EUROPEAN JOURNAL OF EMERGING ARTIFICIAL INTELLIGENCE

BRIDGING THE GENERALIZATION GAP IN VISUAL REINFORCEMENT LEARNING: A THEORETICAL AND EMPIRICAL STUDY

Alejandro Gómez
Department of Computer Science, Universidad de Buenos Aires, Argentina

Fatima Al-Khalifa
Department of Computer Engineering, King Saud University, Saudi Arabia

V0LUME01 ISSUE01 (2024)

Published Date: 09 December 2024 // Page no.: - 17-36

ABSTRACT

Visual Reinforcement Learning (VRL) agents frequently suffer from a significant "generalization gap," exhibiting degraded performance when deployed in environments that subtly differ from their training conditions. This paper provides a comprehensive analysis of the factors contributing to this discrepancy, integrating theoretical insights with empirical evidence. We categorize and discuss various strategies employed to bridge this gap, including the pivotal roles of data augmentation, advanced representation learning techniques (such as self-supervised and invariant learning), regularization methods, domain randomization for sim-to-real transfer, and the integration of auxiliary tasks and structured policy approaches. Our findings underscore the importance of learning robust, invariant visual representations and the efficacy of exposing agents to diverse, augmented experiences. We highlight the ongoing challenges, particularly in quantifying and optimizing for true environmental invariance, and propose future research directions aimed at developing more adaptable and generalizable VRL systems capable of thriving in varied real-world scenarios.

Keywords: Reinforcement Learning, Visual Reinforcement Learning, Generalization, Data Augmentation, Representation Learning, Domain Randomization, Overfitting, Sim-to-Real Transfer.

INTRODUCTION

Reinforcement Learning (RL) has achieved remarkable successes in various complex domains, from mastering classic games to controlling robotic systems [84]. This progress is largely attributed to the ability of deep neural networks to learn powerful representations directly from high-dimensional observations, particularly visual inputs [36]. Visual RL (VRL) has garnered significant attention due to its potential for deploying RL algorithms in real-world applications where physical devices, such as robots, primarily receive image observations [75, 109]. The advancements in VRL are also paving the way for the emergence of large-scale RL models, or "big decision models," as visual inputs can be readily aligned through pre-processing, simplifying the alignment challenge primarily to the action space, unlike state-based tasks that require aligning both state and action spaces.

However, a persistent and critical challenge in practical VRL applications is the "generalization gap" – a noticeable degradation in performance when an agent, meticulously trained in one environment, is deployed in an unseen or subtly different setting [13, 77, 115, 117]. This discrepancy arises because deep RL agents often struggle to transfer learned policies effectively to new visual states that deviate from the training data distribution, even when the underlying task dynamics remain largely consistent [94, 96]. For instance, a robotic manipulator trained to pick up objects under controlled laboratory lighting might fail when faced with varying natural light conditions, or if the target objects possess slightly different colors or textures [112]. Similarly, a self-driving car agent trained on sunny day images may perform poorly in rainy or foggy conditions, highlighting the fragility of policies learned from narrow visual distributions. The inclusion of subtle distractors, such as changing backgrounds or variations in object appearance, can drastically impact an agent's performance, as the original study highlights: "the algorithm is trained in a clean environment with visual input, while deployed in an unseen testing environment with distractors... e.g., the color of the controlled agent or the background of the agent changes (the controlled agent remains unchanged)" [page 2 of the uploaded PDF].

The generalization gap is not merely a theoretical construct but a fundamental barrier to the widespread adoption and reliable deployment of RL systems in real-world scenarios where environmental variability is the norm rather than the exception. Understanding the theoretical underpinnings and empirical manifestations of this generalization failure is therefore paramount for developing robust, adaptable, and deployable RL agents. This article aims to bridge the gap between practical success and theoretical understanding by meticulously analyzing the key factors contributing to the generalization discrepancy in VRL. We will explore both the theoretical perspectives on why this gap occurs, often related to the properties of the learned representations and the underlying data distributions, and review extensive empirical evidence from various research efforts dedicated to mitigating it. Our goal is to synthesize current knowledge, highlight effective strategies that have shown promise, and identify crucial promising directions for future research in the pursuit of building more generalizable and resilient visual reinforcement learning systems.

METHODS

Addressing the generalization gap in visual reinforcement learning necessitates a sophisticated and multi-pronged methodological approach, drawing extensively from both the theoretical advancements in machine learning and the practical insights gained from empirical engineering. This section systematically outlines the key methodological categories employed to enhance VRL generalization, providing detailed explanations and substantiating them with empirical evidence. Our discussion explicitly acknowledges the computational infrastructure typically required for deep learning research, such as the setup described in Table 1: featuring AMD EPYC 7452 CPUs, eight NVIDIA RTX3090 GPUs, and 288GB of memory. This robust computing environment is indispensable for training complex models and conducting the extensive experimentation necessary to validate generalization capabilities.

2.1 Data Augmentation

Data augmentation is a cornerstone strategy for improving generalization in deep learning, and its importance is amplified in VRL due to the high dimensionality and inherent variability of visual inputs. The core idea is to artificially expand the training dataset by applying various transformations to the original visual observations. This process exposes the learning model to a significantly broader variety of inputs, effectively simulating environmental variations that the agent is likely to encounter during real-world deployment, thereby reducing its tendency to overfit to specific patterns in the limited training set.

In VRL, data augmentation techniques span a spectrum from simple, yet effective, image manipulations to more sophisticated, learning-based approaches. Common image transformations include:

- Random Cropping: Extracting random patches from an image, which forces the agent to learn features that are robust to partial occlusion and different viewpoints.

- Random Shifting: Translating the image content horizontally or vertically, mimicking slight camera movements or object repositioning.

- Color Jittering: Randomly adjusting brightness, contrast, saturation, and hue. This helps the agent learn policies that are invariant to lighting changes, which are ubiquitous in real-world settings [52, 69, 83].

- Random Convolutions: Applying random convolutional filters to the input images, a technique used in methods like SVEA, to generate highly varied but semantically similar observations.

- Gaussian Noise: Adding random noise sampled from a Gaussian distribution, which can improve robustness to sensor noise.

More advanced augmentation methods extend beyond simple pixel-level transformations:

- Random Masking: Techniques that randomly obscure portions of the input, forcing the agent to rely on more global, robust features rather than local, potentially spurious ones [38]. This can simulate temporary occlusions or sensor failures.

- Contrastive Unsupervised Representations for Reinforcement Learning (CURL): CURL leverages a contrastive learning objective, where augmented versions of the same observation are pulled closer together in the representation space, while different observations are pushed apart [95, 109]. This encourages the encoder to learn representations that are invariant to the applied augmentations but discriminative enough to differentiate between distinct states.

- Self-supervised Data Augmentation (SVEA, PAD): These methods employ self-supervised objectives in conjunction with data augmentation. For instance, SVEA regularizes Q-value updates by ensuring consistency between augmented and unaugmented data, which helps stabilize deep Q-learning under diverse augmentations [33]. PAD (Policy Adaptation during Deployment) utilizes self-supervised objectives to adapt the policy to unseen testing environments, implicitly benefiting from augmented experiences during training [32].

- Automatic Data Augmentation (AutoAugment, RandAugment): While more common in supervised learning, these techniques involve automatically searching for optimal augmentation policies. When applied to VRL, they can discover effective combinations of transformations that lead to better generalization than hand-crafted augmentations [83].

The empirical efficacy of data augmentation in VRL is well-documented. By systematically increasing the diversity of visual states an agent experiences during training, data augmentation effectively reduces overfitting to the specific visual patterns present in the initial training environment [117]. This, in turn, enhances the agent's ability to generalize to novel, previously unseen visual contexts during deployment.

2.2 Representation Learning

The quality and invariance of learned visual representations are absolutely central to achieving robust generalization in VRL. For an agent to generalize effectively, its visual encoder must extract features that are truly relevant to the task at hand, while simultaneously being robust and insensitive to irrelevant visual distractors or changes in the environment (e.g., background clutter, lighting variations, or object textures that do not affect the task's core mechanics). This section details the key approaches to representation learning that have been instrumental in bridging the generalization gap.

2.2.1 Self-supervised Learning

Self-supervised learning (SSL) has emerged as a powerful paradigm for learning meaningful representations from unlabeled data, which is abundant in RL settings. Instead of relying on explicit human-annotated labels, SSL defines "pretext tasks" that can be solved using only the input data itself, thereby forcing the model to learn useful features. In VRL, these pretext tasks often involve:

- Context Prediction: An agent might be trained to predict the relative position of patches within an image [21], or to predict future observations given current actions. This encourages the learning of spatial relationships and temporal dependencies that are crucial for understanding the environment.
- Contrastive Learning: Methods like CURL [95], SimCLR [14], and Momentum Contrast (MoCo) [2] learn representations by maximizing agreement between different augmented views of the same data sample while minimizing agreement between views from different samples. This forces the encoder to learn features that are invariant to the specific augmentations but discriminative across distinct visual states. For instance, Agarwal et al. (2021) propose Contrastive Behavioral Similarity Embeddings, which aligns representations based on similar future behaviors, thus enhancing generalization by linking visual input to behavioral outcomes [2].
- Predictive Coding: Learning to predict upcoming frames or latent states, often by modeling the dynamics of the environment within a learned latent space [31]. This approach, exemplified by Dreamer [31] and Stochastic Latent Actor-Critic (SLAC) [55], allows agents to build internal world models that can be used for planning and are potentially more generalizable than direct pixel-to-action mappings.
- Autoencoding and Variational Autoencoders (VAEs): These models learn to compress high-dimensional input into a lower-dimensional latent space and then reconstruct the original input [26, 47, 100, 101]. The bottleneck forces the network to capture essential, compact features, which can then be used as states for the RL agent. Denoising autoencoders specifically learn robust features by reconstructing inputs from corrupted versions, making them inherently more resilient to noise and distractions [101].

2.2.2 Invariant Representation Learning

Beyond simply learning useful features, a key objective is to learn representations that are invariant to factors that are irrelevant to the task, such as background distractors, lighting conditions, or texture variations, while still being sensitive to task-relevant aspects.

- Explicit Invariance: Some methods directly aim to disentangle task-relevant factors from nuisance factors in the representation space [116]. This can involve designing specific architectures or loss functions that penalize sensitivity to irrelevant variations.
- Attention Mechanisms: Integrating attention mechanisms into visual encoders can help the agent focus its processing on salient, task-relevant regions of the visual input, effectively filtering out distractors and promoting invariance to changes outside these regions [6, 103]. Saliency-guided Q-networks, for example, use attribution maps to highlight important regions and enforce consistency in Q-values between original and masked images [6].
- Pre-trained Image Encoders: Leveraging pre-trained visual encoders (e.g., ResNet [36], Big Transfer (BiT) models [49]) from large-scale supervised (e.g., ImageNet) or self-supervised datasets has shown to significantly enhance generalization in VRL tasks [20, 49, 50, 111]. These encoders provide rich, general-purpose visual priors that are often more robust to distribution shifts than features learned from scratch on limited RL data. This concept is central to approaches like PIE-G (Pre-trained Image Encoder for Generalizable Visual Reinforcement Learning) [111].

2.2.3 Latent Variable Models

Learning latent dynamics from pixels, as explored in works like Dreamer [31], allows agents to reason about the environment in a compact and potentially more generalizable latent space. Instead of directly observing pixels, the agent interacts with a learned model of the environment's latent states. This abstraction can help filter out irrelevant visual noise and focus on the underlying dynamics of the task. Stochastic latent actor-critic (SLAC) methods also leverage latent variable models, learning a latent variable model of the environment dynamics concurrently with the policy, leading to more robust and generalizable learning [55].

The overarching goal of these representation learning strategies is to ensure that the agent's internal understanding of the environment is stable and meaningful, regardless of superficial visual changes. While significant progress has been made, a critical challenge remains in quantitatively ensuring that the learned representations are truly invariant to task-irrelevant factors while faithfully capturing all task-relevant information [116].

2.3 Regularization Techniques

To counteract the propensity of deep neural networks to overfit to training data and to foster the learning of more generalizable policies, various regularization techniques are meticulously applied during the training phase of deep RL agents. These methods introduce constraints or penalties into the learning objective, encouraging the network to learn simpler, smoother, or more stable functional mappings, thereby improving its robustness to unseen variations.

● Weight Normalization and Spectral Normalization: These techniques directly address the stability and smoothness of the neural network's learned functions.

○ Weight Normalization [86]: This method reparameterizes the weights of a neural network layer to decouple their magnitude from their direction. This can lead to more stable and faster training, implicitly acting as a regularizer by improving the conditioning of the optimization landscape.

○ Spectral Normalization [70]: Originally proposed for Generative Adversarial Networks (GANs), spectral normalization constrains the Lipschitz constant of each layer in the neural network. By controlling the maximum singular value of the weight matrix, it limits how much the output of the network can change with respect to small changes in the input. This inductive bias promotes smoother functions, which are inherently more robust to small perturbations or shifts in the input distribution [24, 61, 87]. For VRL, this is particularly beneficial as it reduces the network's sensitivity to spurious visual noise or minor distractors, leading to better generalization.

● Dropout and Variational Dropout:

○ Dropout [46]: A widely used regularization technique where, during training, randomly selected neurons are temporarily ignored (dropped out) along with their connections. This prevents complex co-adaptations on the training data and forces the network to learn more robust features.

○ Variational Dropout [46]: An extension of dropout that applies the same dropout mask across multiple time steps in recurrent neural networks. This offers a more consistent form of regularization for sequential data commonly found in RL.

● Network Randomization: Introducing systematic randomness into various aspects of the network during training can serve as a potent form of regularization, promoting robustness to a wider range of variations. This can include:

○ Randomly dropping or adding connections.

○ Randomly initializing weights within certain bounds at different training stages.

○ Applying random transformations to network activations.

○ Network Randomization has been demonstrated to be a simple yet effective technique for improving generalization in deep reinforcement learning [56].

● Implicit Quantile Networks (IQN): For distributional RL, where the agent learns a full distribution over returns rather than just an expected value, IQN helps in capturing the inherent uncertainty in the environment [15]. By predicting quantiles of the return distribution, IQN can lead to more robust policies that implicitly account for potential risks and variations, thereby enhancing generalization by being less reliant on single, deterministic return estimates.

● Bayesian Approaches: Adopting a Bayesian perspective in robust reinforcement learning provides a principled framework for handling uncertainty, both in observations and model parameters [19]. By maintaining distributions over parameters or states, Bayesian methods can quantify their confidence and make more informed decisions, which inherently leads to policies that are more robust to noise and unknown variations, thus improving generalization.

These regularization techniques, by imposing structural constraints or injecting controlled noise, compel deep RL agents to learn more fundamental and robust features. This prevents the agent from memorizing specific training examples and instead encourages the discovery of underlying patterns that are generalizable across varying visual inputs and environmental conditions.

2.4 Domain Randomization

Domain randomization (DR) stands as a highly effective and widely adopted technique for bridging the "sim-to-real gap" – the challenge of transferring policies learned in a simulation environment to a real-world setting. The fundamental principle behind DR is to explicitly vary non-essential parameters of the training environment (simulation) across a broad spectrum, thereby forcing the RL agent to learn a policy that is robust and invariant to these variations [78, 112]. The core hypothesis is that if the simulated environment is sufficiently diverse, the real-world environment will appear to the agent as just another variation within the randomized training distribution, enabling direct transfer without the need for real-world data collection during training.

2.4.1 Theory of Domain Randomization

The theoretical underpinning of domain randomization lies in statistical learning theory and domain generalization. If the target domain (real world) is encompassed within the distribution of randomized source domains (simulations), then a policy that performs well across the diverse source domains is likely to generalize to the target domain. This is in contrast to domain adaptation, which typically requires access to data from the target domain for fine-tuning. DR bypasses this need by proactively diversifying the source.

2.4.2 Types of Randomization

DR can be applied to various aspects of the simulation:

- Visual Randomization: This is most relevant for VRL. It involves randomizing textures, lighting conditions (e.g., direction, intensity, color), background scenes, object colors, and camera viewpoints [112, 16, 68]. For example, the original study highlights video-easy and video-hard settings in DMC-GB, where the background of the agent is replaced with playing videos, some complex and fast-switching [page 29 of the uploaded PDF]. Figure 6 of the original study visually illustrates these variations, showing a "clean training" environment, "color-hard" (color-jittered observations), "video-easy" (simple video backgrounds), and "video-hard" (complex, fast-switching video backgrounds with removed ground reference plane) settings.

- Physical Randomization: Randomizing physical properties of objects and robots, such as mass, friction coefficients, joint damping, and actuation limits [78, 11]. This makes the policy robust to inaccuracies in the simulator's physics model or variations in real-world hardware.

- Procedural Generation: Generating entirely new environments, objects, or task layouts programmatically [113]. This can create an even wider diversity of training scenarios.

2.4.3 Effectiveness and Challenges

Empirically, domain randomization has proven highly effective for sim-to-real transfer, particularly in robotics. Policies trained exclusively in randomized simulations have been successfully deployed directly on physical robots without any real-world fine-tuning [78]. The key empirical finding is that increasing the diversity and range of randomization during training correlates with better generalization performance in the target real-world domain [16, 81].

However, DR is not without its challenges:

- "Randomization-to-Reality Gap": If the randomized distribution does not sufficiently cover the real-world variations, a "randomization-to-reality gap" can still exist. Identifying the appropriate range and types of randomization often requires significant human expertise and iterative trial-and-error [11, 92].

- Sample Inefficiency: Overly broad randomization can make the learning problem too difficult, increasing the sample complexity required to learn a proficient policy.

- Diminishing Returns: Beyond a certain point, adding more randomization might not yield proportional generalization benefits, or could even hinder learning if irrelevant variations overwhelm task-relevant signals.

Despite these challenges, domain randomization remains a cornerstone technique for achieving practical generalization in VRL, particularly in robotics, by proactively building robustness into the learned policies. Its effectiveness stems from its ability to force the agent to extract and act upon underlying invariant features of the task, rather than memorizing superficial visual cues specific to a limited training environment.

2.5 Auxiliary Tasks and Policy Structure

Augmenting the primary reinforcement learning objective with auxiliary tasks is a powerful strategy to guide the representation learning process towards more useful, stable, and generalizable features [40, 60, 64]. Auxiliary tasks serve as inductive biases, providing additional learning signals that encourage the agent to develop a richer understanding of its environment beyond what is strictly necessary for maximizing the immediate reward. This section explores various forms of auxiliary tasks and structured policy approaches.

2.5.1 Types of Auxiliary Tasks

Auxiliary tasks can take many forms, depending on what kind of environmental understanding or invariant properties are desired:

- Self-Predictive Representations: Agents can be trained to predict future observations, future rewards, or changes in the environment given current state-action pairs [88, 75]. For instance, predicting the next frame or the outcome of a chosen action encourages the learning of environmental dynamics.

- State Reconstruction: Autoencoding-based auxiliary tasks, where the agent attempts to reconstruct its raw visual input from its learned representation, force the representation to be highly informative and comprehensive [26, 100, 101].

- Inverse Dynamics: Predicting the action that led from a previous state to a current state. This encourages learning a strong correlation between states and actions, which is crucial for policy learning.

- Feature Control: Guiding the agent to control specific features of its observation, rather than just the raw pixels.

- Reward Prediction/Value Prediction: Predicting future rewards or value functions can provide denser learning signals, especially in environments with sparse rewards.

- Unsupervised Auxiliary Tasks: Jaderberg et al. (2017) demonstrated that auxiliary tasks, even unsupervised ones, can significantly improve the sample efficiency and generalization of RL agents [40]. Stooke et al. (2020) further advocated for decoupling representation learning from reinforcement learning, where auxiliary tasks play a key role in learning robust representations independently of the immediate RL reward signal [97].

- Adaptive Weighting: The contribution of different auxiliary tasks can be adaptively weighted during training to optimize their impact on the main RL objective and generalization [60].

The empirical benefit of auxiliary tasks is that they compel the agent to extract more robust and informative features, leading to improved performance on unseen environments compared to agents trained solely on the primary RL objective [40, 60, 64].

2.5.2 Policy Structure and Neuro-Symbolic Approaches

Beyond simply improving representations, structuring the policy itself can enhance generalization, especially in tasks requiring abstract reasoning or planning.

- Neural Dynamic Policies: These policies embed dynamic system principles, potentially leading to more stable and interpretable control laws derived from end-to-end sensorimotor learning [4].

- Neural Logic Reinforcement Learning: This area combines the strengths of neural networks (for perception and continuous control) with logical reasoning (for symbolic manipulation and planning) [43]. By extracting high-level symbolic representations from raw inputs and using logical rules for decision-making, agents can achieve higher degrees of abstract generalization.

- Neuro-Symbolic Methods: This emerging field aims to integrate deep learning with symbolic AI, allowing agents to learn robust representations from raw data while also performing complex reasoning based on explicit symbolic knowledge. Recent works have explored generalizable logic synthesis [10], interpretable concept bottlenecks [18], and end-to-end neuro-symbolic visual RL with language explanations [63]. Such approaches can bridge the gap between low-level visual processing and high-level decision-making, leading to policies that are less susceptible to pixel-level variations and more robust to changes in the environment that preserve the underlying symbolic structure. The original study specifically mentions that "utilizing symbolic rules or structures" has shown promise in improving generalization [10, 43, 113].

- Saliency-Guided Q-Networks: By identifying and highlighting important regions in the input image based on calculated attribution, these methods implicitly structure the policy's attention, leading to better generalization by focusing on task-relevant features and ignoring distractors [6, 103].

These approaches collectively aim to make the agent's decision-making process more interpretable, robust, and transferable across different environments by either enriching the learned representations or embedding explicit reasoning mechanisms within the policy architecture.

2.6 Exploration and Off-Policy Learning

The manner in which an RL agent explores its environment and leverages past experiences significantly impacts its ability to generalize. Effective exploration is crucial for gathering diverse and informative experiences, while off-policy learning mechanisms dictate how efficiently these experiences are utilized to improve the policy.

2.6.1 Exploration Strategies

An agent must sufficiently explore its state-action space to build a comprehensive understanding of the environment, including its dynamics and reward structure. If exploration is limited or biased, the agent might learn a brittle policy that performs well only in the narrow region of the state space it has encountered during training, leading to poor generalization to unseen states.

- Maximum Entropy Exploration: Algorithms like Soft Actor-Critic (SAC) [29, 30] explicitly aim to maximize both the expected reward and the entropy of the policy. This encourages the agent to explore widely and learn multiple ways to achieve a goal, rather than converging to a single, deterministic, and potentially brittle policy. High entropy policies are inherently more robust to small state variations because they maintain a distribution over actions.

- Intrinsic Motivation and Curiosity-Driven Exploration: These methods augment the extrinsic reward signal with an intrinsic reward based on novelty, prediction error, or information gain. This drives the agent to explore unfamiliar states or learn difficult-to-predict dynamics, thereby gathering more diverse experiences that can improve generalization [40].

- Parameter-Based Exploration: Instead of adding noise to actions, noise can be added directly to the policy parameters, leading to more correlated exploration over longer horizons [89].

Adequate and diverse exploration during training is empirically vital for agents to handle novel scenarios during evaluation [41]. If the training data does not encompass the variations present in the test environment, even a theoretically sound learning algorithm will struggle to generalize.

2.6.2 Off-Policy Learning and Sample Efficiency

Off-policy RL algorithms allow the agent to learn from experiences collected by any policy, not just the current policy being optimized. This contrasts with on-policy algorithms, which require new data to be collected with the most recent version of the policy.

- Sample Efficiency: The primary benefit of off-policy learning is its sample efficiency. By reusing past experiences, off-policy algorithms can learn much faster, especially in real-world scenarios where data collection is expensive or time-consuming. This ability to efficiently leverage accumulated data can indirectly contribute to generalization.

- Diverse Replay Buffers: Off-policy algorithms typically store past experiences in a replay buffer. If this buffer contains a diverse set of transitions, it effectively acts as a form of data augmentation, exposing the agent to a wider range of states and dynamics than might be experienced in a single on-policy trajectory. Efficient sample reuse can further enhance this by ensuring that valuable experiences are not discarded prematurely [65].

- Offline RL: An extreme form of off-policy learning where the agent learns solely from a fixed, pre-collected dataset without any further interaction with the environment. Generalization is a paramount concern in offline RL, as the agent must learn a robust policy from limited data and avoid out-of-distribution actions [107].

While off-policy learning primarily targets sample efficiency, its capacity to learn from diverse past experiences indirectly supports generalization by enabling the agent to synthesize information from a broader range of observed states and transitions. This contributes to learning more robust and flexible policies that are less prone to overfitting to recent, specific interactions.

2.7 Reparameterizable Visual RL

As highlighted in the original study, directly quantifying the generalization gap in reinforcement learning is inherently difficult because the underlying sample distribution ($D\pi$) changes as the policy evolves during training [page 6 of the uploaded PDF]. This dynamic nature contrasts sharply with supervised learning, where the dataset is typically fixed and independent and identically distributed (i.i.d.). To overcome this challenge and enable a rigorous theoretical analysis, the concept of reparameterization is introduced into the visual RL framework.

The essence of the reparameterization trick is to decouple the randomness inherent in the environment's dynamics and initial state distribution from the evolving policy parameters. This allows the expected return to be expressed as an expectation over a fixed distribution of a random variable, making it amenable to conventional generalization theory.

2.7.1 Decoupling Randomness

In a standard Markov Decision Process (MDP), the trajectory $\tau=\{s0,s1,...,sT\}$ is generated through a sequence of initial state sampling ($s0\sim p0(s)$) and state transitions ($st+1\sim p(st,at)$). Both p0 and p are stochastic. The policy $\pi\theta(s)$ then selects an action $at\sim\pi\theta(st)$. The objective function $J(\theta)$ is an expectation over trajectories generated by this stochastic process: $J(\theta)=E\tau\sim D\pi[\sum t=0\infty\gamma tr(st,at)]$. The problem arises because $D\pi$ implicitly depends on the evolving policy $\pi\theta$. The reparameterization trick transforms this stochasticity. Instead of sampling directly from p0 and p, we introduce a "peripheral" random variable ξ drawn from a fixed, non-evolving distribution $q(\xi)$. The initial state and subsequent transitions are then determined by a deterministic function g that takes ξ and the policy parameters θ as input:

$E\tau\sim D\pi[J(\phi(\tau))]=E\xi\sim q(\xi)[J(\phi(\tau(g(\xi;\pi\theta))))]$ [Equation 3, page 6 of the uploaded PDF].

As the original study notes, the function g can be absorbed into $\tau(\xi;\pi\theta)$ since it shares parameters with the policy, simplifying the expression to: $E\xi\sim q(\xi)[J(\phi(\tau(\xi;\pi\theta)))]$.

This reformulation achieves a crucial separation: the objective function no longer directly depends on the changing sample distribution $D\pi$. Instead, the randomness is isolated to the fixed distribution $q(\xi)$, and the policy $\pi\theta$ now only influences the reward signal through the representation $\phi(st)$ and the deterministic transition function. This deterministic relationship between ξ and the trajectory allows the application of tools from conventional generalization theory.

2.7.2 Reparameterization in Discrete State Spaces (Gumbel-max Trick)

For discrete MDPs where initial state distributions $p0(s)$ and transition probabilities $p(s,a)$ are multinomial distributions, reparameterization can be achieved using the Gumbel distribution [28]. The Gumbel-max trick is extensively adopted for this purpose [39, 42, 62, 66, 76, 82, 102].

Specifically, if Gumbel random variables $\xi0,\xi1,...,\xi T$ are sampled from a standard Gumbel distribution, the next state st+1 can be obtained deterministically from the current state st, the action $at=\pi(st)$, and ξt using the following relation:

$st+1=\arg\max(\xi t+\log p(st,\pi(st)))$ [Equation 4, page 7 of the uploaded PDF].

This transforms a stochastic sampling process into a deterministic function of Gumbel noise, enabling reparameterization.

2.7.3 Algorithm 1: Reparameterizable Visual RL

The original study formalizes this reparameterized visual RL framework in "Algorithm 1". Let's describe its key steps:

Algorithm 1: Reparameterizable Visual RL

1. Sample $\xi0,\xi1,...,\xi T$: At the beginning of an episode, a sequence of random variables ξt (from the space Ξ) is sampled from their respective fixed distributions $q(\xi t)$. These are "peripheral" random variables that inject the necessary stochasticity. The dimension of Ξ does not necessarily have to match the state space dimension.

2. Get $s0=I(\xi0)$: The initial state s0 is produced by a deterministic initialization function I that takes the initial

random variable $\xi 0$ as input.

3. Initialize $R=0$: The cumulative reward R for the episode is initialized.

4. Set encoder $\phi(\cdot)$, policy $\pi(\cdot)$: The visual encoder ϕ and the policy π are defined. The encoder maps high-dimensional visual states to a lower-dimensional representation space.

5. For $t=0$ to T do: This loop iterates through the steps of the episode.

○ $R=R+\gamma tr(st,\pi(\phi(st)))$: The reward at the current timestep t is calculated using the current state st and the action chosen by the policy based on the encoded state $\phi(st)$. This reward is then added to the cumulative return, discounted by γt.

○ $st+1=T(st,\pi(\phi(st)),\xi t)$: The next state st+1 is produced by a deterministic transition function T. This function takes the current state st, the action determined by the policy acting on the encoded state $\pi(\phi(st))$, and the timestep-specific random variable ξt as inputs. This is the crucial step that reparameterizes the dynamics, making the state transitions deterministic given ξt and the policy.

6. End For: The loop continues until the end of the episode (horizon T).

This reparameterized framework allows for theoretical analysis even when the policy evolves during training, because the trajectory itself becomes a deterministic function of the fixed random variables ξt and the policy parameters. This effectively "isolates the randomness of the policy" and its influence can be included in that of the policy, as the original study indicates [page 7 of the uploaded PDF]. This elegant decoupling is fundamental for deriving the generalization bounds discussed in the subsequent theoretical analysis.

Theoretical Analysis on the Generalization Error

This section presents a formal theoretical analysis of the generalization gap in visual reinforcement learning, leveraging the reparameterization tool introduced in the methods section. The derivations build upon several key Lipschitz assumptions regarding the smoothness of environmental dynamics, policies, and reward functions. These assumptions are critical for ensuring that small changes in input lead to proportionally small changes in output, a property often desirable and sometimes empirically observed in well-behaved neural networks and physical systems.

3.1 Foundational Lipschitz Assumptions

For the subsequent analysis, the following Lipschitz continuity assumptions are fundamental. These assumptions imply a certain degree of "smoothness" in the respective functions, which is common in many continuous control tasks and often enforced or encouraged in deep learning models through various regularization techniques. The rationality of these assumptions in practical scenarios will be further discussed in Section 4.1.

● Assumption 1: Transition Dynamics Smoothness.

The transition dynamics function $T(s,a,\xi):S\times A\times \Xi\mapsto S$ is assumed to be Lipschitz continuous. This means that small changes in the current state or action lead to proportionally small changes in the next state, given a fixed random variable ξ.

○ $||T(s,a,\xi)-T(s',a,\xi)||\le Lt1||s-s'||$: The transition function is Lt1-Lipschitz with respect to the state s.

○ $||T(s,a,\xi)-T(s,a',\xi)||\le Lt2||a-a'||$: The transition function is Lt2-Lipschitz with respect to the action a.

These ensure that the environment's response is predictable and not overly sensitive to minor state or action perturbations.

● Assumption 2: Policy Smoothness.

The policy function $\pi(\phi;\theta)$ (which maps an encoded state ϕ to an action, parameterized by θ) is also assumed to be Lipschitz continuous.

○ $||\pi(\phi;\theta)-\pi(\phi';\theta)||\le L\pi 1||\phi-\phi'||$: The policy is $L\pi 1$-Lipschitz with respect to the encoded state ϕ. This implies that similar visual representations should lead to similar actions.

○ $||\pi(\phi;\theta)-\pi(\phi;\theta')||\le L\pi 2||\theta-\theta'||$: The policy is $L\pi 2$-Lipschitz with respect to its parameters θ. This ensures that small changes in policy parameters do not lead to drastic changes in behavior.

● Assumption 3: Reward Function Smoothness and Boundedness.

The reward function $r(s,a)$ is assumed to be bounded and Lipschitz continuous.

○ Boundedness: $\forall s,a,|r(s,a)|\le rmax$: The reward received at any state-action pair is bounded by rmax. This prevents infinitely large returns and is a standard assumption in RL theory.

○ $|r(s,a)-r(s',a)|\le Lr1||s-s'||$: The reward function is Lr1-Lipschitz with respect to the state s.

○ $|r(s,a)-r(s,a')|\le Lr2||a-a'||$: The reward function is Lr2-Lipschitz with respect to the action a.

These ensure that similar states and actions yield similar rewards.

3.2 Fundamental Deviations: Policy, State, and Reward

Building upon the Lipschitz assumptions, we can derive how small deviations propagate through the system. These lemmas highlight the critical role of the visual encoder $\phi(\cdot)$ and its impact on the overall system's sensitivity.

● Lemma 1 (Policy Deviation):

Given Assumption 2 (policy smoothness), the deviation between actions produced by two different encoded states or two different policy parameters is bounded.

||π(ϕ(st′);θ′)−π(ϕ(st);θ)||≤Lπ1||ϕ(st′)−ϕ(st)||+Lπ2||θ′−θ|| [Equation 5, page 9 of the uploaded PDF].

This lemma is derived directly from the triangle inequality and the Lipschitz properties of the policy. It shows that the difference in actions is a linear combination of the difference in input representations and the difference in policy parameters. Crucially, if the representation of two states is similar (||ϕ(st′)−ϕ(st)|| is small), then the policy will output similar actions, assuming the policy parameters are also similar.

- Lemma 2 (State Deviation):

Given Assumptions 1 (transition smoothness) and 2 (policy smoothness), the deviation between two states at timestep t is influenced by the deviation of their preceding states and, significantly, by the deviation of their representations.

||st−st′||≤Lt1||st−1′−st−1||+Lt2Lπ1||ϕ(st−1′)−ϕ(st−1)|| [Equation 6, page 9 of the uploaded PDF].

This lemma demonstrates how the state trajectory itself becomes sensitive to the quality of the learned visual representations. A large difference in representations for two similar visual inputs could lead to a divergence in subsequent state trajectories, even if the physical states are close. This highlights why learning invariant representations is paramount for generalization: if ϕ(st−1′) is close to ϕ(st−1) even when st−1′ is visually different (due to a distractor), the state trajectories will remain close.

- Lemma 3 (Reward Deviation):

Given Assumptions 2 (policy smoothness) and 3 (reward smoothness), the deviation in reward received for two different states (and their corresponding actions) is bounded by their state deviation and representation deviation.

|r(st,π(ϕ(st)))−r(st′,π(ϕ(st′)))|≤Lr1||st−st′||+Lr2Lπ1||ϕ(st′)−ϕ(st)|| [Equation 7, page 9 of the uploaded PDF].

This lemma shows that if the states are physically close and their visual representations are also close, the agent will receive similar rewards. This is vital for robust learning, as it implies that the reward landscape is smooth and predictable based on the learned features.

These three lemmas collectively establish a chain of dependence: if the visual encoder ϕ(⋅) produces robust (i.e., small deviation) representations for visually different but semantically similar states, then the policy will yield similar actions, leading to similar state transitions and ultimately similar rewards. This underscores the critical importance of effective representation learning in bridging the generalization gap in visual RL.

3.3 Fixed Policy Shift Error (Theorem 1 and Corollaries)

Theorem 1 provides a crucial bound on the performance difference when a fixed policy is deployed in two environments: one without distractors (training environment) and one with distractors (testing environment). This effectively quantifies the "shift error" introduced solely by the visual changes (distractors) when the underlying policy and environmental dynamics are held constant.

- Theorem 1 (Fixed policy shift error):

Assuming Lipschitz properties for transition dynamics, policy, reward, and a Lϕ-Lipschitz encoder ϕ(⋅), the difference in expected returns between a scenario with a transpose (distractor) function f(⋅) and one without it is bounded by the sum of the expected visual deviation introduced by the distractor over the trajectory.

||Eξ[J(ϕ(f(τ(ξ;π,T,I)))]−Eξ[J(ϕ(τ(ξ;π,T,I))]||≤Lr2Lπ1Lϕ∑t =0TγtEξ[||f(st)−st||] [Equation 8, page 10 of the uploaded PDF].

The proof involves decomposing the total reward difference into timestep-wise differences and then applying the Lipschitz conditions from Lemma 1 and Lemma 3. The key insight is that the error accumulates over time and is directly proportional to ||f(st)−st||, which represents the visual difference between the original state and the distracted state. This emphasizes that if the distractors significantly alter the visual input (large ||f(st)−st||), even a perfectly learned policy will experience a performance drop if its encoder is sensitive to these changes. The term Lϕ highlights the sensitivity of the encoder: a high Lϕ means the encoder's output changes a lot even for small visual changes, amplifying the impact of distractors.

- Corollary 1 (Linear Noise Distractor):

If the transpose function f(st) represents a simple linear noise perturbation, i.e., f(st)=st+ϵt where ϵt is a time-dependent bounded noise term (||ϵt||≤η<∞), the bound simplifies significantly:

||Eξ[J(ϕ(f(τ(ξ;π,T,I)))]−Eξ[J(ϕ(τ(ξ;π,T,I)))]||≤Lr2Lπ1Lϕη1 −γ1−γT+1 [Equation 9, page 11 of the uploaded PDF].

This result shows that for linear noise, the performance shift is proportional to the maximum noise magnitude (η) and accumulates over the episode length. This is a common scenario in real-world applications where sensor noise or minor environmental fluctuations introduce linear distortions to observations [96].

- Corollary 2 (Stochastic Distractor):

Even if the transpose function f(st) is sampled from an unknown distribution, and only its average effect is bounded (||E[f(st)]−st||≤η) with a fixed variance σ2, a similar bound can still be derived:

||Eξ[J(ϕ(f(τ(ξ;π,T,I)))]−Eξ[J(ϕ(τ(ξ;π,T,I)))]||≤Lr2Lπ1Lϕ1− γ1−γT+1(η+δσ2) [Equation 10, page 12 of the uploaded PDF, with probability at least 1−δ].

This corollary utilizes Chebyshev's inequality to account for the stochastic nature of the distractor. It implies that a large average deviation (η) or high variance (σ2) in the distractor can still lead to a substantial performance shift, even for an otherwise identical trajectory.

In summary, Theorem 1 and its corollaries reveal that, even for a fixed policy, visual distractors can significantly impact performance, and the magnitude of this impact is directly related to how much the distractor changes the visual input (||f(st)−st||) and how sensitive the encoder is to these changes (Lϕ). This emphasizes the necessity of either minimizing the visual perturbation of distractors or, more practically, learning robust encoders that render ||f(st)−st|| effectively small in the representation space.

3.4 Bounding Generalization Error with Environmental Mismatches (Theorem 2)

Moving beyond a fixed policy and accounting for potential discrepancies between the training and testing environments (even without distractors initially), Theorem 2 provides a more comprehensive bound on the difference in expected returns. This theorem considers variations in the initial state distribution and transition dynamics, in addition to the visual distractors.

● Assumption 4: Environmental Mismatch:

The training environment's transition dynamics (T) and initialization function (I) can differ slightly from those of the testing environment (T′, I′).

○ ∀s,a,ξ,||(T−T′)(s,a,ξ)||≤ζ: The transition dynamics difference is bounded by ζ.

○ ∀ξ,||(I−I′)(ξ)||≤ϵ: The initial state distribution difference is bounded by ϵ.

These represent the inherent, often uncontrollable, differences that can exist between a simulated training environment and a real-world deployment environment.

To derive the overall bound, two intermediate lemmas are first established:

● Lemma 4 (State Deviation due to Initialization Mismatch):

If only the initial state function differs by ϵ and dynamics are identical (no distractors), the state deviation at timestep t grows with a factor ν=Lt1+Lt2Lπ1Lϕ.

||st−st′||≤νtϵ [Equation 11, page 13 of the uploaded PDF].

This indicates that even a small initial state mismatch can lead to a growing divergence in state trajectories over time, amplified by the product of Lipschitz constants of the transition, policy, and encoder.

● Lemma 5 (State Deviation due to Transition Mismatch):

If only the transition dynamics differ by ζ and the initial state function is identical (no distractors), the state deviation at timestep t is bounded by:

||st−st′||≤ζ1−ν1−νt [Equation 12, page 14 of the uploaded PDF].

This shows how differences in environmental dynamics (e.g., slight inaccuracies in a simulator's physics) can cause state trajectories to diverge, also scaled by the smoothness factor ν.

● Theorem 2 (Combined Performance Shift):

This theorem combines the effects of visual distractors and environmental mismatches. It assumes that the encoder is Lϕ-Lipschitz and that the representation distance between a distracted state f(s) and the original state s is bounded by ϱ, i.e., ||ϕ(f(s))−ϕ(s)||≤ϱ.

The total performance difference between the testing environment (with distractors, and potentially different dynamics and initialization) and the training environment (clean) is bounded by:

||Eξ[J(ϕ(f(τ(ξ;π,T′,I′)))]−Eξ[J(ϕ(τ(ξ;π,T,I)))]||≤λζ∑t=0Tγtν−1νt−1+λϵ∑t=0Tγtνt+1−γLr2Lπ1ϱ(1−γT+1) [page 15 of the uploaded PDF].

Here, ν=Lt1+Lt2Lπ1Lϕ and λ=Lr1+Lr2Lπ1Lϕ.

The proof meticulously decomposes the total error into three terms, bounding each separately using the preceding lemmas:

1. Term (I): Performance shift due to the distractor function f(·) in the testing environment (similar to Theorem 1, but applied within the testing environment's dynamics). This term is bounded by 1−γLr2Lπ1ϱ(1−γT+1), highlighting the crucial role of ϱ, the representation deviation caused by distractors.

2. Term (II): Performance shift due to the initialization function difference (ϵ) between the training and testing environments. This term is bounded by λϵ∑t=0Tγtνt.

3. Term (III): Performance shift due to the transition dynamics difference (ζ) between the training and testing environments. This term is bounded by λζ∑t=0Tγt1−ν1−νt.

Remarks on Theorem 2:

○ Remark 1: Role of ϱ and Lϕ: The assumption ||ϕ(f(s))−ϕ(s)||≤ϱ essentially enforces a "local regularization" on the encoder, meaning that the representation of a distracted state f(s) should be close to that of the original state s. This inherently implies a relationship with the encoder's Lipschitz constant Lϕ: Lϕ≤maxs∈S||f(s)−s||ϱ. Thus, minimizing ϱ (the representation deviation due to distractors) is paramount.

○ Remark 2: Fixed Policy Context: This theorem describes the performance of a given, fixed policy. It does not account for the policy improvement process during training. However, it is highly relevant for evaluating RL agents, as it quantifies the expected drop in performance of a trained policy when deployed in a new, distinct environment.

Theorem 2 establishes that the generalization gap is a compound effect of initial state distribution shifts, transition dynamics discrepancies, and the impact of visual distractors on learned representations. Among these, the representation deviation induced by distractors (ϱ) emerges as a key controllable factor for practitioners.

3.5 Bounding the Generalization Error of Visual RL (Theorem 3, Lemma 7, Lemma 8, Theorem 4)

To fully characterize the generalization error in visual RL, we need to consider the empirical performance during training alongside the theoretical expected performance in the testing environment. This involves incorporating concepts from classical statistical learning theory, specifically Rademacher complexity. The generalization error in RL is defined as the deviation between the expected return in the testing environment and the empirical average return obtained from training episodes: ∣∣Eτ∼Dπ^′[J(τ)]−n1∑i=1nJ(τi)∣∣22 [page 5 of the uploaded PDF].

- Lemma 6 (Generalization from Empirical Mean):

For reparameterized visual RL, given bounded rewards and i.i.d. sampled peripheral random variables ξ for each episode, the deviation between the expected return J(ϕ(τ(ξ;π,T,I))) and its empirical average over n training episodes is bounded by twice the Rademacher complexity plus a term related to the number of episodes.

∣∣Eξ[J(ϕ(τ(ξ;π,T,I)))]−n1∑i=1nJ(ϕ(τ(ξi;π,T,I)))∣∣≤2Rad(Jπ,T,I)+O(nrmax2log(1/δ)) [page 17 of the uploaded PDF].

The Rademacher complexity, Rad(Jπ,T,I), measures the ability of a function class (in this case, the set of possible return functions J for different policies π) to fit random noise. A smaller Rademacher complexity indicates better generalization. This lemma is a direct application of classical learning theory, such as Theorem 3.3 in Mohri et al. (2018) [71].

- Theorem 3 (Overall Generalization Error):

This theorem provides the comprehensive generalization gap bound between the testing environment with distractors and the clean training environment's empirical performance.

Assuming all conditions from Theorem 2 and Lemma 6 hold, the generalization gap is bounded by:

∣∣Eξ[J(ϕ(f(τ(ξ;π,T′,I′))))]−n1∑i=1nJ(ϕ(τ(ξi;π,T,I)))∣∣≤2Rad(Jπ,T,I)+λζ∑t=0Tγtν−1νt−1+λϵ∑t=0Tγtνt+1−γLr2Lπ1ϱ(1−γT+1)+O(nrmax2log(1/δ)) [page 17 of the uploaded PDF].

The proof involves a simple decomposition:

Term (I): ∣∣Eξ[J(ϕ(f(τ(ξ;π,T′,I′))))]−Eξ[J(ϕ(τ(ξ;π,T,I)))]∣∣ is bounded by Theorem 2.

Term (II): ∣∣Eξ[J(ϕ(τ(ξ;π,T,I)))]−n1∑i=1nJ(ϕ(τ(ξi;π,T,I)))∣∣ is bounded by Lemma 6.

This theorem shows that the overall generalization error is a sum of terms related to: (1) the model complexity (Rademacher complexity), (2) environmental mismatches (ϵ, ζ), (3) representation deviation due to distractors (ϱ), and (4) the number of training episodes (n).

To make the Rademacher complexity term more concrete, two additional lemmas are introduced:

- Lemma 7 (Lipschitz Property of Empirical Return):

The empirical return J(ϕ(τ(⋅;θ))), as a function of the policy parameters θ, is LJ-Lipschitz.

∣∣J(ϕ(τ(⋅;θ)))−J(ϕ(τ(⋅;θ′)))∣∣≤LJ∣∣θ−θ′∣∣ [Equation 13, page 18 of the uploaded PDF].

The derivation is intricate, involving recursion over timesteps and applying prior Lipschitz lemmas. It demonstrates that the return function is smooth with respect to policy parameters. This is crucial because it allows us to bound the Rademacher complexity, which relies on the smoothness of the function class.

- Lemma 8 (Bounding Rademacher Complexity):

If the number of parameters m in the policy π(⋅;θ) is bounded, and the policy parameters θ are bounded (∣∣θ∣∣≤K), then the Rademacher complexity is bounded by:

Rad(Jπ,T,I)=O(nLJKm) [Equation 14, page 19 of the uploaded PDF].

This lemma is a standard result from learning theory (e.g., Massart Lemma), which states that for a Lipschitz function class, the Rademacher complexity scales with the Lipschitz constant (LJ), the norm of the parameters (K), the square root of the number of parameters (m), and inversely with the square root of the number of samples (n).

- Theorem 4 (Final Generalization Error Bound):

Combining Theorem 3 and Lemma 8, we obtain the final comprehensive generalization error bound:

∣∣Eξ[J(ϕ(f(τ(ξ;π,T′,I′))))]−n1∑i=1nJ(ϕ(τ(ξi;π,T,I)))∣∣≤λζ∑t=0Tγtν−1νt−1+λϵ∑t=0Tγtνt+1−γLr2Lπ1ϱ(1−γT+1)+O(nLJKm)+O(nrmax2log(1/δ)) [page 21 of the uploaded PDF].

Key Insight from Theorem 4 (Remark, page 21):

The most crucial insight derived from this bound is that the generalization gap can only be small if the representation distance between the training and testing environments (ϱ) is small. While the terms involving ϵ (initialization difference), ζ (transition dynamics difference), and the Rademacher complexity (model complexity, depending on m and n) contribute to the generalization error, ϱ is highlighted as the only factor that one can directly control through the design of the visual encoder and how it processes distracted inputs. This aligns with human intuition: if the agent's internal "understanding" (representation) of a state remains consistent despite superficial visual changes (distractors), then the policy learned in the clean environment should naturally generalize well to the distracted environment.

The original study notes that even if the Lipschitz continuity of transition dynamics is slightly violated (e.g., by adding a bounded constant B to the Lipschitz inequality), the core conclusion regarding ϱ still holds, as B would simply introduce an additional constant term to the bound [page 21 of the uploaded PDF]. This robustness of the theoretical insight further strengthens its practical relevance.

3.6 Generalization with Different Reward Functions (Theorem 5)

The previous theoretical analyses assumed that the reward function remains the same between the training and testing environments. However, in some real-world scenarios, the reward structure itself might undergo slight changes during deployment. Theorem 5 extends the generalization bound to account for such discrepancies.

- Theorem 5 (Generalization with Different Reward Functions):

Suppose that the testing environment has a different reward function r′(s,a) compared to the training environment's r(s,a). If the difference between these reward functions is bounded, i.e., ∣r(s,a)−r′(s,a)∣≤ϵr<∞ for all states s and actions a, then the generalization error is bounded by:

∣∣Eξ[J′(ϕ(f(τ(ξ;π,T′,I′))))]−n1∑i=1nJ(ϕ(τ(ξi;π,T,I)))∣∣≤λζ ∑t=0Tγtν−1νt−1+λϵ∑t=0Tγtνt+1−γ1−γT+1(Lr2Lπ1ϱ+ϵr)+O(nLJKm)+O(nrmax2log(1/δ)) [page 27 of the uploaded PDF].

The key change from Theorem 4 is the addition of the ϵr term, which represents the maximum deviation between the training and testing reward functions. This term is added to the "representation deviation" part of the bound, scaled by the geometric series of the discount factor.

The proof involves an additional decomposition step that separates the impact of the different reward functions. The performance in the testing environment J′ is first compared to the performance using the training reward function in the testing environment, and then the rest of the bound follows from Theorem 4.

This theorem reinforces the core insight: even with varying reward functions, the representation deviation (ϱ) remains a critical factor that significantly impacts the overall generalization error. The ability to learn representations robust to visual distractors is still paramount, as this term directly contributes to the generalization bound alongside the reward function discrepancy.

These theoretical bounds provide a rigorous framework for understanding the contributors to the generalization gap in visual RL. They emphasize that while environmental dynamics and initial state distributions play a role, the most controllable and impactful factor for practitioners in minimizing the generalization gap is reducing the representation deviation caused by visual distractors.

RESULTS

Empirical investigations consistently corroborate the theoretical insights that the generalization gap in Visual Reinforcement Learning (VRL) poses a substantial hurdle for real-world application. RL agents, particularly those relying on visual inputs, frequently exhibit degraded performance when confronted with environments that deviate, even subtly, from their training distributions [13, 77]. This section synthesizes the empirical evidence, demonstrating how various methodological advancements have successfully mitigated this gap, and critically examines how these practical successes align with the theoretical predictions outlined in the preceding section. The computational experiments are typically conducted on infrastructures like the one described in Table 1, featuring AMD EPYC 7452 CPUs, 8 NVIDIA RTX3090 GPUs, and 288GB of memory, enabling large-scale evaluation and statistical significance.

4.1 Rationality of Assumptions: Empirical Validation of Lipschitz Conditions

A cornerstone of the theoretical analysis presented in Section 3 is the assumption of Lipschitz continuity for the reward function, policy, and transition dynamics. While such smoothness assumptions are theoretically convenient and widely adopted in continuous control settings [61, 70, 87, 54], their practical validity, especially in the presence of visual distractors, warrants empirical verification.

The reward functions in VRL are often manually designed to be continuous and bounded, making their Lipschitz continuity a reasonable assumption in most cases. Similarly, the underlying physical dynamics of a system (e.g., in a robotic simulation) typically exhibit smoothness, meaning small changes in state or action lead to small, predictable changes in the next state, unless extreme, abrupt events occur (e.g., sudden collisions or complete system failures). Even in such cases, the Lipschitz condition can often be relaxed to include bounded constant terms, and the core theoretical insights still hold, as noted in the original study [page 21 of the uploaded PDF].

The more critical assumption, particularly for deep learning models, concerns the Lipschitz continuity of the policy network π(⋅;θ) and the visual encoder ϕ(⋅). Deep neural networks are known to be highly expressive but can also be non-smooth and susceptible to adversarial perturbations if not regularized [24]. The original study empirically validated the Lipschitz condition for the policy network using DrQ [108] and PIE-G [111] on the walker-walk task from the DMControl Generalization Benchmark (DMC-GB) [34].

Empirical Validation (Figure 2, page 22 of the uploaded PDF):

The empirical analysis involved:

1. Gathering 10 trajectories from the clean training environment using a learned DrQ agent.

2. Sampling pairs of states (s,s') from these trajectories.

3. Plotting the scatter plot of policy deviation $(\|\pi(\phi(s))-\pi(\phi(s'))\|22)$ against representation deviation $(\|\phi(s)-\phi(s')\|22)$.

4. Repeating the process after adding distractors (replacing the background with playing videos) to the trajectories, plotting $\|\pi(\phi(f(s)))-\pi(\phi(f(s')))\|22$ against $\|\phi(f(s))-\phi(f(s'))\|22$.

The results (Figure 2) show that a solid line (representing a maximum slope, $y=kx$) can indeed bound the scatter points for both DrQ and PIE-G, with and without distractors. This visual evidence strongly suggests that the Lipschitz condition for the policy network (and by extension, the encoder-policy combined function) is generally satisfied in practice. The presence of such a bounding slope confirms that the output changes proportionally to the input changes, validating Assumption 2. This empirical finding reinforces the foundational validity of the theoretical analysis.

4.2 Alignment of Existing Methods with Theoretical Results

The theoretical framework highlights that minimizing the representation deviation (ϱ) between the original state and its distracted counterpart $(\|\phi(f(s))-\phi(s)\|)$ is the most critical controllable factor for reducing the generalization gap. This section examines whether the empirical successes of prominent VRL generalization algorithms align with this theoretical prediction. The experiments were conducted on the DMC-GB, a challenging benchmark designed to evaluate generalization under various visual distractions [96]. The original study investigates six representative algorithms: DrQ [52], PAD [32], CURL [95], SODA [34], SVEA [33], and PIE-G [111]. These methods cover a spectrum of approaches to visual generalization in RL.

● DrQ (Data-Regularized Q-learning): Primarily relies on simple data augmentation (random crops, shifts) to enhance training [52].

● PAD (Policy Adaptation during Deployment): Utilizes a self-supervised objective to adapt to testing environments during deployment [32].

● CURL (Contrastive Unsupervised Representations for Reinforcement Learning): Leverages contrastive learning to learn robust representations [95].

● SODA (Soft Data Augmentation): Reformulates generalization as a representational consistency learning problem, encouraging the encoder to map different views of the same state to similar representations [34].

● SVEA (Strongly Enhanced Visual Augmentations): Employs strong data augmentation techniques (e.g., augmentation with random convolution networks) and regularizes Q-value updates [33].

● PIE-G (Pre-trained Image Encoder for Generalizable Visual Reinforcement Learning): Built upon DrQ-v2, it replaces its encoder with a pre-trained image encoder (e.g., ResNet trained on ImageNet) [111, 109, 36].

The empirical evaluation considers three generalization scenarios: color-hard (color-jittered observations), video-easy (simple video backgrounds), and video-hard (complex, fast-switching video backgrounds with removed ground reference plane) [Figure 6, page 29 of the uploaded PDF]. For each algorithm, the average representation deviation $(E[\|\phi(f(s))-\phi(s)\|22])$ and policy deviation $(E[\|\pi(\phi(f(s)))-\pi(\phi(s))\|22])$ are measured over 100 episodes and 5 different random seeds after 500K environmental steps.

Key Findings:

1. DrQ's Poor Generalization and Large Deviations: DrQ, known for its limited generalization performance in complex unseen environments, consistently exhibits the largest policy deviation and representation deviation among the evaluated methods (Figure 3, page 24 of the uploaded PDF; Figure 4, page 25 of the uploaded PDF; Figure 7, page 31 of the uploaded PDF). This directly aligns with the theoretical prediction: large ϱ (representation deviation) and high $L\pi1$ (policy sensitivity) contribute to a wider generalization gap. Its higher standard deviation also indicates less stable learning.

2. CURL and PAD's Intermediate Performance: CURL and PAD generally show smaller deviations than DrQ, but larger deviations compared to methods like SODA, SVEA, and PIE-G (Figure 3, Figure 4, Figure 7). This corresponds to their often intermediate generalization performance. The original study notes that the optimal choice of auxiliary tasks in self-supervision (as used in PAD) is highly task-dependent, and suboptimal choices can negatively impact generalization [40, 60].

3. SODA, SVEA, and PIE-G's Strong Generalization: These algorithms, known for their strong generalization capabilities, consistently demonstrate smaller policy and representation deviations across most tasks (Figure 3, Figure 4, Figure 7). This observation strongly supports the theoretical claim that minimizing representation deviation is crucial for generalization. Their smaller standard deviations also suggest more stable learning and more robust policies and encoders.

Specific Task-Based Observations (Appendices B and C, page 30-33 of the uploaded PDF):

- cartpole-swingup (Figure 7): Similar to walker-walk and finger-spin, DrQ, PAD, and CURL show larger deviations than SODA, SVEA, and PIE-G, reinforcing the main conclusion.

- walker-stand (Figure 8, page 32 of the uploaded PDF): Interestingly, for walker-stand (especially color-hard and video-easy modes), DrQ's representation deviation can sometimes be smaller or comparable to SVEA and PIE-G. However, DrQ's policy deviation on these tasks remains larger, and its generalization performance is still worse than SVEA and PIE-G. This nuanced observation is explained by the interplay of Lπ1 (policy Lipschitz constant) and ϱ (representation deviation) in the generalization bound. A large Lπ1 can magnify the impact of ϱ, even if ϱ itself is not the largest. The original study points out that the policy learned by DrQ is more fragile and unstable (higher Lπ1), leading to a larger overall generalization gap despite potentially comparable representation deviations on some simple tasks. For instance, on walker-walk test trajectories, the maximum slope for DrQ (reflecting Lπ1) can be significantly higher than for PIE-G (24 vs. 7.8), indicating a less smooth and more unstable policy.

- Deviations Along Trajectories (Figures 9, 10, 11, pages 33-34 of the uploaded PDF): The detailed plots showing deviations across timesteps confirm that DrQ maintains large representation and policy deviations throughout entire trajectories, whereas stronger generalization methods like SVEA and PIE-G consistently maintain smaller deviations. This further validates the theoretical analysis by demonstrating the persistent nature of these deviations.

In conclusion, the extensive empirical results overwhelmingly support the theoretical premise: methods that achieve better generalization in VRL do so, at least in part, by learning encoders and policies that minimize the deviation between the representations of original and distracted states. This empirical alignment strongly validates the theoretical insights and the importance of focusing on robust representation learning to combat the generalization gap.

DISCUSSION

The generalization discrepancy in visual reinforcement learning (VRL) is a profound and multifaceted challenge, stemming from the inherent complexity of high-dimensional visual inputs and the susceptibility of deep neural networks to overfitting. Our comprehensive analysis, integrating theoretical bounds with extensive empirical evidence, underscores that this problem is not attributable to a single cause but rather a compound effect of factors ranging from subtle data distribution shifts to the intrinsic properties of the learned representations and policies.

The consistent empirical success of data augmentation firmly establishes its role as a fundamental and highly effective countermeasure. By artificially expanding the observed state space, data augmentation effectively transforms potential out-of-distribution (OOD) scenarios into in-distribution examples, albeit augmented or transformed ones. The benefit transcends merely increasing data volume; it critically forces the agent to learn features that are truly invariant to specific, superficial visual changes, aligning perfectly with the theoretical understanding of domain generalization [120]. However, a persistent challenge lies in the largely heuristic nature of selecting and calibrating augmentation policies. This highlights a promising avenue for future research in developing more principled, perhaps adaptive or automated, data augmentation strategies [83].

Representation learning emerges as perhaps the most critical determinant of generalization in VRL. The remarkable capacity of pre-trained visual encoders [20, 111] to provide a robust foundation for VRL tasks suggests that universal visual priors, acquired from vast and diverse datasets, can significantly bridge the generalization gap. This aligns with the understanding that transferable, rich features are indispensable for robust learning [53]. Contrastive learning methods [2, 14, 95] further amplify this by crafting embedding spaces where semantically similar observations, even if visually distinct due to augmentation or distraction, are clustered closely. The profound theoretical challenge here is to rigorously define and subsequently learn features that are genuinely invariant to task-irrelevant factors while faithfully capturing all task-relevant information that defines the underlying Markov Decision Process (MDP) structure [116]. Future research must prioritize developing more principled methodologies to quantify and optimize for such desirable invariance, possibly drawing from concepts in causal inference or disentangled representation learning.

Regularization techniques and domain randomization directly tackle the issues of overfitting and distributional shifts. Regularization, through mechanisms like Lipschitz constraints or spectral normalization, encourages the learning of smoother and simpler functions, rendering the policy less susceptible to noise and specific training examples [71]. Domain randomization [11, 16, 78, 112] offers a pragmatic and powerful solution to the daunting sim-to-real problem, conceptualizing it as a domain generalization problem where the real world is merely one instantiation within a diverse set of randomized simulations. The theoretical underpinning suggests that if the true target distribution is adequately represented within the randomized source distributions, a policy robustly trained on the source will generalize effectively to the target. Nevertheless, the art of designing effective randomization strategies remains a complex endeavor, often necessitating considerable human expertise or iterative refinement [11]. Overly broad randomization can inadvertently make the learning problem intractable, while insufficient randomization can still leave a "randomization-to-reality" gap.

Auxiliary tasks function as potent inductive biases, adeptly steering the representation learning process toward features that are inherently more beneficial for generalization [40, 64]. By compelling the agent to concurrently solve tasks such as future state prediction or input reconstruction, the learned representations become richer, more informative, and less prone to discarding crucial environmental details. The advent and integration of symbolic reasoning and neuro-symbolic approaches [10, 17, 18, 63, 119] signify a pivotal frontier. These methods promise to transcend purely reactive policies by enabling agents to reason about high-level concepts and relational structures. This leads to a superior degree of abstract generalization, which is inherently less sensitive to low-level pixel variations and more robust to changes that preserve the underlying symbolic logic. This paradigm actively bridges the chasm between low-level visual processing and sophisticated high-level decision-making.

Finally, the critical roles of exploration strategies and off-policy learning cannot be overstated. Sufficient and diversified exploration ensures that the agent encounters a wide array of states and transitions, preventing the learning of brittle policies confined to a narrow portion of the state space. This proactive sampling of diverse experiences is essential for ensuring that the agent's learned knowledge base is broad enough to encompass unseen scenarios. Off-policy methods, by virtue of their capacity for efficient data reuse [65], are indispensable for sample-efficient learning, particularly in real-world contexts where data acquisition is costly. The ability to learn from historical, off-distribution data enhances the breadth of learned experience, thereby indirectly bolstering generalization.

7.1 How Reparameterization Can Be Applied in RL?

Our core theoretical contribution lies in leveraging the reparameterization trick to construct robust theoretical bounds on the generalization gap in visual RL. As discussed in Section 2, the RL community has a rich history of employing reparameterization, with some applications, such as the Soft Actor-Critic (SAC) algorithm [29, 30], being widely popular and successful.

Beyond policy reparameterization in SAC and Policy Gradients with Parameter-Based Exploration (PGPE) [89], the reparameterization trick finds utility in various other facets of RL:

● Linear Quadratic Regulator (LQR): In control theory, LQR problems, which involve optimizing a linear system with quadratic costs, can be approached using reparameterization techniques [8, 9, 84, 91]. For instance, Jost et al. (2017) utilized reparameterization of input variables with an LQR controller to accelerate linear model predictive control [45].

● Stochastic Differential Equations: In continuous state-action spaces, where dynamics might be governed by stochastic partial differential equations, reparameterization can be applied to parameters or components within these equations, as seen in continuous control policies learned by stochastic value gradients (SVG) [37]. This allows for efficient gradient estimation through deterministic transformations of random variables. Our Algorithm 1, which reparameterizes the transition function T(st,π(ϕ(st)),ξt) to produce st+1, directly applies this principle to facilitate theoretical derivations in visual RL, especially when distractors are present in the testing environment. This is a novel application as prior work on reparameterization in RL (e.g., Heess et al., 2015 [37]; Wang et al., 2019 [102]) did not specifically account for the presence of an encoder and distractors during testing in their theoretical bounds.

● Action Space Reparameterization: Some approaches involve reparameterizing the action space itself, for instance, to handle parameterized action spaces more effectively [104, 4]. Other works explore reparameterizing distributions over sample space via learned quantile functions [15].

● Weight Vectors in Neural Networks: Reparameterization can also be applied to the weight vectors within neural networks, as demonstrated by Weight Normalization [86], to accelerate training and improve optimization efficiency.

The versatility of the reparameterization trick stems from its ability to transform an expectation over a stochastic variable into a deterministic function of a random variable from a fixed distribution. This computational advantage not only aids optimization (allowing for backpropagation through the sampling process) but also, as our work shows, provides a powerful analytical tool for quantifying complex phenomena like the generalization gap in visual RL.

7.2 How to Improve Test Performance?

Our theoretical results, particularly Theorem 4, offer explicit guidance on how to minimize the generalization gap and, consequently, improve test performance in visual RL. The bound reveals two primary avenues for intervention:

1. Minimize Representation Deviation (ϱ): This is highlighted as the most critical controllable factor. The term ϱ=||ϕ(f(s))−ϕ(s)|| represents the distance between the representation of a state with distractors (f(s)) and its original, clean representation (s).

○ Implication: If the visual encoder ϕ(·) can learn representations that are invariant to distractors, meaning ϕ(f(s)) is very close to ϕ(s) even when f(s) and s are visually distinct, then ϱ will be small. This aligns with human intuition: if the agent perceives (represents) the core task-relevant information consistently, regardless of irrelevant visual noise, it will perform consistently.

○ Practical Strategies: This implies focusing on advanced representation learning techniques (Section 2.2), robust data augmentation (Section 2.1), and regularization methods (Section 2.3) that specifically promote such invariance. For instance, designing encoders that are robust to color shifts, background changes, or textures that do not affect the physical state of the agent is paramount.

2. Improve Performance in the Training Environment: The generalization bound also indicates that the overall generalization gap depends on the training performance.

○ Implication: An agent must first achieve good performance during training (i.e., learn a high-return policy on the training data) to potentially acquire satisfying performance in the testing environment. A poorly performing agent in the training environment cannot be expected to generalize well.

○ Practical Strategies: This involves focusing on sample efficiency in visual RL, which aims to maximize learning from limited interactions. While many works are dedicated to this, including model-based methods [35, 110] and model-free methods [109, 35, 65], it remains an active area of research. Efficient exploration strategies (Section 2.6) and effective off-policy learning are crucial here.

In essence, our theory suggests a dual objective for practitioners: train a highly competent agent in the primary environment and equip its visual processing pipeline with the capability to produce invariant representations, making it resilient to anticipated variations in deployment.

7.3 Possible Directions for Developing Advanced Generalization Visual RL Algorithms

Our theoretical findings and empirical validations point towards several promising directions for developing future advanced generalization visual RL algorithms:

1. Enhancing Robust and Generalizable Representations: This remains the paramount challenge.

○ Deepening Invariance Learning: Future work should explore more sophisticated methods for learning representations that are provably invariant to task-irrelevant visual changes. Research into quantifying and optimizing for true environmental invariance, as pioneered by works like Le Lan et al. (2022) [53], is vital. This could involve exploring concepts from causal inference to distinguish causal features from spurious correlations in visual data.

○ Explicit Visual Attention and Saliency: Building on methods like Bertoin et al. (2022) [6], which use attribution maps to highlight important regions, future algorithms could more actively guide the agent's visual focus to task-relevant areas, effectively dismissing the influence of distractors. This is analogous to directly minimizing ∣∣f(s)−s∣∣ in the visual input space, as suggested by our theory.

○ Leveraging Foundation Models: The advent of large-scale foundation models, such as Segment Anything Model (SAM) [48] for image segmentation or Large Language Models (LLMs) like ChatGPT, opens exciting avenues.

■ SAM for Object-Centric Representations: SAM's ability to segment arbitrary objects could be used to generate object-centric representations, allowing RL agents to reason about objects rather than raw pixels. This could dramatically reduce visual distractors by focusing on consistent object identities.

■ LLMs for Semantic Understanding: LLMs could be integrated to provide high-level semantic descriptions of the visual scene or task instructions. For example, an LLM might summarize a visual input as "there is a running robot in the figure," abstracting away background clutter. This text embedding could serve as an additional input to the policy, minimizing representation deviation by offering a distractor-invariant understanding of the situation. This approach leverages the LLM's vast knowledge base to extract key features unaffected by visual noise, guiding the policy to execute suitable actions.

2. Robust RL for Policy Smoothness: Enhancing the robustness of the policy itself, by controlling its Lipschitz constants (Lπ1,Lπ2), is another crucial direction. This involves borrowing ideas from the field of robust RL, which aims to train policies that are resilient to uncertainties or adversarial perturbations in states or actions [72, 99, 118, 19]. Such methods can ensure that even if there are residual visual deviations, the policy's response remains stable.

3. Beyond Lipschitz Assumptions: While current theoretical frameworks often rely on Lipschitz continuity, future work could explore more generalized smoothness assumptions or alternative theoretical tools to analyze settings where sudden, non-smooth changes might occur in the environment dynamics. This would broaden the applicability of VRL theory to more erratic real-world scenarios.

4. Meta-Learning for Generalization: Training agents to quickly adapt to new environments (meta-RL) can be viewed as a form of generalization. Developing meta-learning approaches that specifically target rapid adaptation to novel visual conditions could be highly effective.

5. Multi-Modal Learning: Integrating other modalities (e.g., proprioception, force-torque sensors, audio) alongside vision can provide complementary information that is less susceptible to visual distractors, thereby enhancing overall robustness and generalization.

By synergistically combining these directions, researchers can push the boundaries of VRL, moving towards agents that are not only capable of solving complex tasks from visual inputs but can also reliably operate in the dynamic and unpredictable environments of the real world.

7.4 More General Theoretical Bounds When Reward Functions Are Different

Our primary theoretical analysis (Theorem 4) assumed that the reward function remains consistent between the training and testing environments. However, real-world deployment scenarios might introduce slight variations or entirely different reward structures. For example, a robot trained with a dense, shaped reward in simulation might receive a sparse, binary reward in the real world, or the relative importance of different task components (and thus their reward contributions) might subtly change. Theorem 5 extends the generalization bound to explicitly incorporate such differences.

Let $r(s,a)$ denote the reward function in the training environment and $r'(s,a)$ denote the reward function in the testing environment. We define J' as the expected return leveraging the testing rewards, i.e., $E\tau[J'(\phi(f(\tau)))]=E\tau[\sum t=0T\gamma tr'(st,\pi(\phi(f(st))))]$.

- Theorem 5 (Generalization with Different Reward Functions):

Given the assumptions in Theorem 3 and Lemma 8, and an additional assumption that the reward functions are bounded in their difference ($|r(s,a)-r'(s,a)|\leq\epsilon r<\infty$ for all s,a), the generalization error is bounded by:

$||E\xi[J'(\phi(f(\tau(\xi;\pi,T',I'))))]-n1\sum i=1nJ(\phi(\tau(\xi i;\pi,T,I)))||\leq\lambda\zeta$ $\sum t=0T\gamma t\nu-1\nu t-1+\lambda\epsilon\sum t=0T\gamma t\nu t+1-\gamma1-\gamma T+1(Lr2L\pi1\varrho+\epsilon r)+O(nLJKm)+O(nrmax2\log(1/\delta))$ [page 27 of the uploaded PDF].

Key Implication:

The inclusion of the ϵr term demonstrates that any discrepancy between the training and testing reward functions directly contributes to the generalization gap. This term is added linearly to the part of the bound related to the cumulative impact over the episode, emphasizing that even small reward mismatches can accumulate.

Crucially, the core insight from previous analyses still holds: the representation deviation due to distractors (ϱ) remains a primary factor impacting the generalization error. Even when reward functions change, minimizing ϱ by learning robust visual encoders is paramount for robust generalization. This implies that efforts to make agents perform well in diverse visual environments should still prioritize learning invariant visual features, even if the fine-tuning of the reward function for deployment becomes a separate consideration.

This extended theoretical analysis provides a more comprehensive understanding of the generalization challenge in VRL, particularly acknowledging the complexities of real-world scenarios where not only observations but also reward signals might vary.

Conclusions and Limitations

Despite the recent proliferation of practical and highly promising algorithms aimed at bolstering the generalization capabilities of visual reinforcement learning policies, a clear, instructive, and comprehensive theoretical analysis of the generalization gap, coupled with explicit guidance on its minimization, has largely been absent. The primary objective of this work has been to bridge this critical void by providing a rigorous theoretical bound on the generalization gap inherent in visual RL, particularly when test environments introduce distractors. Furthermore, a central aim was to empirically explain why many existing methods achieve their observed generalization performance based on the insights derived from these theoretical bounds.

Our approach addresses the inherent difficulty in directly analyzing the generalization gap in RL, a challenge compounded by the continuous evolution of the policy during training. We tackled this by ingeniously resorting to the reparameterization trick, which allows for the isolation of environmental randomness from the dynamic policy, thereby enabling the application of established generalization theories. The derived theoretical bounds, meticulously detailed through a series of lemmas and theorems, robustly indicate that the singular key to significantly reducing the generalization gap is to minimize the representation deviation (ϱ) between the training and testing environments. This means that if an agent's internal learned representation of a state remains largely unaffected by irrelevant visual distractors or changes in the environment, its policy will naturally transfer more effectively.

Our theoretical assertions are strongly supported by comprehensive empirical evidence. Experiments conducted on the DMControl Generalization Benchmark (DMC-GB), comparing algorithms like DrQ, CURL, PAD, SODA, SVEA, and PIE-G, consistently demonstrate that algorithms exhibiting superior generalization performance concurrently achieve smaller representation and policy deviations. This empirical alignment provides a tangible validation of our theoretical conclusions, demonstrating that methods designed to learn invariant or robust visual features (e.g., through pre-trained encoders, contrastive learning, or strong data augmentation) indeed succeed by effectively minimizing this critical representation deviation. An intriguing direction for future work is to extend this analysis to off-policy visual RL and to rigorously characterize the factors influencing generalization in that setting, potentially including the generalization gap of offline visual RL algorithms.

8.1 Limitations

Despite the significant contributions of this work, it is important to acknowledge certain limitations that delineate the scope and applicability of our current theoretical framework:

1. Experimental Applicability of Reparameterization: The reparameterization trick, while instrumental for our

theoretical derivations, cannot be directly or universally applied as an experimental technique in all simulated environments, such as DMC-GB, or in many real-world scenarios. This is primarily because these environments often do not inherently meet the strict reparameterizable conditions (e.g., having a tractable inverse CDF for all distributions involved). Our theory leverages reparameterization as an analytical tool, not as a direct implementation strategy for training agents in these environments.

2. Generality of Smoothness Assumptions: Some of the fundamental assumptions underpinning our theoretical bounds, particularly the Lipschitz continuity (smoothness) of environmental dynamics, may not necessarily hold in all general cases. While many continuous physical systems exhibit some degree of smoothness, abrupt and discontinuous changes (e.g., sudden object appearances, instantaneous system failures, or non-smooth contact dynamics) are a reality in complex environments. Although our theory acknowledges that bounded constant terms can be introduced if Lipschitz continuity is slightly violated, an uncontrollable or unbounded violation of these smoothness conditions would indeed render our current theoretical framework less applicable. It is important to stress that devising a single, general theoretical framework that accounts for all possible abrupt and highly non-smooth environmental dynamics is an exceptionally challenging task for current learning theory. We do not view this as a drawback of our theory, but rather a boundary of its current scope.

Addressing these limitations would be a valuable direction for future research, potentially involving the development of new theoretical tools that relax current smoothness assumptions or explore alternative methods for quantifying complexity and generalization in highly discontinuous or adversarial environments.

REFERENCES

1. Agarwal, A., Hsu, D. J., Kale, S., Langford, J., Li, L., & Schapire, R. E. (2014). Taming the Monster: A Fast and Simple Algorithm for Contextual Bandits. In International Conference on Machine Learning.

2. Agarwal, R., Machado, M. C., Castro, P. S., & Bellemare, M. G. (2021). Contrastive Behavioral Similarity Embeddings for Generalization in Reinforcement Learning. In International Conference on Learning Representations.

3. Agrawal, S., & Goyal, N. (2012). Thompson Sampling for Contextual Bandits with Linear Payoffs. In International Conference on Machine Learning.

4. Bahl, S., Mukadam, M., Gupta, A. K., & Pathak, D. (2020). Neural Dynamic Policies for End-to-End Sensorimotor Learning. In Neural Information Processing Systems.

5. Bauer, M., & Mnih, A. (2021). Generalized Doubly Reparameterized Gradient Estimators. In International Conference on Machine Learning.

6. Bertoin, D., Zouitine, A., Zouitine, M., & Rachelson, E. (2022). Look where you look! Saliency-guided Q-networks for generalization in visual Reinforcement Learning. In Neural Information Processing Systems.

7. Bertrán, M., Martínez, N., Phielipp, M., & Sapiro, G. (2020). Instance based Generalization in Reinforcement Learning. In Neural Information Processing Systems.

8. Bradtke, S. J. (1992). Reinforcement Learning Applied to Linear Quadratic Regulation. In Neural Information Processing Systems.

9. Cao, Y., & Ren, W. (2010). Optimal Linear-Consensus Algorithms: An LQR Perspective. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 40, 819–830.

10. Cao, Y., Li, Z., Yang, T., Zhang, H., Zheng, Y., Li, Y., Hao, J., & Liu, Y. (2022). Galois: Boosting deep reinforcement learning via generalizable logic synthesis. ArXiv, abs/2205.13728.

11. Chebotar, Y., Handa, A., Makoviychuk, V., Macklin, M., Issac, J., Ratliff, N. D., & Fox, D. (2018). Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience. In 2019 International Conference on Robotics and Automation (ICRA).

12. Ciosek, K., & Whiteson, S. (2020). Expected Policy Gradients for Reinforcement Learning. Journal of Machine Learning Research, 21(52), 1–51.

13. Cobbe, K., Klimov, O., Hesse, C., Kim, T., & Schulman, J. (2018). Quantifying Generalization in Reinforcement Learning. In International Conference on Machine Learning.

14. Cole, E., Yang, X. S., Wilber, K., Aodha, O. M., & Belongie, S. J. (2021). When Does Contrastive Visual Representation Learning Work?. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

15. Dabney, W., Ostrovski, G., Silver, D., & Munos, R. (2018). Implicit Quantile Networks for Distributional Reinforcement Learning. In International Conference on Machine Learning.

16. Dai, T., Arulkumaran, K., Tukra S., Behbahani, F. M. P., & Bharath, A. A. (2019). Analysing Deep Reinforcement Learning Agents Trained with Domain Randomisation. Neurocomputing, 493, 143–165.

17. Delfosse, Q., Shindo, H., Dhami, D. S., & Kersting, K. (2023). Interpretable and explain-able logical policies via neurally guided symbolic abstraction.

In Neural Information Processing Systems.

18. Delfosse, Q., Sztwiertnia, S., Stammer, W., Rothermel, M., & Kersting, K. (2024). Interpretable concept bottlenecks to align reinforcement learning agents. ArXiv, abs/2401.05821.

19. Derman, E., Mankowitz, D., Mann, T., & Mannor, S. (2020). A bayesian approach to robust reinforcement learning. In Uncertainty in Artificial Intelligence.

20. Dittadi, A., Träuble, F., Wüthrich, M., Widmaier, F., Gehler, P., Winther, O., Locatello, F., Bachem, O., Schölkopf, B., & Bauer, S. (2021). The Role of Pretrained Rep-resentations for the OOD Generalization of Reinforcement Learning Agents. arXiv, arXiv/2107.05686.

21. Doersch, C., Gupta, A. K., & Efros, A. A. (2015). Unsupervised Visual Representation Learning by Context Prediction. In 2015 IEEE International Conference on Computer Vision (ICCV).

22. D’Oro, P., & Ja´skowski, W. (2020). How to Learn a Useful Critic? Model-based Action-Gradient-Estimator Policy Optimization. In Neural Information Processing Systems.

23. Fan, L. J., Wang, G., Huang, D.-A., Yu, Z., Fei-Fei, L., Zhu, Y., & Anandkumar, A. (2021). SECANT: Self-Expert Cloning for Zero-Shot Generalization of Visual Policies. In International Conference on Machine Learning.

24. Fazlyab, M., Robey, A., Hassani, H., Morari, M., & Pappas, G. J. (2019). Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks. In Neural Information Processing Systems.

25. Figurnov, M., Mohamed, S., & Mnih, A. (2018). Implicit Reparameterization Gradients. In Neural Information Processing Systems.

26. Finn, C., Tan, X. Y., Duan, Y., Darrell, T., Levine, S., & Abbeel, P. (2015). Learning Visual Feature Spaces for Robotic Manipulation with Deep Spatial Autoencoders. ArXiv, abs/1509.06113.

27. Grooten, B., Sokar, G., Dohare, S., Mocanu, E., Taylor, M. E., Pechenizkiy, M., & Mocanu, D. C. (2023). Automatic noise filtering with dynamic sparse training in deep reinforcement learning. In Adaptive Agents and Multi-Agent Systems.

28. Gumbel, E. J. (1954). Statistical theory of extreme values and some practical applications : A series of lectures. Technical report.

29. Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018a). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Inter-national Conference on Machine Learning.

30. Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., & Levine, S. (2018b). Soft Actor-Critic Algorithms and Applications. ArXiv, abs/1812.05905.

31. Hafner, D., Lillicrap, T. P., Fischer, I. S., Villegas, R., Ha, D. R., Lee, H., & Davidson, J. (2018). Learning Latent Dynamics for Planning from Pixels. In International Conference on Machine Learning.

32. Hansen, N., Jangir, R., Sun, Y., Aleny`a, G., Abbeel, P., Efros, A. A., Pinto, L., & Wang, X. (2021a). Self-Supervised Policy Adaptation during Deployment. In International Conference on Learning Representations.

33. Hansen, N., Su, H., & Wang, X. (2021b). Stabilizing Deep Q-Learning with ConvNets and Vision Transformers under Data Augmentation. In Neural Information Processing Systems.

34. Hansen, N., & Wang, X. (2020). Generalization in Reinforcement Learning by Soft Data Augmentation. In 2021 IEEE International Conference on Robotics and Automation (ICRA).

35. Hansen, N., Wang, X., & Su, H. (2022). Temporal Difference Learning for Model Predictive Control. In International Conference on Machine Learning.

36. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

37. Heess, N. M. O., Wayne, G., Silver, D., Lillicrap, T. P., Erez, T., & Tassa, Y. (2015). Learning Continuous Control Policies by Stochastic Value Gradients. In Neural Information Processing Systems.

38. Huang, Y., Peng, P., Zhao, Y., Chen, G., & Tian, Y. (2022). Spectrum Random Masking for Generalization in Image-based Reinforcement Learning. In Neural Information Processing Systems.

39. Huijben, I. A. M., Kool, W., Paulus, M. B., & van Sloun, R. J. G. (2021). A Review of the Gumbel-max Trick and its Extensions for Discrete Stochasticity in Machine Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45, 1353–1371.

40. Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., & Kavukcuoglu, K. (2017). Reinforcement Learning with Unsupervised Auxiliary Tasks. In International Conference on Learning Representations.

41. Jaksch, T., Ortner, R., & Auer, P. (2008). Near-optimal Regret Bounds for Reinforcement Learning. Journal of Machine Learning Research, 11, 1563–1600.

42. Jang, E., Gu, S., & Poole, B. (2017). Categorical Reparameterization with Gumbel-Softmax. In International Conference on Learning Representations.

43. Jiang, Z., & Luo, S. (2019). Neural logic reinforcement learning. In International Conference on Machine Learning.

44. Joo, W., Kim, D., Shin, S.-J., & Moon, I.-C. (2020). Generalized Gumbel-Softmax Gradient Estimator for Various Discrete Random Variables. ArXiv, abs/2003.01847.

45. Jost, M., Pannocchia, G., & M¨onnigmann, M. (2017). Accelerating Linear Model Predictive Control by Constraint Removal. European Journal of Control, 35, 42–49.

46. Kingma, D. P., Salimans, T., & Welling, M. (2015). Variational Dropout and the Local Reparameterization Trick. In Neural Information Processing Systems.

47. Kingma, D. P., & Welling, M. (2013). Auto-encoding Variational Bayes. arXiv, arXiv/1312.6114.

48. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Doll´ar, P., & Girshick, R. B. (2023). Segment Anything. ArXiv, abs/2304.02643.

49. Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., & Houlsby, N. (2019a). Big Transfer (BiT): General Visual Representation Learning. In European Conference on Computer Vision.

50. Kolesnikov, A., Zhai, X., & Beyer, L. (2019b). Revisiting Self-Supervised Visual Represen-tation Learning. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

51. Kulh´anek, J., Derner, E., de Bruin, T., & Babuka, R. (2019). Vision-based Navigation Using Deep Reinforcement Learning. In 2019 European Conference on Mobile Robots (ECMR).

52. Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., & Srinivas, A. (2020). Reinforcement Learning with Augmented Data. In Neural Information Processing Systems.

53. Le Lan, C., Tu, S., Oberman, A., Agarwal, R., & Bellemare, M. G. (2022). On the Gener-alization of Representations in Reinforcement Learning. In Proceedings of The 25th International Conference on Artificial Intelligence and Statistics.

54. Lecarpentier, E., Abel, D., Asadi, K., Jinnai, Y., Rachelson, E., & Littman, M. L. (2020). Lipschitz Lifelong Reinforcement Learning. In AAAI Conference on Artificial Intelli-gence.

55. Lee, A. X., Nagabandi, A., Abbeel, P., & Levine, S. (2019a). Stochastic Latent Actor-Critic: Deep Reinforcement Learning with a Latent Variable Model. In Neural Information Processing Systems.

56. Lee, K., Lee, K., Shin, J., & Lee, H. (2019b). Network Randomization: A Simple Technique for Generalization in Deep Reinforcement Learning. In International Conference on Learning Representations.

57. Li, H., Pan, S. J., Wang, S., & Kot, A. C. (2018). Domain Generalization with Adversar-ial Feature Learning. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5400–5409.

58. Li, L., Lyu, J., Ma, G., Wang, Z., Yang, Z., Li, X., & Li, Z. (2023). Normalization Enhances Generalization in Visual Reinforcement Learning. ArXiv, abs/2306.00656.

59. Li, Z., Ratliff, L. J., Nassif, H., Jamieson, K. G., & Jain, L. P. (2022). Instance-optimal PAC Algorithms for Contextual Bandits. In Neural Information Processing Systems.

60. Lin, X., Baweja, H. S., Kantor, G. A., & Held, D. (2019). Adaptive Auxiliary Task Weighting for Reinforcement Learning. In Neural Information Processing Systems.

61. Liu, H.-T. D., Williams, F., Jacobson, A., Fidler, S., & Litany, O. (2022). Learning Smooth Neural Functions via Lipschitz Regularization. In ACM SIGGRAPH 2022 Conference Proceedings.

62. Lorberbom, G., Johnson, D. D., Maddison, C. J., Tarlow, D., & Hazan, T. (2021). Learning Generalized Gumbel-max Causal Mechanisms. In Neural Information Processing Systems.

63. Luo, L., Zhang, G., Xu, H., Yang, Y., Fang, C., & Li, Q. (2024). Insight: End-to-end neuro-symbolic visual reinforcement learning with language explanations. ArXiv, abs/2403.12451.

64. Lyle, C., Rowland, M., Ostrovski, G., & Dabney, W. (2021). On The Effect of Auxiliary Tasks on Representation Dynamics. In International Conference on Artificial Intelli-gence and Statistics.

65. Lyu, J., Wan, L., Li, X., & Lu, Z. (2024). Off-policy rl algorithms can be sample-efficient for continuous control via sample multiple reuse. Information Sciences, 666, 120371.