the bisimulation distance $d(x_i, x_j)$ converges to a fixed point d^*, yielding the final metric between states that minimizes the loss. This iterative process that progressively shapes the representation space ensures that states with similar rewards and transition dynamics are mapped closer in the representation space, while dissimilar states are mapped further apart. Details on the convergence, the formalism of this operator and the space it operates on can be found in Castro et al. (2021).

Methods: Several methods integrate the bisimulation metric for learning more compact and generalizable representations in reinforcement learning. DBC (Zhang et al., 2020) uses the bisimulation metric to map behaviorally similar states closer in latent space, improving robustness to distractions, but is susceptible to embedding explosions/collapses, and relies on the assumption of Gaussian transitions for metric computation. Kemertas & Aumentado-Armstrong (2021) addresses these issues by (1) adding a norm constraint to prevent embedding explosion and (2) using intrinsic rewards plus latent space regularization through the learning of an IDM as an auxiliary task to prevent embedding collapse. The second point is particularly relevant in sparse or near-constant reward settings, where early similar trajectories can incorrectly lead the encoder to assume bisimilarity. More recent works addressing this sparse-reward challenge for bisimulation-based methods include Chen et al. (2024b) and Anonymous (2024). Castro et al. (2021) resolves some computational limitations of traditional bisimulation metrics with a scalable, sample-based approach that removes the need for assumptions like Gaussian or deterministic transitions Zhang et al. (2020) Castro (2020b), and explicitly learns state similarity without requiring additional network parameters.

b) Lax Bisimulation Metric

The lax bisimulation metric (Taylor et al., 2008) extends this concept to state-action equivalence by relaxing the requirement for exact action matching when comparing states, allowing both MDPs to have different action sets, thus providing greater flexibility. For example, Rezaei-Shoshtari et al. (2022) demonstrated the use of this metric for representation learning, which led to improved performance when learning from pixel observations. Le Lan et al. (2021)'s work also highlights why the lax bisimulation metric can provide continuity advantages over the original bisimulation metric.

c) Related Metrics

Several alternative metrics have been proposed to shape the representation space of RL agents. For instance, a temporal distance metric was used in Florensa et al. (2019) and Park et al. (2024b), which captures the minimum number of time steps required to transition between states in a goal-conditioned value-based setting. In Rudolph et al. (2024), their action-bisimulation metric replaces the reward-based similarity term of traditional bisimulation with a control-relevant term obtained by training an IDM model, making the approach reward-free. Agarwal et al. (2021a) introduced the Policy Similarity Metric (PSM), which replaces the absolute reward difference in bisimulation with a probability pseudometric between policies and has been shown to improve multi-task generalization.

d) Impact of distance $\hat{d}$ on Representations

The choice of how distances between representations are measured often influences the actual nature of the learned representations. For example, the L1 distance (1), based on absolute differences, promotes sparsity by applying a constant penalty that drives smaller values toward zero, emphasizing distinct features. This can be useful when only a few key features matter in distinguishing states. In contrast, the L2 distance (2), which uses squared differences, promotes smoother representations by spreading the error across all components, reducing large individual components while retaining contributions from smaller ones. This is more effective when information from all features is relevant, even if some contributions are minor. Some methods instead use orientation-based metrics, such as cosine similarity (4) or angular distance (3), which can be advantageous in high-dimensional spaces where direction is more significant than magnitude, or where specific properties, such as non-zero self-distances, are desirable (Castro et al., 2021). They can however come with drawbacks, such as potential embedding norm growth and convergence slowdowns when optimizing cosine similarity, limiting effectiveness without additional normalization (Draganov et al., 2024).

3.3 Auxiliary Tasks Methods

Definition: This category is composed of methods that enhance the primary learning task (RL) by having agents simultaneously predict additional environment-related outputs. This is done by splitting the representation part of an agent into n different heads, each with their own set of weights and dedicated task. During training, the errors from these heads are propagated back to the shared encoder ϕ, guiding the learning in conjunction with the main objective. The role of these supplementary tasks is to help agents enrich their representations with additional auxiliary signals coming from the same amount of data.

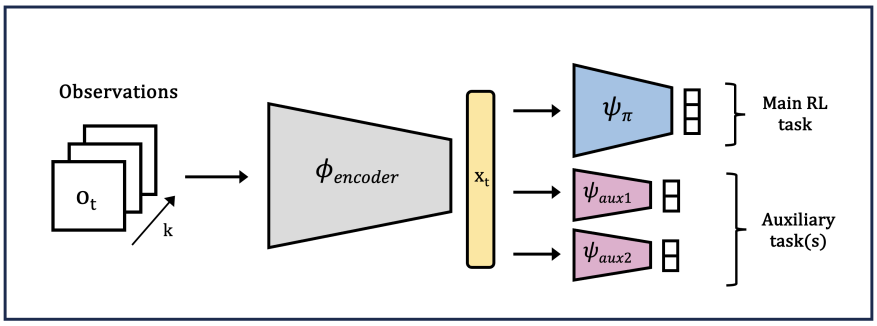

Fig. 7. The representation x_t of an RL agent is used to make additional predictions on auxiliary task function(s). These predictions are used to improve the representation itself.

Details: Let $\mathcal{L}_{\text{primary}}(\theta)$ denote the loss associated with the primary RL objective. Auxiliary tasks are defined as additional functions $\psi_{aux_i}(x_t)$ that, with their own set of parameters $\theta_{Aux} = \{\theta_{\text{aux}_1}, \theta_{\text{aux}_2}, ..., \theta_{\text{aux}_n}\}$, process the representation x_t to output a set of real values with task-dependent dimensions. The loss for each auxiliary task i is represented as $\mathcal{L}_i(\theta_{\text{aux}_i})$, and the overall auxiliary task loss is the sum of all task losses. The combined objective to optimize is then (3), with λ_i as the weighting factor(s) between primary and auxiliary tasks.

$$\mathcal{L}_{\text{method}}(\theta, \theta_{\text{Aux}}) = \mathcal{L}_{\text{primary}}(\theta) + \sum_{i=1}^{n} \lambda_i \mathcal{L}_i(\theta_{\text{aux}_i}) \tag{3}$$

Benefits: Auxiliary tasks for RL can enhance the learning process by utilizing additional supervised signals from the same experiences. When faced with environments with sparse rewards, auxiliary tasks can still provide some degree of learning signals for shaping the representation, which increases the learning efficiency of an agent. They can also serve as regularizers, enhancing generalization and reducing overfitting during learning. Finally, they can promote better exploration by guiding the agent toward states that provide more informative signals for the auxiliary tasks.

Limitations: However, a downside of using auxiliary tasks to improve representations is the lack of theoretical guarantee when it comes to whether it is actually benefiting the learning process of the main RL objective or not (Du et al., 2020). Defining precisely what makes a good auxiliary task is also in itself a hard problem (Lyle et al., 2021) (Rafiee et al., 2022). Finally, choosing the auxiliary weight(s) that balance(s) the importance of the auxiliary task(s) compared to the main task requires the right tuning of hyper-parameters.

Categorization: In the following sections, we explore the inner mechanisms of some auxiliary tasks that are frequently employed to learn good state representations in RL.

a) Reconstruction-based Methods

Definition: These methods aim to improve state representations by learning to reconstruct original observations o_t using a decoder $\hat{o}_t = \psi_{recon}(x_t)$ that takes as input the encoded representations $x_t = \phi(o_t)$. This reconstruction process can be performed using simple autoencoders (AE), where the objective is to minimize the reconstruction error between the original observation o_t and its predicted reconstruction $\hat{o}_t$. Additionally, it can be achieved using variational autoencoders (VAEs) (Kingma & Welling, 2022), where an additional regularization objective encourages the latent variables to follow a predefined distribution (commonly a Gaussian), promoting better generalization and disentanglement of the representations.

Purpose: These methods enforce the latent space to capture the essential features needed for accurate reconstruction, promoting the learning of compact, denoised representations. This helps the agent to focus on key task-relevant information, improving generalization across environments and enhancing sample efficiency in high-dimensional spaces. Mask-based latent reconstruction avoids the need to reconstruct full observations by focusing the reconstruction only on latent variables, thereby discarding irrelevant features from observational space.

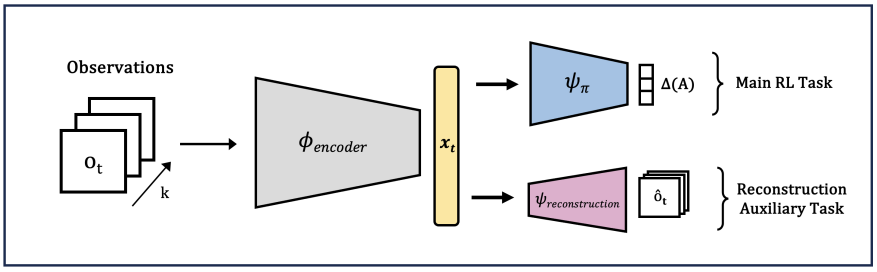

Fig. 8. Reconstruction as an auxiliary task: The encoder learns compact latent representations by ensuring that the original observation can be reconstructed from the representation.

Failure Cases: A common failure arises when reconstructing observations that contain significant irrelevant noise or distractions. In such cases, especially when task-relevant features occupy only a small portion of the observation, the model may learn to preserve unnecessary details, leading to poor state representations that degrade learning performance.

Disentangled Representations: Reconstruction-based methods can also be used to achieve disentangled representations, where the latent space $\mathcal{X}$ is ideally structured into independent subspaces $\mathcal{X}_i$, each capturing a distinct factor of variation v_i from the observation space. Methods like β-VAE (Higgins et al., 2016) enforce stronger constraints on the latent space than regular VAEs in order to promote disentanglement, ensuring that changes in one factor (e.g., object color) do not affect others (e.g., arm position), thus enhancing the robustness and adaptability of learned representations in complex environments. More related methods include (Higgins et al., 2017) (Thomas et al., 2018) (Kabra et al., 2021) (Dunion et al., 2023) (Dunion et al., 2024) (Dunion & Albrecht, 2024).

b) Dynamics Modeling Methods

Definition: Dynamics modeling methods use latent forward and inverse models as auxiliary tasks to implicitly improve state representation learning. A latent forward dynamic model (FDM) predicts the next representation $\hat{x}_{t+1} = f(x_t, a_t; \phi_{\text{fwd}})$ from the current representation x_t and action a_t, while a latent inverse dynamic model (IDM) predicts the action $\hat{a}_t = g(x_t, x_{t+1}; \phi_{\text{inv}})$ that caused a transition from x_t to x_{t+1}.

Purpose: FDMs help the agent learn a representation that captures environment dynamics, ensuring that the latent space encodes the essential transition information needed to predict future states. IDMs ensure that the representations encode information to recover the action that led to the state change, focusing on controllable aspects of the environment.

Failure Case: A failure case of using FDMs occurs when the transition model lacks a grounding objective, such as reward prediction (Tomar et al., 2021). In such cases, the model can collapse by mapping all observations to the same representation, minimizing the loss trivially, and failing to learn meaningful representations, especially if the critic's signal becomes noisy due to distractions.

k-step predictions: Using k-step predictions, where the model predicts multiple future representations instead of just one at each step, can further enhance the representation by capturing longer-term dependencies and improving performance across time (Schwarzer et al., 2020). For latent IDMs, predicting the first action a_t on a trajectory from o_t to o_{t+k} can also ensure positive properties for control Lamb et al. (2023) (Islam et al., 2023a).

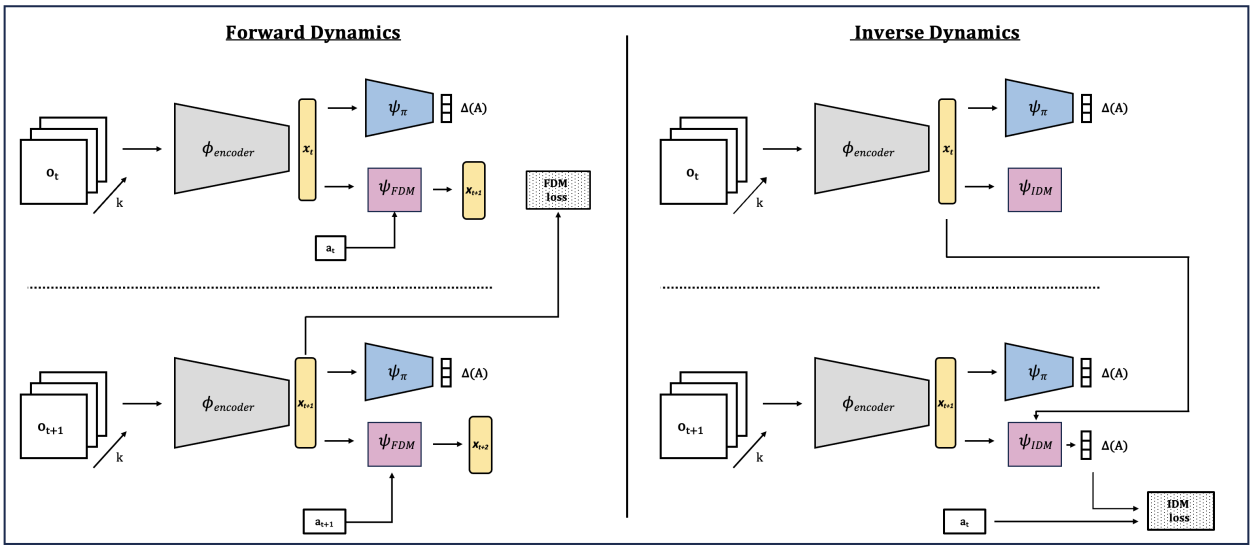

Fig. 9. Dynamics modeling as an auxiliary task: Forward Dynamics Models (FDMs) predict future representation(s) based on the current representation and action, capturing environment dynamics. Inverse Dynamics Models (IDMs) predict action that caused transitions between representations, emphasizing controllable features.

Hierarchical Models: McInroe et al. recently introduced a hierarchical approach utilizing multiple latent forward models (FDMs) to capture environment dynamics at varying temporal scales. Each level in the hierarchy learns a distinct FDM that predicts the representation x_{t+k} k-steps ahead based on previous representations and actions. Additionally, a learned communication module facilitates the sharing of higher-level information with lower-level modules. When compared on a suite of popular control tasks, it achieves noticeable performance and efficiency gains over baseline approaches. Importantly, this differs from predicting all k next representations x_{t+1} to x_{t+k}.

c) More Auxiliary Tasks

A wide variety of additional predictions can be used to support representation learning in RL. Here, we highlight a few additional examples. **(i)** Reward Prediction (Yang et al., 2022) (Zhou et al., 2023) involves predicting the immediate reward r_t based on the current state x_t and action a_t, guiding the agent to encode task-relevant features essential for value estimation. This task is especially useful in non-sparse reward settings, where it serves as a discriminator of critical information and benefits from being combined with latent modeling to capture relevant dynamics. **(ii)** Random General Value Functions (GVFs) (Zheng et al., 2021) predict random features of observations based on random actions, generating varied signals that enhance state representations, even when the main RL task is detached through a stop-gradient. **(iii)** Termination Prediction (Kartal et al., 2019) anticipates whether a state

will lead to the end of an episode, helping the agent recognize conditions for task completion and improving decisions around critical states. **(iv)** Multi-Horizon Value Prediction (Fedus et al., 2019) involves predicting value functions over multiple future horizons, allowing the agent to account for both short and long-term consequences, supporting more balanced and informed decision-making.

d) Precision – Auxiliary Tasks vs Auxiliary Losses

We define auxiliary tasks as something different than what is called auxiliary losses in the literature. Auxiliary losses refers to any loss optimized jointly with the main RL objective, which is something done in methods belonging to most classes here. However, we specifically define auxiliary tasks as additional predictions made during training, using the representation x_t as input, which indirectly enhance the quality of x_t. By definition, those tasks require additional parameters for each task-head, unlike auxiliary losses.

3.4 Data Augmentation Methods

Definition: Data augmentation (DA) methods represent a class of techniques that enhance sample-efficiency and generalization capabilities of RL agents through the manipulation of their observations. By applying geometric and photometric transformations to their inputs, such as rotations, translations, and color changes, these methods enforce invariance to irrelevant changes in observations, enabling agents to focus on essential features.

Details: These methods normally introduce an observation transformation function T that generates augmented observations $\tilde{o}$ based on the original observations o, where $\tilde{o} = T(o)$. The transformation T is chosen such that it preserves the essential task-relevant properties of o. A form of explicit and/or implicit regularization is then used to enforce some degree of Q-invariance and/or π-invariance. Formally, the invariance of a Q-function with respect to a transformation f_T is defined as $Q(s, a) = Q(f_T(s), a)$ for all $s \in S, a \in A$. Similarly, a policy π is considered invariant to f_T if $\pi(a \mid s) = \pi(a \mid f_T(s))$ for all $s \in S, a \in A$.

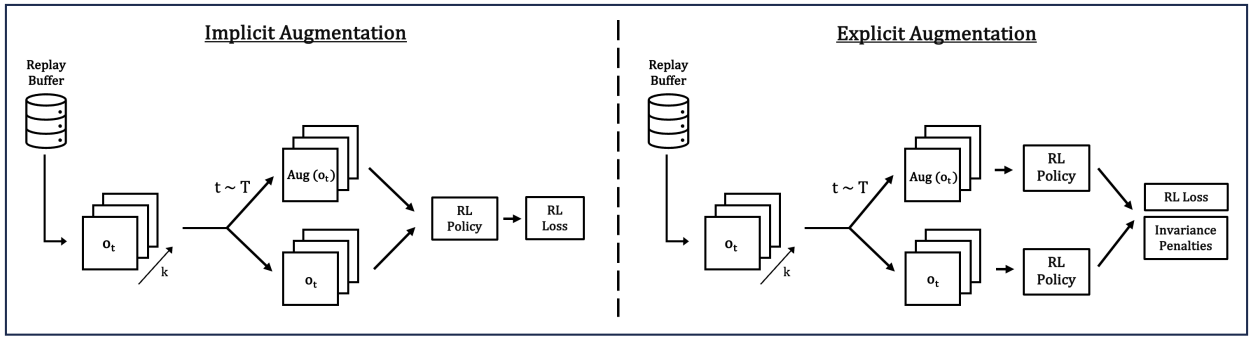

Fig. 10. Implicit D.A (left) augments observations directly used to train the policy and/or value network, promoting robustness through diversity without explicit constraints. Explicit D.A (right) augments observations supplemented by regularization penalties that enforce Q/π invariances.

Two type of strategies can be used to enforce invariance: **(1)** Implicit regularization applies transformations directly to the input data during the training process, using both original and transformed observations to train the network to generalize across these variations (Hu et al., 2024); **(2)** Explicit regularization, on the other hand, achieves invariance by modifying the loss functions to ensure that both the policy (actor) and the value estimates (critic) remain unchanged by the transformations f_T. This is done by incorporating additional terms in the loss functions that measure and penalize discrepancies between outputs, such as Q-values or action distributions, for both original and transformed inputs.

Benefits: DA-based methods enhance sample efficiency by diversifying training samples, enabling robust policy learning with fewer interactions and reducing overfitting. They improve generalization by simulating visual variations, reducing sensitivity to distribution shifts and aiding adaptation to new settings. Crucially, they preserve plasticity (Ma et al., 2024), essential for non-stationary objectives, while remaining simple and effective across environments.

Limitations: However, these kinds of approaches don't directly structure the representation space or incorporate task-specific information, making them fundamentally limited in capturing task-relevant features. Additionally, strong augmentations can introduce noise that disrupts training (e.g., high variance in Q-value estimates), and augmentations non-adapted to a task may affect the learning of critical features, reducing performance.

Data Augmentations: Four common augmentations applied to observations in DRL are: **(i)** Random Cropping, which modifies the image borders without altering central objects; **(ii)** Color Jittering, which adjusts brightness, contrast, and saturation to mimic varying lighting conditions; **(iii)** Random Rotation, involving slight image rotations that do not affect task orientation; and **(iv)** Noise Injection, where stochastic noise is added to images to simulate sensory disturbances or camera imperfections Ma et al. (2022). Certain augmentations, such as random cropping, have often demonstrated greater benefits compared to others (Laskin et al., 2020), although their effectiveness can be highly task-dependent.

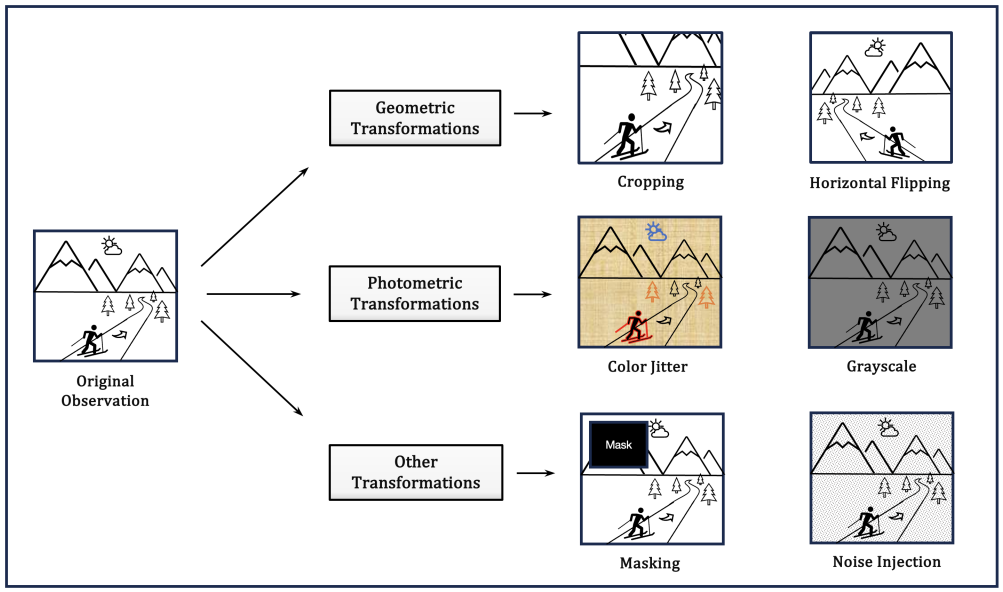

Fig. 11. Common observation augmentations in RL. Geometric transformations alter spatial properties like cropping or flipping, while photometric transformations alter visual features such as lighting and color. Other augmentations also exist, such as cropping or noise injection.

Methods: Data-regularized Q (DrQ) (Kostrikov et al., 2020) enhances data efficiency and robustness by integrating image transformations and averaging the Q target and function over multiple transformations, thereby reducing the variance in Q-function estimation. Building on DrQ, DrQ-v2 (Yarats et al., 2021) introduces improvements like switching from SAC to DDPG and incorporating n-step returns, along with more sophisticated image augmentation techniques such as bilinear interpolation to further enhance generalization and computational efficiency. RAD (Laskin et al., 2020), on the other hand, focuses on training with multiple views of the same input through simple augmentations, improving efficiency and generalization without altering the underlying algorithm.

DrAC (Raileanu et al., 2021) was introduced as an explicit regularization method that automatically determines suitable augmentations for any RL task and uses regularization terms for both the policy and value functions, enabling DA for actor-critic methods. (Hansen et al., 2021) proposed a data augmentation framework for off-policy RL called SVEA, which improves the stability of Q-value estimation. Addressing some limitations of SVEA, SADA (Almuzairee et al., 2024) enhances both stability and generalization by systematically augmenting both actor and critic inputs, allowing for a broader range of augmentations.

Some methods also relies on more unique techniques: (Li et al., 2024) propose normalization techniques for improved generalization, acting as latent data augmentations by altering feature maps instead of raw observations. To stabilize policy/Q-estimation outputs on augmented observations even further, Yuan et al. (2022a) proposed to identify task-relevant pixels with large Lipschitz constants (by measuring the effect of pixel perturbations on output decisions), and then to augment only the task-irrelevant pixels, which preserve critical information while benefiting from data diversity. Inspired by Fourier analysis in computer vision, Huang et al. (2022) introduced frequency domain augmentations, which provide a task-agnostic plug-and-play alternative to traditional spatial domain DA methods.

Precision: Although surveyed here, DA methods do not necessarily learn state representations via an encoder but can still be viewed as a class of representation learning techniques for improving efficiency and generalization in RL. Furthermore, the class focuses solely on observation augmentations; however, other forms of augmentations exist, such as transition or trajectory ones, which can also enhance various learning aspects (Ma et al., 2022) (Yu et al., 2021). For more details on data augmentations for reinforcement learning, the studies made by Hu et al. (2024) and Ma et al. (2022) are good resources to refer to.

3.5 Contrastive Learning Methods

Definition: Contrastive learning methods aim to learn effective representations for deep RL agents by contrasting positive pairs (similar data points) against negative pairs (dissimilar data points). This approach utilizes a contrastive loss function that encourages the model to increase the similarity of representations derived from positive pairs while simultaneously decreasing the similarity of representations from negative ones. These methods can leverage different strategies to define positive pairs, such as using data augmentations or exploiting the temporal structures in the data.

Details: The InfoNCE loss (van den Oord et al., 2018b) is a widely used contrastive loss for learning representations, both in vision-based SSL and RL specifically. It can be defined as:

$$\mathcal{L}_{\text{NCE}} = -\mathbb{E}_{(o,o^+,\{o_i^-\})}\left[\log \frac{\exp(\text{sim}(\phi_\theta(o), \phi_\theta(o^+)))}{\exp(\text{sim}(\phi_\theta(o), \phi_\theta(o^+))) + \sum_{i=1}^{N} \exp(\text{sim}(\phi_\theta(o), \phi_\theta(o_i^-)))}\right]. \quad (4)$$

In the objective above, o represents an observation (anchor), o^+ is a positive sample—typically a similar observation to o, generated through data augmentations like cropping, rotation, or jittering—and $\{o_i^-\}_{i=1}^N$ are negative samples, which are dissimilar observations selected randomly or based on temporal differences. The encoder ϕ_θ, as usual, maps observations into a representation space. The similarity between two representations, $\text{sim}(x, x')$, is often measured using cosine similarity or the dot product.

Intuitively, optimizing this loss encourages the model to make the numerator (similarity between o and o^+) as large as possible relative to the denominator (which sums the similarities between o and each negative sample). This pushes representations of positive pairs closer together and separates representations of negative pairs, ensuring that observations with similar underlying features cluster in the representation space, while dissimilar observations are spread apart. Consequently, this helps the reinforcement learning agent to better distinguish between important and irrelevant aspects of the environment.

Categorization: Contrastive methods can be categorized based on how they generate positive and negative pairs. Instance-discriminative contrastive learning typically leverages data augmentation, creating variations of the same observation to form positive pairs, while treating different observations in the batch as negatives. In contrast, temporal contrastive learning focuses on leveraging the sequential nature of the data, treating observations from nearby time steps as positive pairs, which captures temporal consistency in dynamic environments and distinguishes them from temporally distant observations as negatives.

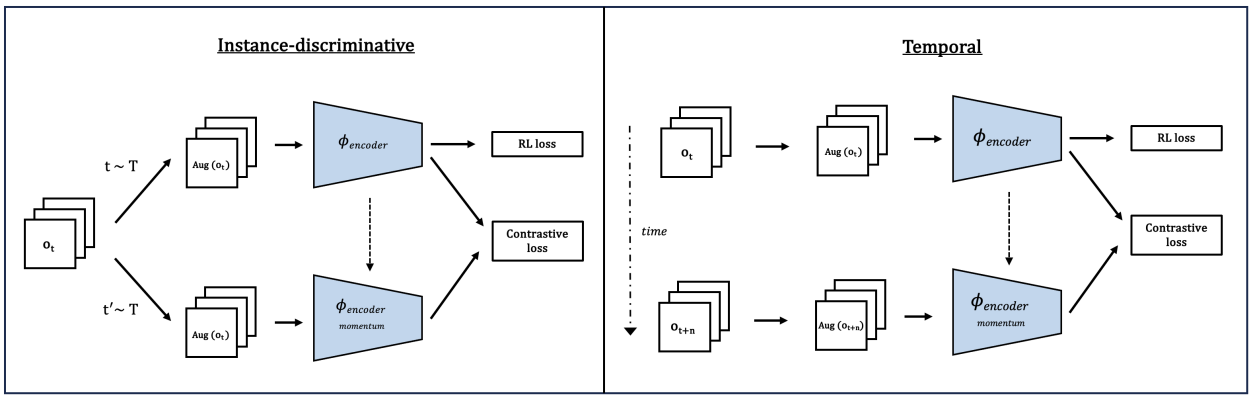

Fig. 12. Two contrastive learning frameworks: (1) Instance-discriminative contrastive learning with data augmentation (left), and (2) Temporal contrastive learning (right).

Benefits: Contrastive methods structure the representation space informatively, either by creating invariance to non-task-relevant variations in observations and/or by making the representation space temporally coherent and smooth. This invariance aspect is especially valuable in complex environments, where different observations might not alter the fundamental true state s_t, thus aiding in maintaining consistent decision-making processes.

Limitations: Scaling contrastive methods to high-dimensional spaces is challenging, as the number of contrastive samples required to learn meaningful representations may grow exponentially with the input space's dimension (LeCun, 2022). Additionally, finding appropriate negative pairs is crucial: if negatives are too easy, learning plateaus without gaining useful insights, while overly difficult negatives can hinder learning. Contrastive methods therefore require high batch sizes to avoid biased gradient estimates caused by limited negative samples within a batch (Chen et al., 2022). Finally, they typically do not leverage reward signals, which can restrict the structuring of the latent space when such information is available.

a) Instance-Discriminative Contrastive Learning

CURL (Srinivas et al., 2020) is a popular approach that make uses of a contrastive loss to ensure that representations of augmented versions of the same image are closer together than representations of different images, which enforces some beneficial invariance properties in the representation space that improve generalization and efficiency in visual RL tasks. To advance this direction further, future methods could integrate ideas similar to (Wang et al., 2024c), where their notion of data augmentation consistency ensures that stronger augmentations push an augmented sample's representation further from the original than weaker ones, structuring the representation space more informatively.

b) Temporal Contrastive Learning

Temporal contrastive methods like Contrastive Predictive Coding (CPC) (van den Oord et al., 2018a) use autoregressive models to predict future latent states by distinguishing between true and false future states, encouraging representations that capture essential predictive features. Building on CPC, Contrastive Difference Predictive Coding (CDPC) (Zheng et al., 2024a) introduces a temporal difference estimator for the InfoNCE loss used in CPC, improving sample efficiency and performance in stochastic environments. Augmented Temporal Contrast (ATC) (Stooke et al., 2020) aligns temporally close observations under augmentation, learning representations independently of policy updates, which has proven effective in complex RL settings.

Additional related approaches, such as ST-DIM (Anand et al., 2019) and DRIML (Mazoure et al., 2020), formulate their objectives based on mutual information maximization between global and local representations (Hjelm et al., 2018). Some methods combine contrastive learning with auxiliary tasks, such as Allen et al. (2021) who combines contrastive learning with an Inverse Dynamics Model (IDM) to learn Markov state abstractions. TACO (Zheng et al., 2024b) takes a different approach and learns both state and action representations by maximizing mutual information between: representations of current states combined with action sequences, and representations of the corresponding future states.

c) Similarity to Metric-based Approaches

Both metric-based and contrastive methods define similarity between state embeddings differently. Contrastive methods use a binary approach, treating pairs of observations as either positive (similar) or negative (dissimilar), aiming to minimize representation distances for positives and maximize them for negatives. Metric-based methods, however, quantify similarity more precisely with continuous distances, reflecting task-relevant criteria like expected rewards or transition dynamics. While contrastive methods often rely on image features or temporal proximity, metric-based approaches incorporate task-specific information, which can enable a richer representation space that captures varying degrees of task similarity.

3.6 Non-Contrastive Learning Methods

Definition: Non-contrastive methods aim to learn effective representations for RL by minimizing the distance between representations of similar observations, which are again generated using temporal proximity or data transformations. Unlike the contrastive approaches discussed in the previous section, these methods do not explicitly maximize the distance between dissimilar observations, relying instead solely on positive pairs during training.

Details: Methods in this call rely heavily on techniques to prevent total dimensional collapse—a failure mode where the representation space collapses to a single constant vector. This collapse occurs when embeddings are only drawn together using positive pairs, leading to a trivial solution where all embeddings converge to a constant vector, $\phi(o_t) = \mathrm{c}$, which minimizes error but retains no meaningful information. To mitigate this, non-contrastive approaches mostly employ two type of strategies:

(i) Regularization techniques, which modify the loss function to maintain diversity in the embedding space, for instance by encouraging the covariance matrix of a batch of embeddings to approximate an identity matrix (Bardes et al., 2021); **(ii)** Architectural techniques that introduce asymmetry—such as predictors, momentum encoders, and stop-gradients —which help regulate update paths during training, thereby preventing collapse (Grill et al., 2020).

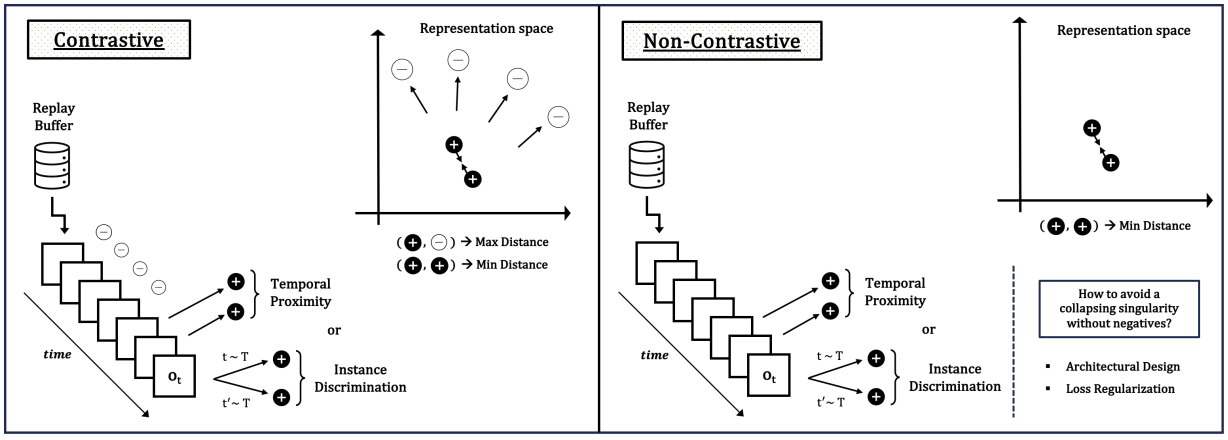

Fig. 13. Distinction between contrastive and non-contrastive approaches. Contrastive methods rely on both positive and negative pairs to structure the representation space, maximizing similarity within positive pairs and minimizing it for negatives. Non-contrastive methods, which avoid the use of negative pairs, address the challenge of representation collapse through architectural designs or loss regularization. In both approaches, positive samples are generated either through instance discrimination using data augmentations of the same o_t or via temporal proximity.

Benefits: Avoiding the need for negative pairs greatly simplifies the learning process, making these methods more computationally efficient and stable in high-dimensional spaces. Additionally, by focusing solely on positive pairs, non-contrastive methods are better suited for scaling to complex observation spaces, avoiding the pitfalls of hard-to-balance negative sampling that can limit contrastive approaches. These approaches are also intuitively aligned with biological representation learning, where positive associations are reinforced without explicitly contrasting them with negative examples.

Limitations: Non-contrastive methods are susceptible to informational collapse (also known as dimensional collapse), where the embedding vectors fail to span the full representation space, resulting in a lower-dimensional subspace that limits the information encoded. This issue, affecting both contrastive and non-contrastive methods, leads to redundancy in the representation, as embedding components can become highly correlated rather than decorrelated, reducing the diversity and effectiveness of the learned features. Additionally, these methods often lack task-specific information, such as rewards, which could guide the formation of more meaningful state representations.

Categorization: Methods in this class can be classified based on whether they make use of a latent predictive component in their architecture or not. In the self-supervised learning terminology, we refer to the former as Joint Embedding Architectures (JEA) and the latter as Joint Embedding Predictive Architectures (JEPA) (Assran et al., 2023). Non-predictive methods aim to make representations invariant to transformations without using a predictor between representation backbones. In contrast, predictive methods incorporate a non-constant predictor, making the representations self-predictive by learning a latent dynamics model during training, which is discarded afterward.

a) Non-Predictive Methods

BarlowRL (Cagatan & Akgun, 2023) can be seen as an example of a non-contrastive method used for RL that does not use a predictive component. Based on the Barlow Twins framework (Zbontar et al., 2021) and the Data-Efficient Rainbow algorithm (DER) (Hessel et al., 2018), this regularization-based method trains an encoder to map closely together embeddings of an observation o_t and its data-augmented version o'_t. Tested on the Atari 100k benchmark, it showed better results than CURL (Srinivas et al., 2020), a contrastive instance discrimination method presented in the last section. However, it didn't outperform SPR (Schwarzer et al., 2020), a non-contrastive predictive method presented in the next section.

b) Self-Predictive Methods

Self-predictive methods can be further categorized by whether the predictor relies only on the representation x_t or is also conditioned on transformation parameters between observations, such as the action a_t (Garrido et al., 2024). Without conditioning, methods like BYOL (Grill et al., 2020) and SimSiam (Chen & He, 2021) learn transformation-invariant representations. Conditioning on a_t, however, enables the predictor to encode the effect of actions on representations, capturing dynamics in the environment. Among different approaches, temporal self-predictive methods effectively leverage the temporal structure of reinforcement learning environments. These methods focus on predicting future latent representations by using temporally close observations as positive pairs, encouraging the encoder to capture compressed and predictive information about future states. Specifically, the state encoder $\phi(o_t)$ is jointly learned with a latent transition model $P(x_t, a_t)$, which can be extended to predict multiple steps into the future. Adding data augmentations on processed observations can further enhance robustness and enable richer representations.

Methods: SPR (Schwarzer et al., 2020), inspired by BYOL (Grill et al., 2020), uses a transition model and data augmentations to predict an agent's latent state representations several steps into the future, achieving strong sample efficiency in pixel-based RL and outperforming expert human scores on several Atari games. PBL (Guo et al., 2020), designed for multi-task generalization, predicts future latent embeddings that recursively predict agent states, creating a bootstrap effect to enhance environment dynamics learning. Both SPR and PBL operate in the latent space, allowing for multimodal inputs and richer representations. Ni et al. (2024) conducted a comprehensive analysis of self-predictive learning in MDPs and POMDPs, introducing a minimalist self-predictive approach validated across various control settings, including standard, distracting, and sparse-reward environments. Tang et al. (2022) also explored self-predictive learning in RL, highlighting its ability to learn meaningful latent representations by avoiding collapse through careful optimization dynamics. Building on their insights, they introduced bidirectional self-predictive learning, using forward and backward predictions to improve representation robustness. Some additional relevant works in self-predictive RL include (Zhang et al., 2024b), (Fujimoto et al., 2024), (Khetarpal et al., 2024), (Voelcker et al., 2024) and (Yu et al., 2022).

3.7 Attention-based Methods

Definition: Attention-based methods in reinforcement learning involve mechanisms that enable agents to focus on relevant parts and features of their complex observations while ignoring less important information. This selective focus allows an agent to process inputs more efficiently, leading to lower convergence time, better performance, and better interpretability of an agent's decision making.

Details: Attention mechanisms in visual RL agents are typically implemented using mask-based or patch-based attention. Mask-based attention learns weights to highlight relevant regions in observations, while patch-based attention divides inputs into patches and learns relevance scores to focus on the most significant ones. Different types of attention—such as regular or self-attention, soft or hard attention, temporal or spatial attention, single-head or multi-head attention, and top-down or bottom-up attention—can be integrated within an RL agent's architecture. For CNN-based encoders, agents extract feature maps $h_1, h_2, \ldots$ through convolutional layers $c_1, c_2, \ldots$, starting from the observation o_t, and map these to the representation x_t using fully connected layers. Self-attention modules can be applied at different stages, targeting high-level or low-level features, or spanning across layers. Alternatively, attention can act as a bottleneck mask directly on the input o_t.

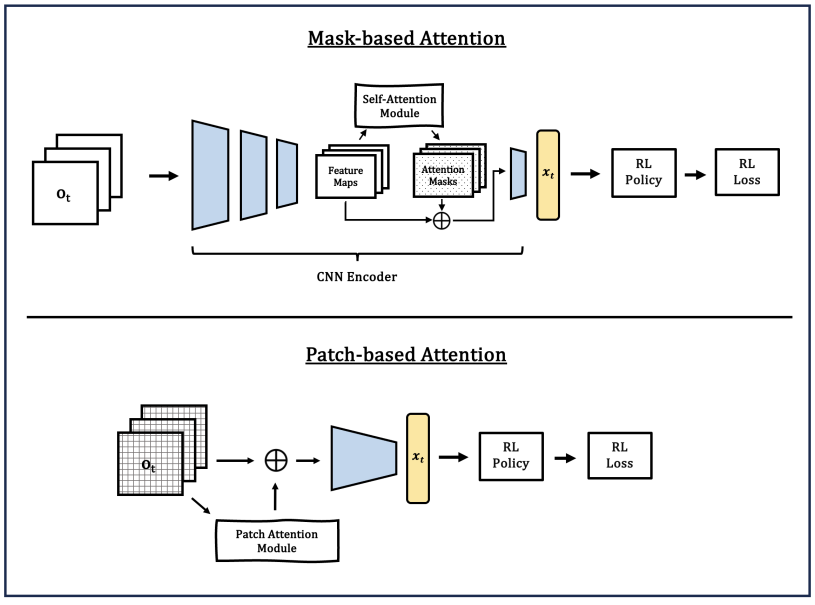

Fig. 14. Top: A self-attention module operates on high-level feature maps extracted from observations, creating attention masks that reweight these feature maps through element-wise multiplication. Bottom: An attention bottleneck is applied directly to observations, where an attention module selectively focuses on patches of the input image.

The computational steps inside a self-attention module that operates on feature maps, similarly to Figure 14 (top), can be understood as follows. Starting with a feature map H, the module projects H into query, key, and value matrices: $Q = W_q H$, $K = W_k H$, and $V = W_v H$, where W_q, W_k, and W_v are learned transformations. Attention weights are calculated as $A = \text{softmax}(QK^\top/\sqrt{d})$, capturing the relevance of regions within H. The resulting attention-weighted representation $Y = AV$ aggregates information from V according to these relevance scores, focusing on parts of the input that enhance the relevance of x_t.

Benefits/Limitations: These methods enhance efficiency by focusing on task-critical regions both temporally and spatially, allowing agents to process relevant observations more effectively and reduce the complexity of the state space. They also improve interpretability, providing insights into the agent's focus areas with the usage saliency maps, making the decision-making process easier to understand. On the other hand, those mechanisms can increase computational complexity due to the additional parameters required. They may also exhibit poor generalization in new settings or with heavy distractions (Tang et al., 2020).

Methods: Tang et al. (2020) introduced a self-attention bottleneck that learns to select the top K image patches for efficient processing of relevant visual information. Wu et al. (2021) proposed an attention module that uses a self-supervised approach to generate attention masks, enhancing CNN-based RL performance on Atari games. Chen et al. (2019) integrated temporal and spatial attention into a hierarchical DRL framework for improved lane changing in autonomous driving, achieving smoother and safer behaviors. Chen et al. (2024a) propose Focus-Then-Decide (FTD), a method that combines two auxiliary tasks with the RL loss to train an attention mechanism. This mechanism identifies task-relevant objects from those returned by a foundational segmentation model, leveraging prior computations to achieve strong performance in complex, visually noisy scenarios. Bertoin et al. (2022) propose Saliency-Guided Q-networks (SGQN) for improved generalization, a framework that generate saliency maps highlighting the most relevant parts of an image for decision-making. The training procedure is supported by two additional objectives: a self-supervised feedback loop where the agent learns to predict its own saliency maps and a regularization term that ensures the value function depends more on the identified important regions. Sodhani et al. (2021b) introduced an attention-based multi-task learning method that uses a mixture of k encoders, with task context from a pre-trained language model determining soft-attention weights for combining encoder outputs, improving task performance. Mott et al. (2019) proposed a soft, top-down attention mechanism that enhances interpretability and performance by generating task-focused attention maps, enabling better generalization and adaptability to unseen game configurations, surpassing bottom-up approaches.

3.8 Alternative Approaches

In this section, we provide a brief overview of some additional classes of methods for state representation learning in DRL. While their principles may be less widely used than other presented classes, they still offer unique approaches and insights on learning representations.

a) Spectral-based Methods

Definition: Spectral-based methods in state representation learning employ the eigenvalues and eigenvectors of matrices derived from transition dynamics to capture structural and geometric information about the environment. These methods create embeddings that preserve the connectivity and global topology of the state space, enhancing the representation x_t of the observation o_t.

Details: In spectral-based methods, observations o_t can be represented as nodes in a graph $G = (O, W)$, where W is a matrix that reflects transition probabilities between observations. The Laplacian matrix $L = D - W$, where D is a matrix capturing how connected each observation is, provides spectral properties used to create embeddings. By using the smallest eigenvectors of L, each observation o_t is mapped to a vector $x_t = [e_1(o_t), e_2(o_t), \ldots, e_d(o_t)]$ that captures both local and global relationships in the environment. Laplacian representations were initially formulated as a pretraining objective, learned for a uniformly random policy and fixed throughout training to avoid complexity. This static approach however fails to adapt to policies during RL, as policy updates during training can necessitate recomputation of representations. Recent methods (Anonymous, 2025) address this limitation by enabling online Laplacian representation learning.

Methods: Gomez et al. (2024) introduced a framework to approximate accurately Laplacian eigenvectors and eigenvalues effectively, while addressing challenges such as hyperparameter sensitivity and scalability, ensuring accurate and robust representations for RL. Wang et al. (2023) improves traditional Laplacian representations by making sure the Euclidean representation distance between two observations also reflects a measure of reachability between them in the environment, allowing better reward shaping and bottleneck state discovery in goal-reaching tasks. Finally, Wu et al. (2019) propose a scalable, sample-efficient approach to compute Laplacian eigenvectors in model-free RL, enabling practical applications in high-dimensional or continuous environments.

b) Information Bottleneck Approaches

Definition: The Information-Bottleneck (IB) principle (Tishby et al., 2000) provides a framework for learning compact and task-relevant representations by optimizing the trade-off between compression and relevance. When used for SRL, IB aims to learn a state encoder that minimizes the mutual information $I(O;X)$ between observations O and representations X to compress irrelevant information while maximizing $I(X;Y)$, where Y corresponds to task-relevant targets such as rewards or actions.

Details: The IB objective balances compression and relevance of state representations by minimizing $I(O;X) - \beta I(X;Y)$, where o_t and x_t denote observations and representations, respectively. Regular IB requires estimating mutual information terms, which is computationally intractable for high-dimensional inputs. Variational Information Bottleneck (VIB) (Alemi et al., 2017) addresses this by introducing parametric approximations with an encoder $q_\phi(X|O)$ and decoder $p_\psi(Y|X)$. The VIB objective therefore combines task relevance and compactness, making it scalable for deep RL.

Methods: REPDIB (Islam et al., 2023b) leverages the IB principle by incorporating discrete latent representations to enforce a structured and compact representation space. It maximizes the relevance of task-specific information while filtering out exogenous noise, leading to improved exploration and performances in continuous control tasks. DRIBO (Fan & Li, 2022) employs a multi-view framework to filter out irrelevant information via a contrastive Multi-View IB (MIB) objective, enhancing robustness to visual distractions. IBAC (Igl et al., 2019) integrates IB into an actor-critic framework, promoting compressed representations and better feature extraction in low-data regimes. Additional methods include IBORM (Jin et al., 2021), which leverage IB in a multi-agent setting, and MIB (You & Liu, 2024), which introduced a multimodal information bottleneck approach for learning task-relevant joint representations from egocentric images and proprioception.

4. Benchmarking & Evaluation

Evaluating correctly state representation learning methods requires appropriate benchmarks and tools to assess the quality of the learned representations. The choice of a benchmark depends on the specific nature of the task to solve, such as whether the environment involves continuous or discrete observation/action spaces. Key properties like reward density, task horizon, and the presence of distractions also play a crucial role in selecting the right environment. In fact, comparisons between SRL methods should primarily focus on these environment-specific properties, as suggested by Tomar et al. (2021). Rather than claiming that approach A is universally better than approach B on a given benchmark, it is better to state that approach A is better suited for distraction-based learning than approach B, based on the experiments conducted.

In the following sections, common evaluation aspects of SRL methods are reviewed, followed by methods for assessing the quality of the learned state representations.

4.1 Common Evaluation Aspects

Accurately evaluating state representation learning methods requires assessing their effectiveness in supporting key objectives both during and after reinforcement learning training. These methods are typically compared based on the following aspects:

Performance: In deep reinforcement learning, better performance is defined by achieving a higher expected cumulative reward. An improved representation ϕ should enable a policy π_ϕ that maximizes reward, such that $J(\pi_\phi) \geq J(\pi_{\text{baseline}})$. This criterion applies whether the representations are pre-trained via SRL and then kept fixed during RL, or when performance is evaluated concurrently as the representations are refined.

Sample Efficiency: Better sample efficiency can be quantified by the number of samples N required to achieve a specific performance level. Let $N(\epsilon)$ be the number of samples needed to achieve a performance within ϵ of the optimal performance J^*. An improved representation ϕ enhances sample efficiency if $N_\phi(\epsilon) < N_{\text{baseline}}(\epsilon)$, meaning fewer samples are needed with the improved representation to achieve the same performance.

Generalization: The generalization ability of a representation ϕ quantifies its capacity to support policy performance in previously unseen environments. A representation generalizes well if the policy π_ϕ achieves a consistent expected return across different environments, measured by the condition $|J(\pi_\phi; \text{train}) - J(\pi_\phi; \text{test})| \leq \delta$, where δ is a small tolerance threshold. Additionally, fine-tuning with minimal environment interactions to achieve prior high performance further indicates the effectiveness of a representation in this context. Generalization may also be assessed by metrics such as transfer success rate or zero-shot adaptation score.

Robustness: The robustness of a state representation ϕ can be assessed by its stability across variations in underlying RL algorithms, hyperparameters, and training conditions within the SRL method. Formally, given a set of configurations C (e.g., different RL algorithms, hyperparameter settings, or noise levels) and a tolerance δ, the representation is considered robust if $\max_{c \in C} |J(\pi_\phi; c) - J(\pi_\phi; c^*)| \leq \delta$, where c^* represents an optimal configuration. A robust representation exhibits minimal performance variance across C, indicating reduced dependency on specific settings and greater applicability across various RL scenarios.

4.2 Assessing the Quality of Representations

Evaluating the quality of learned state representations in reinforcement learning (RL) is essential for understanding how well representations capture task-relevant information. Therefore, having good metrics for quantifying the quality of those learned state representations is crucial. Various methods can be utilized for this purpose, and we categorized those based on whether they require access to ground truth states s_t or not. However, assuming access to true underlying states isn't always realistic and may limit practical applicability.

a) Evaluation Without True States

Total Return: The most common approach for evaluating the quality of learned state representations is simply to let an RL agent use the learned states to perform the desired task and assess the final return obtained with specific representations methods. This verifies that the necessary information to solve the task is embedded in the representation x_t. However, this process is often time-intensive and computationally demanding, necessitating substantial data and multiple random seeds to account for the high variance in performance (Agarwal et al., 2021b). The performance can also vary depending on the base agent, adding further complexity to this evaluation approach.

Visual Inspection: Another technique that does not require access to ground truth states involves extracting the observations of the nearest neighbors in the learned state space and visually examining if those observations match approximately to the close neighbors in the ground truth state space (Sermanet et al., 2018). In other words, close points in the latent state space should correspond to observations with similar task-relevant information.

Latent Information: More general metrics for assessing the quality of representations can be based on measuring properties of good SSL representations, such as the variance of individual dimensions across a batch, the covariance between representation dimensions, the average entropy of representation vectors, or spectral properties of a representation matrix, such as its effective rank or condition number (Garrido et al., 2023). Disentanglement could be measured by perturbing randomly small parts of an input observation and measuring the impact on the dimensions of representation x_t, as disentangled representations are expected to limit the effect of random small perturbations to only a few dimensions, reflecting independent and meaningful feature encoding.

Latent Continuity: Metrics can also be designed to evaluate the continuity and smoothness of Q-functions, action predictions, or simply temporal coherence in the representation space (Le Lan et al., 2021). By examining multiple local areas in the representation space, along with the nearest neighbors and their corresponding Q-values, action distributions, or time-steps, such metrics can assess whether nearby points yield similar values or actions. Ensuring this continuity helps maintain stable decision-making and simplifies the function approximation of the policy $\psi_\pi(x_t)$ and/or value network $\psi_v(x_t)$, enhancing efficiency.

Linear Probing: Zhang et al. (2024a) proposed to use 2 probing tasks for assessing the quality of learned representations: (i) reward prediction in a given state, and (ii) action prediction taken by an expert in a given state. The authors used linear probing specifically, where a linear layer is trained on top of frozen representations for each prediction task, constraining the probe's performance to rely heavily on the quality of x_t. Overall, their probing tasks were shown to strongly correlate with downstream control performance, offering an efficient method for assessing the quality of state representations. Probing on frozen representations x_t can also be used to reconstruct observations, which evaluates how well the representations retain information about the original observations. However, this may be less effective when observations contain significant noise or distractions.

Interpretability: Understanding the key information that RL agents focus on within observations and encoded representations can provide better insights into their decision-making. A general scheme for determining the attention levels at different parts of an observation consist of perturbing random areas of the input, then measuring the resulting policy or value changes (Greydanus et al., 2018) (Yuan et al., 2022a). Regions causing higher variance under similar perturbations are likely more relevant for the agent. This perturbation-based approach can also be extended to individual representation dimensions to evaluate their importance by analyzing the induced changes in policy or value outputs. When dealing with stacked observations, gradient-based techniques such as (Weitkamp et al., 2019) provide a practical alternative through action-specific activation maps.

b) Evaluation with True States

Probing: Evaluating the quality of learned state representations can be done by training a linear classifier on top of frozen representations to predict ground-truth state variables (Jonschkowski et al., 2017), and reporting metrics such as the mean F1 score. This linear probing approach was applied by Anand et al. (2019) to evaluate the performance of a representation learning technique in Atari. The underlying assumption is that successful regression indicates that meaningful features are well-encoded in the learned state, and good generalization performance on the test set suggests a robust encoding of these features. This concept can also be extended to non-linear probes (Tupper & Neshatian, 2020).

Geometry: The KNN-MSE (K-Nearest Neighbors Mean Squared Error) metric from Lesort et al. (2017b) can evaluate learned state representations by first identifying the k-nearest neighbors of each image I in the learned state space. It then calculates the mean squared error between the ground truth states of the original image I and its nearest neighbors I' to assess the preservation of local structure in the learned representations. Manifold learning metrics such as NIEQA (Zhang et al., 2012) can also evaluate how well the learned representations preserve the original state's local and global geometry (Lesort et al., 2017a).

Disentanglement: The disentanglement metric proposed in Higgins et al. (2016) can assess how well a learned representation separates factors of variation by fixing one factor in pairs of data samples, calculating the average differences in their latent representations, and using a linear classifier to predict the fixed factor. The classifier's accuracy indicates the quality of disentanglement, with higher accuracy reflecting better separation of independent factors.

5. Looking Beyond

While current state representation learning (SRL) methods for deep reinforcement learning (DRL) have made good progress in improving sample efficiency, generalization, and performance, there is still room for improvement. As environments become more complex and varied, it's important to explore more ways of enhancing those techniques for more challenging settings. This section looks at several directions, each extending the learning of state representations to broader domains.

Direction	Description
Multi-Task	Explore the sharing of representations across multiple tasks to capture common structures.
Offline Pre-Training	Leverage datasets of past interactions for pre-training state representations, boosting efficiency and transfer.
Pre-trained Vision	Integrate representations from pre-trained visual models into agents for efficiency and generalization gains.
Zero-Shot RL	Produce representations that enable agents to perform new tasks without additional training.
Leveraging Priors	Utilize large language models (LLMs/VLMs) to incorporate prior knowledge into representations.
Multi-Modal	Methods that integrate information from multiple sensory modalities for getting richer representations.

Table 3. Promising directions for enhancing state representation learning in DRL.

5.1 Multi-Task Representation Learning

Definition: Multi-task representation learning (MTRL) involves training an RL agent to extract a shared low-dimensional representation among a set of related tasks and use either one or separate heads attached to this common representation to solve each task. This approach leverages the similarities and shared features among tasks to improve overall learning efficiency and performance. Although various settings of MTRL exist, they often share the common points presented here.

Benefits: MTRL reduces sample complexity by leveraging shared structures between tasks, which facilitates faster convergence, enhances generalization and robustness on new tasks, and enables effective knowledge transfer, where learning one task boosts performance on related ones (Cheng et al., 2022) (Efroni et al., 2022).

Challenges: Negative transfer when shared representations are suboptimal for certain tasks represents an important issue, which can lead to interference and degraded performance (Sodhani et al., 2021a). Balancing task contributions to the shared encoder is also challenging when tasks vary in difficulty or nature. Additionally, differences in data distributions among tasks can limit the effectiveness of representations, with benefits often relying on some assumptions (Lu et al., 2022).

Methods: To mitigate negative interference, CARE (Sodhani et al., 2021a) proposes to encode observations into multiple representations using a mixture of encoders, allowing the agent to dynamically attend to relevant representations based on context. Efroni et al. (2022) introduced a framework for efficient representational transfer in reinforcement learning, showcasing sample complexity gains. Kalashnikov et al. (2022) presented MT-Opt, a scalable multi-task robotic learning system leveraging shared experiences and representations. PBL (Guo et al., 2020) trains representations by predicting latent embeddings of future observations, which are simultaneously trained to predict the original representations, enabling strong performances in multitask and partially observable RL settings. Hessel et al. (2019) introduced PopArt, a framework that automatically adjusts the contribution of each task to the learning dynamics of multi-task RL agents, hence becoming invariant to different reward densities and magnitude across tasks. Ishfaq et al. (2024) proposed MORL, an offline multitask representation learning algorithm that enhances sample efficiency in downstream tasks. Cheng et al. (2022) introduced REFUEL, a representation learning algorithm for multitask RL under low-rank MDPs, with proven sample complexity benefits.

5.2 Offline Pre-Training of Representations

Definition: The offline pre-training of state representations refers to the learning of state representations from static datasets of trajectories $\{(o_{i,t}, a_{i,t}, r_{i,t}) \mid i = 1, \ldots, N;\ t = 1, \ldots, T\}$ or demonstrations $\{(o_{i,t}, a_{i,t}) \mid i = 1, \ldots, N;\ t = 1, \ldots, T\}$ in order to accelerate learning on downstream RL tasks. This strategy is motivated by the necessity to enhance data efficiency on downstream tasks and overcome the limitations of learning tabula rasa, which often leads to some degree of overfitting. Akin to human decision-making, this aims to leverage prior knowledge contained in some already collected interactions.

Benefits: By leveraging large amounts of pre-collected data, offline pre-training of representations can enhance data efficiency by reducing the need for extensive online interactions to achieve high performance. This pretraining process can lead to better initializations for RL algorithms, resulting in faster convergence and superior final performance on downstream tasks. Moreover, representations learned from diverse offline datasets can enhance the robustness of RL agents, allowing them to generalize better across environments and tasks.

Challenges: The quality and diversity of the offline dataset are crucial, as poorly curated or biased datasets can result in suboptimal representations. Ensuring that the learned representations are transferable and useful for a wide range of downstream tasks is also complex, as certain features may not generalize well beyond the pretraining context. Additionally, the pretraining of large models on extensive datasets demands substantial computational resources, making the process both time-consuming and expensive.

Methods: The study performed by Yang & Nachum (2021) demonstrate that offline experience datasets can successfully be used to learn state representations of observations such that learning policies from these pre-trained representations improves performance on a downstream task. Through their investigation, they demonstrate performance gains across 3 downstream applications: online RL, imitation learning, and offline RL. Their results provide good insights on different representation learning objectives, and also suggests that the optimal objective depends on the downstream task's nature and is not absolute. Kim et al. (2024) also investigated the efficacy of various pre-training objectives on trajectory and observation datasets, but focused specifically on evaluating the generalization capabilities of visual RL agents compared to a broader range of pre-training approaches. Farebrother et al. (2023) introduced Proto-Value Networks (PVNs), a method that scales representation learning by using auxiliary predictions based on the successor measure to capture the structure of the environment, producing rich state features that enable competitive performance with fewer interactions. Schwarzer et al. (2021) introduced SGI, a self-supervised method for

representation learning that combines the latent dynamics modeling from SPR Schwarzer et al. (2020), the unsupervised goal-conditioned RL from HER (Andrychowicz et al., 2017), and inverse dynamics modeling for capturing environment dynamics. It achieves strong performance on the Atari 100k benchmark with reduced data, and good scaling properties.

5.3 Pre-trained Visual Representations

Definition: Pre-trained visual representations (PVRs), also called visual foundation models for control, involve utilizing unlabeled pre-training data from images and/or videos to learn representations that can be used for downstream reinforcement learning tasks. These representations are trained to learn the spatial characteristics of observations $\{o_i \mid i = 1, \ldots, N\}$ and the temporal dynamics from videos $\{o_{i,t} \mid i = 1, \ldots, N;\ t = 1, \ldots, T\}$, and can be seen as initializing an RL agent with some initial vision capabilities before learning a task. PVRs can be pre-trained either on domain-similar data or on general data with transferable features.

Benefits: PVRs benefit from abundant and inexpensive image and video data compared to action-reward-labeled trajectory data, enabling scalable learning across domains. They improve sample efficiency by providing pre-learned visual features, reducing the need for task-specific relearning. PVRs also enhance generalization by transferring robust visual features across diverse tasks and environments, even under variations or unseen conditions.

Challenges: However, challenges include the lack of temporal data leverage for Image-based PVRs, while video-based PVRs face challenges like exogenous noise (e.g., background movements) that degrade performance. Without action labels, distinguishing relevant states from noise becomes significantly harder, and sample complexity for video data can grow exponentially Misra et al. (2024). Additionally, distribution shifts between pre-training and target tasks further complicate video representation learning Zhao et al. (2022). Finally, while PVRs benefit model-free RL, Schneider et al. (2024) found they fail to enhance sample efficiency or generalization in model-based RL, especially for out-of-distribution cases.

Methods: Yuan et al. (2022b) demonstrated that frozen ImageNet ResNet representations combined with DrQ-v2 (Yarats et al., 2021) as a base algorithm can significantly improve generalization in challenging settings, though fine-tuning degraded performance. MVP (Xiao et al., 2022) showed that pre-training on diverse image and video data using masked image modeling (He et al., 2022) while keeping the weights of the visual encoder frozen preserves the quality of representations and accelerates RL training on downstream motor tasks. Majumdar et al. (2023) studied PVRs across tasks, finding: (1) no universal PVR method dominate despite overall better performance than learning from scratch; (2) scaling model

size and data diversity improves average performance but not consistently across tasks; (3) adapting PVRs to downstream tasks provides the best results and is more effective than training representations from scratch. The study made by Kim et al. (2024) of different pre-training objectives suggest that image and video-based PVRs improve generalization across different target tasks, while reward-specific pre-training benefits similar domains but performs poorly in different target environments.

Misra et al. (2024) analyzed different approaches for video-based representation learning and found that forward modeling and temporal contrastive learning objectives can efficiently capture latent states under independent and identically distributed (iid) noise, but struggle with exogenous noise, which increases sample complexity. Their empirical results confirm strong performance in noise-free settings but a degradation under noise. Other relevant work includes VIP (Ma et al., 2023) and R3M (Nair et al., 2022), though the latter was initially limited to behavior cloning. Finally, approaches that learn latent representations while recovering latent-action information solely from video dynamics (Schmidt & Jiang, 2024) represent an interesting avenue for video-based PVRs trained on large action-free dataset.

5.4 Representations for Zero-Shot RL

Definition: Zero-shot RL aims to enable agents to perform optimally on any reward function provided at test time without additional training, planning, or fine-tuning. The objective is to train agents that can understand and execute any task description immediately by utilizing a compact environment representation derived from reward-free transitions (s_t, a_t, s_{t+1}). When a reward function is specified, the agent should use the learned representations to generate an effective policy with minimal computation. More precisely, given a reward-free MDP (S, A, P, γ), the goal is to obtain a learnable representation E such that, once a reward function $r : S \times A \to \mathbb{R}$ is provided, we can compute without planning, from E and r, a policy π whose performance is close to optimal.

Benefits/Challenges: Zero-shot RL offers flexibility by enabling agents to adapt to various tasks without retraining, enhancing efficiency and scalability across numerous tasks without additional training or planning. However, challenges include developing comprehensive representations without reward information, ensuring transferability across complex tasks, consistently achieving near-optimal performances, assuming access to good exploration policies during pre-training, and the complexity of the algorithms.

Methods: Forward-Backward (FB) representations, introduced by Touati & Ollivier (2021), enable zero-shot RL by learning two functions: a forward function to capture future transitions and a backward function to encode paths to states, trained in an unsupervised manner on state transitions without rewards. The intuition for these representations can be seen as aligning the future of a state with the past of another by maximizing $F(s)^\top B(s')$ for states s and s' that are closely connected through the environment's dynamics. This approach offers a simpler alternative to world models, enabling efficient computation of near-optimal policies for any reward function without additional training or planning, though it relies on an effective exploration strategy. In a related study by Touati et al. (2023), FB representations have shown to deliver superior performances across a wider range of settings compared to methods based on successor features (SFs), which also aim to do zero-shot RL based on successor representations (SRs) (Dayan, 1993). Recently, Jeen et al. (2024) proposed a Value-Conservative version of FB representations, addressing the performance degradation issue of previous methods when trained on small, low-diversity datasets. Other recent works in that direction include Proto Successor Measure Agarwal et al. (2024), Hilbert Foundation Policies (Park et al., 2024a), and Function Encoder (Ingebrand et al., 2024).

5.5 Leveraging VLMs/LLMs Prior Knowledge

Definition: Leveraging Vision-Language Models (VLMs) and Large Language Models (LLMs) for state representation learning involves using large pre-trained models to transform visual observations into natural language descriptions. These descriptions serve as interpretable and semantically rich state representations, which can then be used to produce task-relevant text embeddings, allowing reinforcement learning (RL) agents to learn policies from embeddings rather than pixels.

Benefits/Challenges: Leveraging VLMs/LLMs for this purpose can improve generalization by creating invariant representations that are less affected by disturbances in observations. Indeed, these models can successfully extract the presence of objects and filter out irrelevant details based on task-specific knowledge, focusing only on what is relevant. This has the advantage of leveraging the vast prior knowledge embedded in those large models, and can also enhance interpretability by allowing for a more transparent understanding of the agent's decision-making. However, many challenges still exist for a successful integration of VLMs/LLMs within an RL framework, starting by the higher computational resources necessary to leverage the capabilities of those models.

Methods: An interesting work in this direction was done by Rahman & Xue (2024) where they proposed to use a VLM to generate an image description, refined by a LLM to remove irrelevant information, and used for producing a state embedding given to the reinforcement learning agent. Their method showed generalization improvements compared to an end-to-end RL baseline. Other related works include (Chen et al., 2024c) (Wang et al., 2024a).

5.6 Multi-Modal Representation Learning

Definition: Multi-Modal State Representation Learning (MMSRL) integrates multiple data types to create richer and more comprehensive state representations for RL agents. By combining diverse information sources with different properties, MMSRL can enhance an agent's understanding of the environment, improving decision-making and generalization. For example, a robot navigating a room with a camera and a microphone will be able to learn unified representations combining sight and sound with MMSRL. Hence if it hears glass shatter but doesn't see it, the robot will be able to infer danger in another room.

Benefits/Challenges: MMSRL creates richer representations by capturing more comprehensive environmental features, making it resilient to noise or occlusion in individual modalities. This robustness improves the agent's ability to generalize and adapt across different tasks, resulting in better performance in varied and complex environments.

Methods: The work done by Becker et al. (2024) introduces a framework that enables the selection of the most suitable loss for each modality, such as using a reconstruction losses for low-dimensional proprioception data and a contrastive one for images with distractions.

5.7 Other Directions

Several directions fall outside the scope of this work but still deserve consideration: (**i**) Exploration strategies for enhanced state representation learning, rather than for rewards directly, will be essential for future open-ended applications, as they can ensure a relevant state space coverage and mitigate the risk of learning good representations for only a small part of the state space. In fact, the interplay between effective exploration and high-quality state representations is particularly important since effective exploration relies on a solid understanding of previously encountered states. (**ii**) State representation learning in continual learning settings, where representations are learned from continually evolving environments, aligns more closely with the dynamic nature of real-world problems and should be investigated further. (**iii**) Evaluating the scalability of SRL approaches remains an open challenge, with future methods needing to scale with increasing environment complexity, number of tasks, and aspects such as computational resources, data availability, and model parameters.

6. Conclusion

To summarize, this survey provides a comprehensive overview of techniques used for representation learning in deep reinforcement learning, focusing on strategies to enhance learning efficiency, performance, and generalization across high-dimensional observation spaces. By categorizing methods into distinct classes, we have highlighted each approach's mechanisms, strengths, and limitations, clarifying the landscape of SRL methods and serving as a practical guide for selecting suitable techniques. We also explored evaluation strategies for assessing the quality of learned representations, especially as techniques are applied to increasingly challenging settings. Robust evaluation remains essential for real-world applications, supporting reliable decision-making and generalization.

Looking forward, SRL methods must adapt to a broader set of settings, such as those outlined in Section 5. For each direction in that section, we reviewed related work that could serve as a foundation for further exploration, emphasizing the importance of continued research in these areas. Ultimately, advancing SRL will be crucial for developing robust, generalizable, and efficient DRL systems capable of tackling complex real-world tasks. We hope this survey serves as a resource for researchers and practitioners aiming to deepen their understanding of SRL techniques and offers a strong foundation for learning state representations in reinforcement learning.

Limitations: This survey primarily examines state representation learning methods within the model-free online RL setting, without addressing model-based approaches and offline RL evaluation. The comparisons between classes are largely theoretical or rely on previous studies. Future work could include experimental evaluations to compare approaches on multiple aspects. Lastly, while the taxonomy provides an overview of the main classes for learning state representations in RL, it does not explore each class in detail, as each could be the focus of its own survey. Some isolated approaches may also be missing due to our focus on categorizing mostly the recent developments.

References

David Abel. A theory of abstraction in reinforcement learning, 2022.

Rishabh Agarwal, Marlos C. Machado, Pablo Samuel Castro, and Marc G. Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. *CoRR*, abs/2101.05265, 2021a. URL https://arxiv.org/abs/2101.05265.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29314–29327, 2021b.

Siddhant Agarwal, Harshit Sikchi, Peter Stone, and Amy Zhang. Proto successor measure: Representing the space of all possible solutions of reinforcement learning, 2024. URL https://arxiv.org/abs/2411.19418.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

Cameron Allen, Neev Parikh, Omer Gottesman, and George Konidaris. Learning markov state abstractions for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34:8229–8241, 2021.

Abdulaziz Almuzairee, Nicklas Hansen, and Henrik I Christensen. A recipe for unbounded data augmentation in visual reinforcement learning. *Reinforcement Learning Journal*, 1: 130–157, 2024.

Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R. Devon Hjelm. Unsupervised state representation learning in atari. *CoRR*, abs/1906.08226, 2019. URL http://arxiv.org/abs/1906.08226.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Anonymous. Weak bisimulation metric-based representations for sparse-reward reinforcement learning. In *Submitted to The Thirteenth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=x7Q0uFTH2a. under

review.

Anonymous. Online laplacian-based representation learning in reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025. Under review.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Philipp Becker, Sebastian Mossburger, Fabian Otto, and Gerhard Neumann. Combining reconstruction and contrastive methods for multimodal representations in rl, 2024. URL https://arxiv.org/abs/2302.05342.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR*, abs/1206.5538, 2012. URL http://arxiv.org/abs/1206.5538.

David Bertoin, Adil Zouitine, Mehdi Zouitine, and Emmanuel Rachelson. Look where you look! saliency-guided q-networks for generalization in visual reinforcement learning. *Advances in Neural Information Processing Systems*, 35:30693–30706, 2022.

Wendelin Böhmer, Jost Tobias Springenberg, Joschka Boedecker, Martin A. Riedmiller, and Klaus Obermayer. Autonomous learning of state representations for control: An emerging field aims to autonomously learn state representations for reinforcement learning agents from their real-world sensor observations. *KI - Künstliche Intelligenz*, 29:353–362, 2015. URL https://api.semanticscholar.org/CorpusID:15176564.

Nicolò Botteghi, Mannes Poel, and Christoph Brune. Unsupervised representation learning in deep reinforcement learning: A review, 2024. URL https://arxiv.org/abs/2208.14226.

Omer Veysel Cagatan and Baris Akgun. Barlowrl: Barlow twins for data-efficient reinforcement learning, 2023.

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, 2020a.

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, 2020b.

Pablo Samuel Castro, Tyler Kastner, Prakash Panangaden, and Mark Rowland. Mico: Learning improved representations via sampling-based state similarity for markov decision processes. *CoRR*, abs/2106.08229, 2021. URL https://arxiv.org/abs/2106.08229.

Changyou Chen, Jianyi Zhang, Yi Xu, Liqun Chen, Jiali Duan, Yiran Chen, Son Tran, Belinda Zeng, and Trishul Chilimbi. Why do we need large batchsizes in contrastive learning? a gradient-bias perspective. *Advances in Neural Information Processing Systems*, 35:33860–33875, 2022.

Chao Chen, Jiacheng Xu, Weijian Liao, Hao Ding, Zongzhang Zhang, Yang Yu, and Rui Zhao. Focus-then-decide: Segmentation-assisted reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11240–11248, 2024a.

Jianda Chen and Sinno Pan. Learning representations via a robust behavioral metric for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 36654–36666, 2022.

Jianda Chen, Wen zheng terence Ng, Zichen Chen, Sinno Jialin Pan, and Tianwei Zhang. State chrono representation for enhancing generalization in reinforcement learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=J42SwBemEA.

William Chen, Oier Mees, Aviral Kumar, and Sergey Levine. Vision-language models provide promptable representations for reinforcement learning, 2024c. URL https://arxiv.org/abs/2402.02651.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.

Yilun Chen, Chiyu Dong, Praveen Palanisamy, Priyantha W. Mudalige, Katharina Muelling, and John M. Dolan. Attention-based hierarchical deep reinforcement learning for lane change behaviors in autonomous driving. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3697–3703, 2019. URL https://api.semanticscholar.org/CorpusID:198119613.

Yuan Cheng, Songtao Feng, Jing Yang, Hong Zhang, and Yingbin Liang. Provable benefit of multitask representation learning in reinforcement learning. *Advances in Neural*

Information Processing Systems, 35:31741–31754, 2022.

Peter Dayan. Improving generalisation for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.

Tim de Bruin, Jens Kober, Karl Tuyls, and Robert Babuška. Integrating state representation learning into deep reinforcement learning. *IEEE Robotics and Automation Letters*, 3(3): 1394–1401, 2018. doi: 10.1109/LRA.2018.2800101.

Andrew Draganov, Sharvaree Vadgama, and Erik J Bekkers. The hidden pitfalls of the cosine similarity loss. *arXiv preprint arXiv:2406.16468*, 2024.

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, pp. 1665–1674. PMLR, 2019.

Yunshu Du, Wojciech M. Czarnecki, Siddhant M. Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity, 2020. URL `https://arxiv.org/abs/1812.02224`.

Mhairi Dunion and Stefano V Albrecht. Multi-view disentanglement for reinforcement learning with multiple cameras. *Reinforcement Learning Journal*, 2:498–515, 2024.

Mhairi Dunion, Trevor McInroe, Kevin Sebastian Luck, Josiah P. Hanna, and Stefano V Albrecht. Temporal disentanglement of representations for improved generalisation in reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=sPgP6aISLTD`.

Mhairi Dunion, Trevor McInroe, Kevin Sebastian Luck, Josiah Hanna, and Stefano Albrecht. Conditional mutual information for disentangled representations in reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Yonathan Efroni, Sham Kakade, Tengyang Xie Ma, and Lin F Yang. Provable benefits of representational transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 29998–30010, 2022.

Jiameng Fan and Wenchao Li. Dribo: Robust deep reinforcement learning via multi-view information bottleneck. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*. PMLR, 2022.

Jesse Farebrother, Joshua Greaves, Rishabh Agarwal, Charline Le Lan, Ross Goroshin, Pablo Samuel Castro, and Marc G. Bellemare. Proto-value networks: Scaling representation learning with auxiliary tasks, 2023. URL `https://arxiv.org/abs/2304.12567`.

William Fedus, Carles Gelada, Yoshua Bengio, Marc G Bellemare, and Hugo Larochelle. Hyperbolic discounting and learning over multiple horizons. *arXiv preprint arXiv:1902.06865*, 2019.

Norman Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. *CoRR*, abs/1207.4114, 2012. URL `http://arxiv.org/abs/1207.4114`.

Carlos Florensa, Jonas Degrave, Nicolas Heess, Jost Tobias Springenberg, and Martin A. Riedmiller. Self-supervised learning of image embedding for continuous control. *CoRR*, abs/1901.00943, 2019. URL `http://arxiv.org/abs/1901.00943`.

Scott Fujimoto, Wei-Di Chang, Edward Smith, Shixiang Shane Gu, Doina Precup, and David Meger. For sale: State-action representation learning for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International conference on machine learning*, pp. 10929–10974. PMLR, 2023.

Quentin Garrido, Mahmoud Assran, Nicolas Ballas, Adrien Bardes, Laurent Najman, and Yann LeCun. Learning and leveraging world models in visual representation learning. *arXiv preprint arXiv:2403.00504*, 2024.

Diego Gomez, Michael Bowling, and Marlos C. Machado. Proper laplacian representation learning. In *International Conference on Learning Representations (ICLR)*, 2024.

Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. In *International conference on machine learning*, pp. 1792–1801. PMLR, 2018.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Zhaohan Daniel Guo, Bernardo Ávila Pires, Bilal Piot, Jean-Bastien Grill, Florent Altché, Rémi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. *CoRR*, abs/2004.14646, 2020. URL `https://arxiv.org/abs/2004.14646`.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *Advances in neural information processing systems*, 34:3680–3693, 2021.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on*

artificial intelligence, volume 32, 2018.

Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado Van Hasselt. Multi-task deep reinforcement learning with popart. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3796–3803, 2019.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2016. URL `https://api.semanticscholar.org/CorpusID:46798026`.

Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. In *International Conference on Machine Learning*, pp. 1480–1490. PMLR, 2017.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Jianshu Hu, Yunpeng Jiang, and Paul Weng. Revisiting data augmentation in deep reinforcement learning, 2024.

Yangru Huang, Peixi Peng, Yifan Zhao, Guangyao Chen, and Yonghong Tian. Spectrum random masking for generalization in image-based reinforcement learning. *Advances in Neural Information Processing Systems*, 35:20393–20406, 2022.

Maximilian Igl, Kamil Ciosek, Yingzhen Li, Sebastian Tschiatschek, Cheng Zhang, Sam Devlin, and Katja Hofmann. Generalization in reinforcement learning with selective noise injection and information bottleneck. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Tyler Ingebrand, Amy Zhang, and Ufuk Topcu. Zero-shot reinforcement learning via function encoders. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21007–21019. PMLR, 21–27 Jul 2024. URL `https://proceedings.mlr.press/v235/ingebrand24a.html`.

Haque Ishfaq, Thanh Nguyen-Tang, Songtao Feng, Raman Arora, Mengdi Wang, Ming Yin, and Doina Precup. Offline multitask representation learning for reinforcement learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=72tRD2Mfjd`.

Riashat Islam, Manan Tomar, Alex Lamb, Yonathan Efroni, Hongyu Zang, Aniket Didolkar, Dipendra Misra, Xin Li, Harm Van Seijen, Remi Tachet Des Combes, and John Langford.

Principled offline rl in the presence of rich exogenous information. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023a.

Riashat Islam, Hongyu Zang, Manan Tomar, Aniket Didolkar, et al. Representation learning in deep rl via discrete information bottleneck. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2023b.

Scott Jeen, Tom Bewley, and Jonathan Cullen. Zero-shot reinforcement learning from low quality data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=79eWvkLjib.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pp. 1704–1713. PMLR, 2017.

Yue Jin, Shuangqing Wei, Jian Yuan, and Xudong Zhang. Information-bottleneck-based behavior representation learning for multi-agent reinforcement learning. *arXiv preprint arXiv:2109.14188*, 2021.

Rico Jonschkowski, Roland Hafner, Jonathan Scholz, and Martin A. Riedmiller. Pves: Position-velocity encoders for unsupervised learning of structured state representations. *CoRR*, abs/1705.09805, 2017. URL http://arxiv.org/abs/1705.09805.

Rishabh Kabra, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matthew Botvinick, Alexander Lerchner, and Christopher P. Burgess. SIMONe: View-invariant, temporally-abstracted object representations via unsupervised video decomposition. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=YSzTMntO1KY.

Dmitry Kalashnikov, Jake Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Scaling up multi-task robotic reinforcement learning. In *Conference on Robot Learning*, pp. 557–575. PMLR, 2022.

Bilal Kartal, Pablo Hernandez-Leal, and Matthew E Taylor. Terminal prediction as an auxiliary task for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 15, pp. 38–44, 2019.

Mete Kemertas and Tristan Aumentado-Armstrong. Towards robust bisimulation metric learning. *CoRR*, abs/2110.14096, 2021. URL https://arxiv.org/abs/2110.14096.

Khimya Khetarpal, Zhaohan Daniel Guo, Bernardo Avila Pires, Yunhao Tang, Clare Lyle, Mark Rowland, Nicolas Heess, Diana Borsa, Arthur Guez, and Will Dabney. A unifying framework for action-conditional self-predictive reinforcement learning, 2024. URL https://arxiv.org/abs/2406.02035.

Donghu Kim, Hojoon Lee, Kyungmin Lee, Dongyoon Hwang, and Jaegul Choo. Investigating pre-training objectives for generalization in vision-based reinforcement learning, 2024. URL https://arxiv.org/abs/2406.06037.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *CoRR*, abs/2004.13649, 2020. URL https://arxiv.org/abs/2004.13649.

Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Pac reinforcement learning with rich observations. *Advances in Neural Information Processing Systems*, 29, 2016.

Alex Lamb, Riashat Islam, Yonathan Efroni, Aniket Rajiv Didolkar, Dipendra Misra, Dylan J Foster, Lekan P Molu, Rajan Chari, Akshay Krishnamurthy, and John Langford. Guaranteed discovery of control-endogenous latent states with multi-step inverse models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=TNocbXm5MZ.

Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *CoRR*, abs/2004.14990, 2020. URL https://arxiv.org/abs/2004.14990.

Charline Le Lan, Marc G Bellemare, and Pablo Samuel Castro. Metrics and continuity in reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8261–8269, 2021.

Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

Timothée Lesort, Mathieu Seurin, Xinrui Li, Natalia Díaz-Rodríguez, and David Filliat. Unsupervised state representation learning with robotic priors: a robustness benchmark. *arXiv preprint arXiv:1709.05185*, 2017a.

Timothée Lesort, Mathieu Seurin, Xinrui Li, Natalia Díaz Rodríguez, and David Filliat. Unsupervised state representation learning with robotic priors: a robustness benchmark. *CoRR*, abs/1709.05185, 2017b. URL http://arxiv.org/abs/1709.05185.

Timothée Lesort, Natalia Díaz Rodríguez, Jean-François Goudou, and David Filliat. State representation learning for control: An overview. *CoRR*, abs/1802.04181, 2018. URL http://arxiv.org/abs/1802.04181.

Lu Li, Jiafei Lyu, Guozheng Ma, Zilin Wang, Zhenjie Yang, Xiu Li, and Zhiheng Li. Normalization enhances generalization in visual reinforcement learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, pp. 1137–1146, Richland, SC, 2024. International Foundation for Autonomous Agents

and Multiagent Systems. ISBN 9798400704864.

Rui Lu, Andrew Zhao, Simon S Du, and Gao Huang. Provable general function class representation learning in multitask bandits and mdp. *Advances in Neural Information Processing Systems*, 35:11507–11519, 2022.

Clare Lyle, Mark Rowland, Georg Ostrovski, and Will Dabney. On the effect of auxiliary tasks on representation dynamics. *CoRR*, abs/2102.13089, 2021. URL https://arxiv.org/abs/2102.13089.

Guozheng Ma, Zhen Wang, Zhecheng Yuan, Xueqian Wang, Bo Yuan, and Dacheng Tao. A comprehensive survey of data augmentation in visual reinforcement learning, 2022. URL https://arxiv.org/abs/2210.04561.

Guozheng Ma, Lu Li, Sen Zhang, Zixuan Liu, Zhen Wang, Yixin Chen, Li Shen, Xueqian Wang, and Dacheng Tao. Revisiting plasticity in visual reinforcement learning: Data, modules and training stages. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=0aR1s9YxoL.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training, 2023. URL https://arxiv.org/abs/2210.00030.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *Advances in Neural Information Processing Systems*, 36:655–677, 2023.

Bogdan Mazoure, Remi Tachet des Combes, Thang Long Doan, Philip Bachman, and R Devon Hjelm. Deep reinforcement and infomax learning. *Advances in Neural Information Processing Systems*, 33:3686–3698, 2020.

Trevor McInroe, Lukas Schäfer, and Stefano V Albrecht. Multi-horizon representations with hierarchical forward models for reinforcement learning. *Transactions on Machine Learning Research*.

Dipendra Misra, Akanksha Saran, Tengyang Xie, Alex Lamb, and John Langford. Towards principled representation learning from videos for reinforcement learning, 2024. URL https://arxiv.org/abs/2403.13765.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. URL http://arxiv.org/abs/1312.5602.

Alex Mott, Daniel Zoran, Mike Chrzanowski, Daan Wierstra, and Danilo J. Rezende. Towards interpretable reinforcement learning using attention augmented agents. *CoRR*, abs/1906.02500, 2019. URL http://arxiv.org/abs/1906.02500.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation, 2022.

Tianwei Ni, Benjamin Eysenbach, Erfan Seyedsalehi, Michel Ma, Clement Gehring, Aditya Mahajan, and Pierre-Luc Bacon. Bridging state and history representations: Understanding self-predictive rl, 2024.

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with Hilbert representations. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39737–39761. PMLR, 21–27 Jul 2024a. URL https://proceedings.mlr.press/v235/park24g.html.

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with hilbert representations, 2024b. URL https://arxiv.org/abs/2402.15567.

Banafsheh Rafiee, Jun Jin, Jun Luo, and Adam White. What makes useful auxiliary tasks in reinforcement learning: investigating the effect of the target policy, 2022. URL https://arxiv.org/abs/2204.00565.

Md Masudur Rahman and Yexiang Xue. Natural language-based state representation in deep reinforcement learning. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 1310–1319, 2024.

Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in reinforcement learning. In *Neural Information Processing Systems*, 2021. URL https://api.semanticscholar.org/CorpusID:221094237.

Sahand Rezaei-Shoshtari, Rosie Zhao, Prakash Panangaden, David Meger, and Doina Precup. Continuous MDP homomorphisms and homomorphic policy gradient. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=Adl-fs-8OzL.

Max Rudolph, Caleb Chuck, Kevin Black, Misha Lvovsky, Scott Niekum, and Amy Zhang. Learning action-based representations using invariance, 2024. URL https://arxiv.org/abs/2403.16369.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=rvUq3cxpDF.

Moritz Schneider, Robert Krug, Narunas Vaskevicius, Luigi Palmieri, and Joschka Boedecker. The surprising ineffectiveness of pre-trained visual representations for model-based reinforcement learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=LvAy07mCxU.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.

Max Schwarzer, Nitarshan Rajkumar, Michael Noukhovitch, Ankesh Anand, Laurent Charlin, R. Devon Hjelm, Philip Bachman, and Aaron C. Courville. Pretraining representations for data-efficient reinforcement learning. *CoRR*, abs/2106.04799, 2021. URL https://arxiv.org/abs/2106.04799.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1134–1141. IEEE, 2018.

Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-task reinforcement learning with context-based representations. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 9767–9779. PMLR, 2021a.

Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-task reinforcement learning with context-based representations. *CoRR*, abs/2102.06177, 2021b. URL https://arxiv.org/abs/2102.06177.

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. CURL: contrastive unsupervised representations for reinforcement learning. *CoRR*, abs/2004.04136, 2020. URL https://arxiv.org/abs/2004.04136.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. *CoRR*, abs/2009.08319, 2020. URL https://arxiv.org/abs/2009.08319.

Yujin Tang, Duong Nguyen, and David Ha. Neuroevolution of self-interpretable agents. *CoRR*, abs/2003.08165, 2020. URL https://arxiv.org/abs/2003.08165.

Yunhao Tang, Zhaohan Daniel Guo, Pierre Harvey Richemond, Bernardo Ávila Pires, Yash Chandak, Rémi Munos, Mark Rowland, Mohammad Gheshlaghi Azar, Charline Le Lan, Clare Lyle, András György, Shantanu Thakoor, Will Dabney, Bilal Piot, Daniele Calandriello, and Michal Valko. Understanding self-predictive learning for reinforcement learning, 2022.

Jonathan Taylor, Doina Precup, and Prakash Panangaden. Bounding performance loss in approximate mdp homomorphisms. pp. 1649–1656, 01 2008.

Valentin Thomas, Emmanuel Bengio, William Fedus, Jules Pondard, Philippe Beaudoin, Hugo Larochelle, Joelle Pineau, Doina Precup, and Yoshua Bengio. Disentangling the independently controllable factors of variation by interacting with the world. *arXiv preprint arXiv:1802.09484*, 2018.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint arXiv:physics/0004057*, 2000.

Manan Tomar, Utkarsh A. Mishra, Amy Zhang, and Matthew E. Taylor. Learning representations for pixel-based control: What matters and why? *CoRR*, abs/2111.07775, 2021. URL `https://arxiv.org/abs/2111.07775`.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. *CoRR*, abs/2103.07945, 2021. URL `https://arxiv.org/abs/2103.07945`.

Ahmed Touati, Jérémy Rapin, and Yann Ollivier. Does zero-shot reinforcement learning exist?, 2023. URL `https://arxiv.org/abs/2209.14935`.

Adam Tupper and Kourosh Neshatian. Evaluating learned state representations for atari. In *2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pp. 1–6, 2020. doi: 10.1109/IVCNZ51579.2020.9290609.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018a. URL `http://arxiv.org/abs/1807.03748`.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018b. URL `http://arxiv.org/abs/1807.03748`.

Cédric Villani. Optimal transport: Old and new. 2008. URL `https://api.semanticscholar.org/CorpusID:118347220`.

Claas A Voelcker, Tyler Kastner, Igor Gilitschenski, and Amir-massoud Farahmand. When does self-prediction help? understanding auxiliary tasks in reinforcement learning. *Reinforcement Learning Journal*, 4:1567–1597, 2024.

Boyuan Wang, Yun Qu, Yuhang Jiang, Jianzhun Shao, Chang Liu, Wenming Yang, and Xiangyang Ji. LLM-empowered state representation for reinforcement learning. In *Forty-first International Conference on Machine Learning*, 2024a. URL `https://openreview.net/forum?id=xJMZbdiQnf`.

Han Wang, Erfan Miahi, Martha White, Marlos C Machado, Zaheer Abbas, Raksha Kumaraswamy, Vincent Liu, and Adam White. Investigating the properties of neural network representations in reinforcement learning. *Artificial Intelligence*, pp. 104100, 2024b.

Kaixin Wang, Kuangqi Zhou, Jiashi Feng, Bryan Hooi, and Xinchao Wang. Reachability-aware laplacian representation in reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*. PMLR, 2023.

Zihu Wang, Yu Wang, Zhuotong Chen, Hanbin Hu, and Peng Li. Contrastive learning with consistent representations. *Transactions on Machine Learning Research*, 2024c. ISSN 2835-8856. URL https://openreview.net/forum?id=gKeSI8w63Z.

Laurens Weitkamp, Elise van der Pol, and Zeynep Akata. Visual rationalizations in deep reinforcement learning for atari games. In *Artificial Intelligence: 30th Benelux Conference, BNAIC 2018,'s-Hertogenbosch, The Netherlands, November 8–9, 2018, Revised Selected Papers 30*, pp. 151–165. Springer, 2019.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, may 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Haiping Wu, Khimya Khetarpal, and Doina Precup. Self-supervised attention-aware reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2021. URL https://api.semanticscholar.org/CorpusID:235349100.

Yifan Wu, George Tucker, and Ofir Nachum. The laplacian in rl: Learning representations with efficient approximations. In *International Conference on Learning Representations (ICLR)*, 2019.

Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

Mengjiao Yang and Ofir Nachum. Representation matters: Offline pretraining for sequential decision making. *CoRR*, abs/2102.05815, 2021. URL https://arxiv.org/abs/2102.05815.

Rui Yang, Jie Wang, Zijie Geng, Mingxuan Ye, Shuiwang Ji, Bin Li, and Feng Wu. Learning task-relevant representations for generalization via characteristic functions of reward sequence distributions. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2242–2252, 2022.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *CoRR*, abs/2107.09645, 2021. URL https://arxiv.org/abs/2107.09645.

Bang You and Huaping Liu. Multimodal information bottleneck for deep reinforcement learning with multiple sensors. *Neural Networks*, 2024. Accepted for publication.

Tao Yu, Cuiling Lan, Wenjun Zeng, Mingxiao Feng, Zhizheng Zhang, and Zhibo Chen. Playvirtual: Augmenting cycle-consistent virtual trajectories for reinforcement learning. *Advances in Neural Information Processing Systems*, 34:5276–5289, 2021.

Tao Yu, Zhizheng Zhang, Cuiling Lan, Yan Lu, and Zhibo Chen. Mask-based latent reconstruction for reinforcement learning. *Advances in Neural Information Processing Systems*, 35:25117–25131, 2022.

Zhecheng Yuan, Guozheng Ma, Yao Mu, Bo Xia, Bo Yuan, Xueqian Wang, Ping Luo, and Huazhe Xu. Don't touch what matters: Task-aware lipschitz data augmentation for visual reinforcement learning. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 3702–3708. International Joint Conferences on Artificial Intelligence Organization, 7 2022a. doi: 10.24963/ijcai.2022/514. URL https://doi.org/10.24963/ijcai.2022/514. Main Track.

Zhecheng Yuan, Zhengrong Xue, Bo Yuan, Xueqian Wang, Yi Wu, Yang Gao, and Huazhe Xu. Pre-trained image encoder for generalizable visual reinforcement learning. *Advances in Neural Information Processing Systems*, 35:13022–13037, 2022b.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pp. 12310–12320. PMLR, 2021.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *CoRR*, abs/2006.10742, 2020. URL https://arxiv.org/abs/2006.10742.

Peng Zhang, Yuanyuan Ren, and Bo Zhang. A new embedding quality assessment method for manifold learning. *Neurocomputing*, 97:251–266, 2012.

Wancong Zhang, Anthony GX-Chen, Vlad Sobal, Yann LeCun, and Nicolas Carion. Lightweight probing of unsupervised representations for reinforcement learning. *Reinforcement Learning Journal*, 4:1924–1949, 2024a.

Wancong Zhang, Anthony GX-Chen, Vlad Sobal, Yann LeCun, and Nicolas Carion. Lightweight probing of unsupervised representations for reinforcement learning, 2024b. URL https://arxiv.org/abs/2208.12345.

Tony Zhao, Siddharth Karamcheti, Kollar Thomas, and Chelsea Finn. What makes representation learning from videos hard for control. In *RSS Workshop on Scaling Robot Learning*, 2022.

Chongyi Zheng, Ruslan Salakhutdinov, and Benjamin Eysenbach. Contrastive difference predictive coding, 2024a.

Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang. Taco: Temporal latent action-driven contrastive loss for visual reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024b.

Zeyu Zheng, Vivek Veeriah, Risto Vuorio, Richard L. Lewis, and Satinder Singh. Learning state representations from random deep action-conditional predictions. *CoRR*, abs/2102.04897, 2021. URL `https://arxiv.org/abs/2102.04897`.

Qi Zhou, Jie Wang, Qiyuan Liu, Yufei Kuang, Wengang Zhou, and Houqiang Li. Learning robust representation for reinforcement learning with distractions by reward sequence prediction. In *Uncertainty in Artificial Intelligence*, pp. 2551–2562. PMLR, 2023.

A. Supplementary Content

Types of representation distance $\hat{d}$

$$d_{L1}(x_t, x_{t'}) = \sum_{i=1}^{n} |x_t^{(i)} - x_{t'}^{(i)}| \tag{1}$$

$$d_{L2}(x_t, x_{t'}) = \sqrt{\sum_{i=1}^{n} \left(x_t^{(i)} - x_{t'}^{(i)}\right)^2} \tag{2}$$

$$d_{\text{angular}}(x_t, x_{t'}) = \frac{\arccos\left(\frac{x_t \cdot x_{t'}}{\|x_t\|\|x_{t'}\|}\right)}{\pi} \tag{3}$$

$$d_{\text{cosine}}(x_t, x_{t'}) = \frac{x_t \cdot x_{t'}}{\|x_t\|\|x_{t'}\|} \tag{4}$$