CROP: Towards Distributional-Shift Robust Reinforcement Learning using Compact Reshaped Observation Processing *

Philipp Altmann , **Fabian Ritz** , **Leonard Feuchtinger** , **Jonas Nüßlein** , **Claudia Linnhoff-Popien** and **Thomy Phan**
LMU Munich
philipp.altmann@ifi.lmu.de

Abstract

The safe application of reinforcement learning (RL) requires generalization from limited training data to unseen scenarios. Yet, fulfilling tasks under changing circumstances is a key challenge in RL. Current state-of-the-art approaches for generalization apply data augmentation techniques to increase the diversity of training data. Even though this prevents overfitting to the training environment(s), it hinders policy optimization. Crafting a suitable observation, only containing crucial information, has been shown to be a challenging task itself. To improve data efficiency and generalization capabilities, we propose Compact Reshaped Observation Processing (CROP) to reduce the state information used for policy optimization. By providing only relevant information, overfitting to a specific training layout is precluded and generalization to unseen environments is improved. We formulate three CROPs that can be applied to fully observable observation- and action-spaces and provide methodical foundation. We empirically show the improvements of CROP in a distributionally shifted safety gridworld. We furthermore provide benchmark comparisons to full observability and data-augmentation in two different-sized procedurally generated mazes.

1 Introduction

To safely deploy *machine learning* (ML) methods in real-world scenarios, generalization is an important challenge. As training data cannot contain all possible situations in general, ML methods should be able to generalize to unseen samples instead of overfitting to the training data. More specifically, their learned behavior should be robust to scenarios not included in the training data, often also refereed to as out-of-distribution (OOD) generalization [Hendrycks *et al.*, 2021; Quinonero-Candela *et al.*, 2008]. Whilst important for supervised- and unsupervised-learning tasks, said generalization is especially important for the safe application of *reinforcement learning* (RL), where unexpected observations may cause unintended and potentially unsafe behavior [Amodei *et al.*, 2016]. For instance, a shift might be present when transferring a robotics model from the training simulation to the real-world [Zhao *et al.*, 2020]. Generally, increased generalization from a few examples, also referred to as frew-shot learning, is induced by altering the training data to increase its diversity and impede overfitting [Wang *et al.*, 2020]. Those *data augmentation* techniques have been successfully applied to both supervised and reinforcement learning by increasing the information used for training the model [Laskin *et al.*, 2020]. However, while the enlargement of the data may prevent overfitting and thus improve robustness to OOD samples, the increased complexity also increases training difficulty and hinders optimal convergence. On the contrary, despite tracing insufficient generalization to overfitting to the training data, occurrences of benign overfitting have been shown and characterized [Bartlett *et al.*, 2020]. Also, the states observed by the agent might not resemble an optimal representation, an "issues that are so often critical to successful applications" of reinforcement learning [Sutton and Barto, 2018].

Extending on these insights, we propose to apply *Compact Reshaped Observation Processing* (CROP) to reduce the observation such that presumably irrelevant details are removed and the remaining information is sufficient for learning a robust policy that generalizes to unseen shifted observations. Overall, we provide the following contribution:

- We formulate three concrete CROP methods applicable to fully observable state and action spaces, reducing the information with regards to the agent's position within the environment, the agent's action space and affect, and the objects within the environment (Figure 1).
- We provide proof-of-concept and show the strengths of each CROP over the full observation in a safety gridworld training environment and a shifted test environment.
- We evaluate the proposed CROPs and compare their effect in zero-shot generalization to the state-of the art of data augmentation in randomly generated mazes.

*Accepted for publication at IJCAI 2023

2 Background

2.1 Distributional Shifts

One key assumption in ML is that the data for training, testing and runtime is independent and identically distributed [Bishop and Nasrabadi, 2006]. Then, good performance in individual stages of the process would imply a similar performance in all other stages [Malinin *et al.*, 2021]. In practice, however, samples encountered during runtime may be out-of-distribution, e.g. due to sensors degrading over time [Hendrycks *et al.*, 2021]. *Distributional Shifts* describe a problem where all samples are out-of-distribution, i.e. the whole distribution shifts during runtime [Quinonero-Candela *et al.*, 2008]. This shift might be a slight unnoticeable change, or a significant alteration. Therefore, the key assumption of identically distributed data does not hold and performance may be impacted negatively. In safety-critical environments, this can cause severe issues, especially if the ML model makes unintended mistakes due to unexpected changes [Leike *et al.*, 2017]. This paper focuses on solutions referred to as *zero-shot generalization*, where, given limited amount of training data, generally applicable solutions for conceptually similar problems shall be inferred.

2.2 Problem Formulation

Fully observable decision making problems can be formulated as *Markov Decision Process (MDP)* $M_{MDP} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$, where $\mathcal{S}$ is a set of states s_t from a feature space $\mathcal{F}$, $\mathcal{A}$ is a set of actions a_t, $\mathcal{T}$ is the transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$ and $\mathcal{R}$ the reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ [Puterman, 1990]. Additionally, we consider a shifted set of states $\mathcal{S}^+ \neq \mathcal{S}$ that may be encountered alternatively.

The goal is to find a *policy* $\pi : \mathcal{S} \mapsto \Delta(\mathcal{A})$, which maximizes the value $V^\pi(s_t) = \mathbb{E}_\pi[G_t|s_t]$ for all $s_t \in \mathcal{S}$, where $G_t = \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)$ is the *return* and $\gamma = [0, 1]$ is the discount factor. An *optimal policy* π^* has the *optimal value function* $V^{\pi^*} = V^*$ satisfying $V^* \geq V^{\pi'}$ for all $s_t \in \mathcal{S}$ and π'.

MDPs can be extended to *Partially Observable Markov Decision Processes (POMDP)* $M_{POMDP} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O}, b_0 \rangle$, additionally consisting of a set Ω of observations o_t, observation function $\mathcal{O} : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\Omega)$, and initial state distribution $b_0 \rightarrow \Delta(\mathcal{S})$, where the agent does not perceive the true state s_t of the environment, but only a noisy observation $o_t \in \mathcal{O}$ according to $\mathcal{O}(o_t|s_t, a_{t-1})$ [Kaelbling *et al.*, 1998].

2.3 Reinforcement Learning

Reinforcement learning (RL) is an experience-based approach to find *optimal policies* π^* using experience tuples $e_t = \langle s_t, a_t, \mathcal{R}(s_t, a_t), s_{t+1} \rangle$.

Actor-critic methods are common RL algorithms using a function approximator $\hat{\pi}_\theta \approx \pi^*$ with learnable parameters θ, which are updated via gradient ascent according to gradient g [Sutton *et al.*, 2000]:

$$g = \hat{A}(s_t, a_t) \nabla_\theta log \hat{\pi}_\theta(a_t|s_t) \tag{1}$$

where $\hat{A}(s_t, a_t) = Q^{\hat{\pi}_\theta}(s_t, a_t) - V^{\hat{\pi}_\theta}(s_t)$ is the *advantage* and $Q^{\hat{\pi}_\theta}$ is the *action value function*. In practice, the return G_t is used to approximate $Q^{\hat{\pi}_\theta}(s_t, a_t)$ [Mnih *et al.*, 2016].

Proximal policy optimization (PPO) is a modified version of the standard *Advantage Actor-Critic* (A2C) method according to Eq. 1, which iteratively minimizes a surrogate loss function $\mathcal{L}_{PPO}$ to ensure stable learning [Schulman *et al.*, 2017]:

$$\mathcal{L}_{PPO}(\theta) = min(r_t \hat{A}(s_t, a_t), clip(r_t, 1 - \epsilon, 1 + \epsilon) \hat{A}(s_t, a_t)) \tag{2}$$

where $r_t = \frac{\hat{\pi}_\theta(a_t|s_t)}{\hat{\pi}_{\theta,old}(a_t|s_t)}$ is the importance sampling ratio of the current and old action probability with $\hat{\pi}_{\theta,old}$ being the policy originally collecting experience samples e_t for the update, and $\epsilon < 1$ is a clipping parameter, ensuring bounded changes θ to mitigate divergence.

3 Related Work

3.1 Robustness and Generalization in RL

Training robust RL agents that act reliably in unknown situations is a known challenge. One branch of research deals with the recognition of distributional shift [Ramanan *et al.*, 2021] and unknown states [Pimentel *et al.*, 2014; Thulasidasan *et al.*, 2021]. If distributional shift or unknown states are detected, one solution is to have the RL agent ask a (human) supervisor for advice [Najar and Chetouani, 2021] or to adjust the RL agent, e.g., through further training. In Adversarial RL [Pinto *et al.*, 2017], an opponent policy partially controls the agent during training, with the aim of minimizing the long-term reward. The result is a policy that is more robust to changes in the environment, as it anticipates intervention. Our approach does not detect distributional shift or include adversaries. Instead, it aims to increase robustness via zero-shot generalization.

Another branch of research aims to improve generalization by avoiding overfitting to the training data. Here, a variety of methods has been developed: stopping the training early [Raskutti *et al.*, 2014], dropping random parts of the underlying neural network [Srivastava *et al.*, 2014], or augmenting the training data with noise [Karystinos and Pados, 2000]. Furthermore, training can be carried out in as many different environments as possible [Tobin *et al.*, 2017; Cobbe *et al.*, 2020; Gisslén *et al.*, 2021], such that an RL agent can not succeed by overfitting to a small number of trajectories and is forced to acquire transferable knowledge. However, this requires a huge number of different environments, that are typically created by exhaustively generating variations of the same procedural environment. This is not sample efficient and makes it difficult to create distinct tests. However, the lack of distinct tests both diminishes the impact of distributional shift and the need for strong generalization to succeed. Thus, we aim to improve generalization by training with a reduced set of more relevant training data.

3.2 Data Augmentation in RL

Quantity and quality of the dataset heavily impact the training in ML [Ying, 2019]. *Data Augmentation* can be used to artificially increase the diversity in training data when only

a limited amount of data is available. The idea is to systematically modify the training data to avoid homogeneous structures [Shorten and Khoshgoftaar, 2019]. In image recognition tasks, this is done by geometric transformations such as mirroring, rotating or hiding pixels. This serves as a regularization against overfitting and increases data efficiency, ultimately improving generalization [Kostrikov *et al.*, 2020; Laskin *et al.*, 2020].

Various approaches adopted these ideas to RL [Raileanu *et al.*, 2020; Chen *et al.*, 2020; Yarats *et al.*, 2021]. Similar to our approach, *Reinforcement Learning with Augmented Data* (RAD) [Laskin *et al.*, 2020] augments observations without domain specific knowledge or changes to the RL algorithms. In addition to image based observations, RAD proposes two methods for state based observations: random amplitude scaling and Gaussian noise. By creating more diverse training data, the authors report increased data efficiency and better generalization to unseen environments.

We propose three additional methods to augment state based observations that reduce the amount of distinct training observations instead of increasing it. Improvements of the training are a welcome positive effect, but we focus on zero-shot generalization to unseen environments.

3.3 Partial Observability and Invariants

In real-world problems, RL agents often face incomplete and imperfect information [Choi *et al.*, 2019] and thus may perceive different states as similar [Spaan, 2012]. In such POMDPs, learning optimal policies with naive approaches is difficult, and the respective stochasticity is regarded as a fundamental challenge for RL [Vlassis *et al.*, 2012; Ghosh *et al.*, 2021; Jaakkola *et al.*, 1994]. However, we propose to train with limited state information on purpose, as this potentially mitigates the effect of directly perceiving the distributional shift and thus may improve robustness. Naturally, the remaining state information must be sufficient to find an optimal policy.

Removing potentially irrelevant information to improve the training is common in other ML areas such as facial recognition [Chen *et al.*, 2014]. While less common in RL, recent approaches followed this concept and proposed to explicitly learn invariants. [Zhang *et al.*, 2020] showed that agents can learn observation representations in latent space which encode task-relevant information. [Agarwal *et al.*, 2021] use a policy similarity metric (states are similar if the optimal policy has a similar behaviour in that and future states) with a contrastive learning approach to learn policies invariant to observation variations. [Mazoure *et al.*, 2021] use clustering methods and self-supervised learning to define an auxiliary task, which is mapping behaviorally similar states to similar representations. In fact, all these approaches require different observations from multiple training contexts and a complex nonlinear encoder that maps observations to a latent representation. On the contrary, we rely on a less complex handcrafted reduction of state information.

4 CROP

To facilitate generalization to mechanics underlying the environment, we propose to reshape observations to a compact format containing information with specific relevance to the agent. We argue, that if the reshaped state is invariant in similar situations, the policy optimization benefits from the more compact representation, while effects, previously described as benign overfitting [Bartlett *et al.*, 2020], can foster a policy that is robust regarding environmental changes.

Formally, we suggest utilizing a reshaping function $\mathrm{CROP} : \mathcal{S} \mapsto \mathcal{S}^*$, where $\mathcal{S}^*$ is the reshaped observation space, similar to the observation function $\mathcal{O}$ in POMDPs. In this paper, we use hard-coded compression functions, but we see great opportunities to extend this work to learning the compression functions as an approach to transfer learning in RL (meta-RL). By using the hard-coded compression function, we can leverage domain knowledge to accelerate learning. For interaction of the agent with the environment, CROP is used as surrogate observation, obtaining the modified observation $s_t^* = \mathrm{CROP}(s_t)$, where s_t is the d-dimensional full observation of the environment at step t. Algorithm 1 demonstrates the application of CROP to an arbitrary policy optimization algorithm.

Algorithm 1 CROPed Policy Optimization

Input: Initialized Policy π_θ
Parameters: An observation processing function `CROP`, A policy optimizer Θ and a learning rate λ

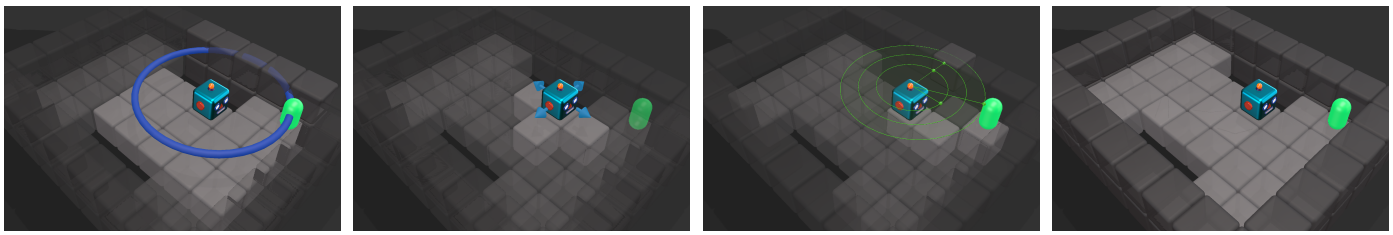
while not done **do** ▷ Collect Episode
  $\tau \leftarrow \emptyset$ ▷ Initialize empty rollout buffer
  for step **do** ▷ Perform Rollout
    $s_t^* = \texttt{CROP}(s_t)$ ▷ CROP Observation
    $a_t \sim \pi_\theta(a_t \mid s_t^*)$ ▷ Sample action
    $r_t = \mathcal{R}(s_t, a_t)$ ▷ Receive reward
    $s_{t+1} \sim \mathcal{T}(s_{t+1} \mid s_t, a_t)$ ▷ Execute action
    $\tau \leftarrow \tau \cup \{s_t, a_t, r_t, s_{t+1}\}$ ▷ Store transition
  end for
  $\theta \leftarrow \theta + \lambda \cdot \Theta(\tau)$ ▷ Update Policy
end while

To illustrate the proposed CROP mechanisms, we show their impact to an exemplary state in the fully observable safety training gridworld (cf. Figure 1d), that is introduced in full detail in section 5. However, it should be noted, that the proposed methods reflect basic concepts to asses the impact of CROP that may be refined, combined, or methodically applied to more complex observations. Concretely we propose the following three CROPs, visualized exemplary in Figure 1:

1. **Radius CROP** (cf. Figure 1a): Reshapes the observation to a ρ-sized radius around the agent:

$$CROP(s)_\rho^{Radius} = s_t[\alpha - \rho : \alpha + \rho] \tag{3}$$

State relevance: ensured by positional proximity
Required Information: the d-dimensional position of the agent α and a padding character to produce consistent-sized observations on the edges



(a) Radius CROP (b) Action CROP (c) Object CROP (d) Full Observation

Figure 1: **CROP**: *Compact Reshaped Observation Processing* based on the agent's position (blue / Figure 1a), action (light blue / Figure 1b) and surrounding objects (green / Figure 1c) in a fully observable (Figure 1d) safety gridworld environment rewarding the attainment of the target (green).

2. **Action CROP** (cf. Figure 1b): Reshapes the observation to states accessible via the immediate actions $\mathcal{A}$:

$$CROP(s_t)^{Action}_{\alpha,\mu} = (s_t[\alpha + \mu_0], \dots, s_t[\alpha + \mu_n]) \quad (4)$$

State relevance: ensured by state interactability and proximity, similar to Radius CROP, but sparser
Additional Information: the agent position α and a set of n action-mutations μ, assigning each possible action a d-dimensional mutation of the agent's position.

3. **Object CROP** (cf. Figure 1c): Reshapes the full observation (observation of every cell) to the distance vectors from the agent to the nearest η cells for each object type $O \subset \mathcal{F}$. Thus, the resulting observation will have the dimension $dim(s^*_t) = (|O| \cdot \eta, d)$

$$CROP(s_t)^{Object}_{O,\eta,\alpha,\sigma} = \\ (o_i - \alpha \quad \forall o \in O \forall o_i \in \sigma(s_t, \alpha, \eta, o)) \quad (5)$$

State relevance: ensured by object interactability and proximity, comparable to a LIDAR sensor
Additional Information: A scan mechanism $\sigma(s, \alpha, \eta, o)$ to find the absolute position of η-nearest cells containing object type o.

All proposed methods transform the observation into a relative state centered/based around the agent, in contrast to the absolute, global, full observation s_t. Radius CROP and Action CROP can be understood as explicit (hard encoded) attention mechanisms. While all proposed methods transform the given MDP into an POMDP, Radius CROP is the closest resemblance of partial observability in a classical sense within a gridworld scenario, that are typically centered around the agent, especially in arbitrary-sized, potentially infinite environments.

5 Experimental Setup

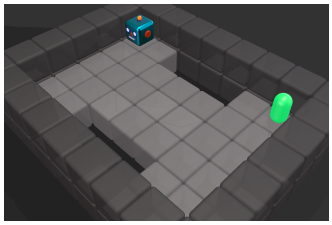
All implementations for the following evaluations can be found here [1].

Environments: To provide poof-of-concept for CROP we used two holey safety gridworlds inspired by [Leike *et al.*, 2017] comprising an $(7, 9)$ observation space with a set of five discrete features $\mathcal{F} = \{Wall, Field, Hole, Goal, Agent\}$ and and four discrete actions $\mathcal{A} = \{Up, Right, Down, Left\}$. To specifically asses the models' robustness to changes in the environment, we train all models in the single training configuration visualized in Figure 2a and test their performance in the unseen distributionally shifted environment shown in Figure 2b. For further evaluation and comparisons in section 7 we use $(7, 7)$- and $(11, 11)$-sized generated mazes inspired by [Cobbe *et al.*, 2020] with an identical action space and the reduced set of fields $\mathcal{F} = \{Wall, Field, Goal, Agent\}$ shown in Figure 2c and Figure 2d respectively. Again, to assess generalization, unseen configurations were used to test the trained policies. Therefore, we use a pool of 100 randomly generated mazes explicitly excluding the deterministic configuration to test polices trained in the single maze configurations and to train policies that are tested in the single deterministic environment. The reward range of randomly generated mazes is dependent on the shortest path, thus variable. However, the hardest possible mazes yielding the longest possible shortest paths, result in an optimal reward of 34 an 2 for Maze-7 and -11 respectively. To increase training speed, we trained all policies using four parallel environments.

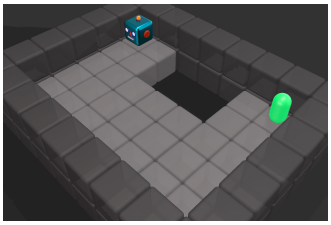
Policy Optimization: Whilst applicable to any policy optimization algorithm, we chose to evaluate CROP as an extension to PPO, having shown to be a robust and universally applicable state-of-the-art choice [Schulman *et al.*, 2017]. We furthermore built upon the implementations by [Raffin *et al.*, 2021], extending upon [Brockman *et al.*, 2016]. To analyze the effect of CROP and demonstrate its strengths we provide ablation studies and compare training and evaluation using *Full Observations* (FO) to all three CROPed observations in section 6. Furthermore we provide benchmark comparisons to a FO *Advantage Actor Critic* (A2C) from [Raffin *et al.*, 2021] and an alternative method for improved generalization, data augmentation, in section 7.

Data Augmentation (RAD): To provide a state-of-the-art comparison, we implemented data augmentation mechanisms for reinforcement learning (RAD) according to [Laskin *et al.*, 2020] and evaluate their impact in randomly generated mazes in section 7. For a fair comparison to CROP however, we also apply the proposed methods to the discrete full representation of the environment, instead of using images, and

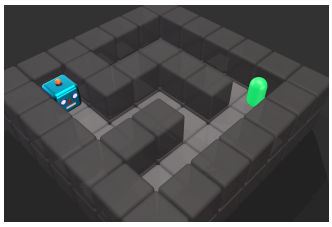
[1] https://github.com/philippaltmann/CROP



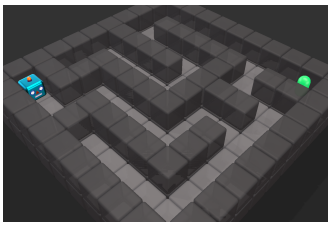
(a) Training $(-150, 42)$



(b) Test $(-150, 40)$



(c) Single Maze-7 $(-100, 42)$



(d) Single Maze-11 $(-100, 30)$

Figure 2: Evaluation environments and reward ranges: Holey distributional shift gridworlds inspired by [Leike *et al.*, 2017] for training (Figure 2a) and shifted evaluation (Figure 2b), as well as deterministic configurations of Maze 7 (Figure 2c) and Maze11 (Figure 2d) inspired by [Cobbe *et al.*, 2020]. The agents' goal is to reach the target (green capsule), rewarded with 50. To incentivize the shortest path, every step is penalized with -1. Holes immediately terminate the episode and are penalized with -50. Episodes are terminated after a maximum of 100 steps.

use PPO for training as suggested by the authors. Therefore, all image-based transformations like grayscale ore color-jitter are not applicable. Thus, we use random-crop, -translate, and -cutout with the same ratios suggested by [Laskin *et al.*, 2020], causing the full observation to be randomly cropped to (6,6) and (9,9), randomly translated to the original shape (7,7) and (11,11), and cut-out by patches sized in the ranges (0,2) and (0,3) in all dimensions.

Hyperparameters: For training PPO, we adopted the default parameters implemented by [Raffin *et al.*, 2021], also suggested in [Schulman *et al.*, 2017]. For the safety environments, we trained all models for a maximum of 1M steps, terminating once 99% of the optimal return (40 for the training environment) is reached. For **Radius CROP** we set the radius $\rho = (2, 2)$, resulting in an observation shape of $dim(s_t^*) = \rho \cdot 2 + 1 = (5, 5)$, padded with wall fields. Given the four possible actions $\mathcal{A} = \{Up, Right, Down, Left\}$ we parameterized **Action CROP** with $\mu = [(-1, 0), (0, 1), (1, 0), (0, -1)]$, resulting in an observation shape of $dim(s_t^*) = |\mathcal{A}| = (4)$. Regarding **Object CROP** we chose $\eta = 1$ for all safety environments and $\eta = 2$ for all mazes and the set of objects to be detected to be all possible objects excluding the agent itself: $O = \mathcal{F} \setminus \{Agent\}$, resulting in $O = \{Wall, Field, Hole, Goal\}$ and the observation shape $dim(s_t^*) = (4, 2)$ for the train and test environments (cf. Figure 2a and Figure 2b), as well as $O = \{Wall, Field, Goal\}$ and the observation shape $dim(s_t^*) = (3, 2)$ for all maze environments (cf. Figure 2c and Figure 2d).

Metrics: To reflect both the training performance and the generalization capabilities, we regularly (every 2^{13} steps) evaluated the policies during training (without further training) in both the training and an unseen test configuration, reflected by the metrics *Validation Return* and *Evaluation Return* respectively. The return is either averaged over 100 non-deterministic episodes (for all maze configurations) or based on a single deterministic episode (for all holey environments, to reflect the certain safety of the current policy). Additionally, to increase significance, all runs are averaged over eight independent seeds.

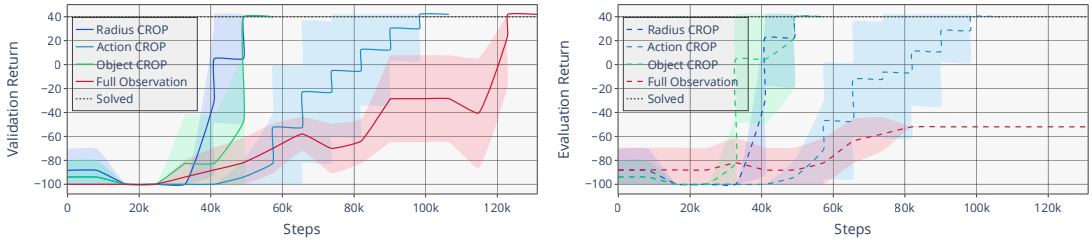
6 Evaluation

To provide proof of concept, that the reduced information is sufficient for learning an optimal policy, the following section contains ablation studies comparing the performance of PPO for learning a policy in the holey safety training environment (Figure 2a) using Full Observations (FO), Object CROP (O-CROP), Action CROP (A-CROP) and Radius CROP (R-CROP) (cf. Figure 1).

The progress of the Validation Return throughout training in the holey safety environment (Figure 2a) is shown in Figure 3a. Overall, all compared approaches find optimal policies, reaching returns above the solution threshold 40, within 150k steps. However, presumably caused by an insufficient representation, finding an optimal policy is the slowest using FOs. On the other hand, the results for all CROPs show, that the compressed information serves sufficient for training an optimal policy. Furthermore, caused by the increased relevance of observed states, the required training steps within the environment are reduced by 50% for R-CROP and O-CROP, compared to FO. The comparably slower training observed for A-CROP is probably caused by its too sparse observation, containing only the four neighboring states, which, again hinders fast convergence.

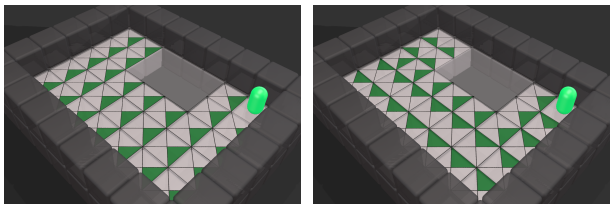
However, the real benefits of CROP are exposed in the Evaluation Return shown in Figure 3b. Caused by the previously unseen shift of the positions of the holes in the environments, polices trained using FO only reach returns up to -50. This return is most likely caused by polices that are terminated by a hole, instead of the agent reaching the target state. On the other hand, all CROP-trained polices show to be robust to said shift, resulting in significantly increased Evaluation Returns. Comparing the Evaluation Return with the Validation Returns in Figure 3a shows that said robustness is obtained in parallel to learning to solve the training environment, manifesting the advantages of CROP. Furthermore, all CROP trained polices reach Evaluation Returns above the solution threshold of the test environment of 38, resulting not only in a behavior that is able to navigate to the target state, but also uses the shortest possible path.

Figure 4a and Figure 4b provide further insights into the resulting policies, showing a heatmap visualization of the dominant (deterministic) action in each state of the unseen test environment, chosen by the FO- and R-CROP-trained policies respectively. Heatmaps depicting A-CROP and O-CROP policies are omitted, as their behavior resemble the results shown for R-CROP. As assumed above FO-trained policies



(a) Validation Return in Training Environment (b) Evaluation Return in Test Environment

Figure 3: **CROP Evaluation:** Comparing Radius- (blue), Action- (light blue), Object-CROP (green), and Full Observability (red) in the Distributional Shift Safety Environment. The number of steps taken in the environment is on the x-axis and the Validation Return (solid lines in Figure 3a), and Evaluation Return (dashed lines in Figure 3a) on the y-axis averaged over eight random seeds. The shaded areas mark the 95% confidence intervals. The reward threshold of 40 is displayed by the dotted line.



(a) FO Policy (b) R-CROP Policy

Figure 4: **Validation Heatmaps** visualizing the dominant action of the PPO-trained policies using Full Observations (Figure 4a) and Radius CROP (Figure 4b) for each possible state in the unseen shifted test environment (cf Figure 2b)

reveal a behavior directly navigating into the nearest hole, even though the policy has learned to evade the holes in the training environment (cf. Figure 4a). This behavior suggest that the trained policies overfit to the full observation of the training environment. The polices trained using R-CROP on the other hand are able to evade the shifted holes, even though, the environment has not been seen during training (cf. Figure 4b). Furthermore, the heatmap reveals, that the trained policy is able to reach the target within the shortest possible path, from any position, even though, the training was only conducted with the agent starting in the top left field. This generalization capability can not be observed for the FO-trained policies at all, only reaching the target from the neighboring field and otherwise failing to fulfill the intended task.

Overall, the evaluations results provide evidence that CROP reduces the information to an efficient representation containing the important information for finding an optimal policy, whilst accelerating training performance and improving robustness to distributional shifts by the removal of unimportant details, otherwise prone to overfitting.

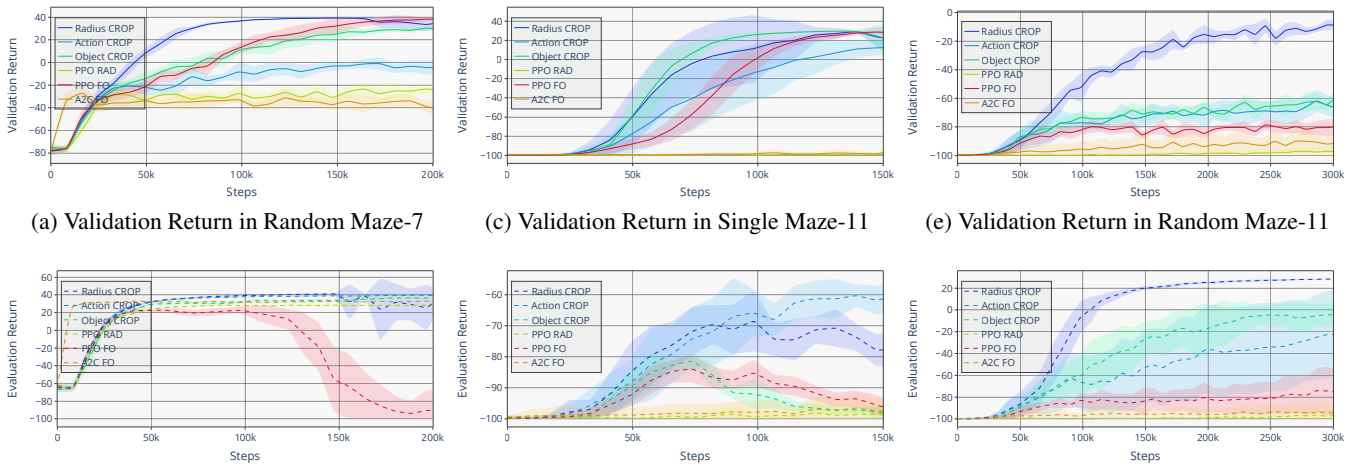
7 Benchmark Comparison

To further asses the generalization capabilities of CROP-trained policies, we provide comparisons to the Full Observation (FO) for training both PPO and A2C and Augmented Observations (RAD) for training PPO in three increasingly complex maze settings. As suggested in [Cobbe *et al.*, 2020], we first evaluate the generalization performance when training for 200k steps using a pool of randomly generated configurations.

The Validation Return is shown in Figure 5a. Similar to the previous evaluations, all observation types provide sufficient information for learning an expedient policy using PPO. However, both FO-A2C and RAD only reach performances below -20, while both FO-PPO and R-CROP reach near optimal performance of around 40. Again, presumably caused by the compressed observation, R-CROP shows the fastest convergence within about 100k steps, where FO and O-CROP increase the required steps by about 50%, A-CROP also provides sufficient information to reach the target, but its sparsity seems to hinder convergence to an optimal behavior.

However, evaluating the polices' generalization capabilities by analyzing their performance in the single unseen Maze-7 configuration, shown in Figure 5b, all CROP-trained polices show near optimal performance, with final Evaluation Returns of A-CROP, R-CROP, and O-CROP around 40. On the other hand, FO-trained policies initially show good generalization to the unseen test configuration within the first 100k steps. However, as the trained policy explores a near optimal solution in the training configuration, reaching Validation Returns upwards of 20, generalization drastically decreases and nearly drops to the minimum of -100 after 200k steps, indicating evaluation episodes where the final policy is not able to reach the target at all. This artifact of overfitting not showing for all CROP methods again confirms above assumptions regarding benign overfitting, thus improved generalization, encouraged by the improved observation. Interestingly, both FO-A2C and RAD-PPO show Evaluation Returns similar to CROP, for the single maze configuration, indicating either improved generalization, or, just less overfitting, given their Validation Return performed worst in comparison.

Increasing the maze size to 11 whilst relaxing the training complexity by using only a single configuration, similar effects show (cf. Figure 5c). All CROP-trained, as well as the FO-PPO trained policies reach near optimal performance within about 140k steps, while R-CROP and O-CROP show the fastest convergence. FO-A2C- and RAD-PPO-trained polices show the worst performance around the minimal re-



(a) Validation Return in Random Maze-7 (c) Validation Return in Single Maze-11 (e) Validation Return in Random Maze-11

(b) Evaluation Return in Single Maze-7 (d) Evaluation Return in Random Maze-11 (f) Evaluation Return in Single Maze-11

Figure 5: **Generalization Benchmark:** Comparing R-CROP (blue), A-CROP (light blue), O-CROP (green), RAD (yellow) and FO (red) for training PPO, and A2C using FO (orange) in 100 random Maze-7, a single Maze-11, and 100 random Maze-11 configurations for 200k, 150k, and 300k steps (x-axis) respectively, w.r.t. the Validation Return (solid lines in Figure 5a, Figure 5c, and Figure 5e) and the Evaluation Return (dashed lines in Figure 5b, Figure 5d, and Figure 5f) on the y-axis, averaged over eight random seeds. The shaded areas mark the 95% confidence intervals.

turn of -100. Interestingly however, even though showing slightly suboptimal training performance of around 15, A-CROP reaches the highest Evaluation Returns, thus best generalization. Note, that in this scenario, generalizing from a single training environment to 100 random unseen test environments is a notably harder challenge compared to the previous experiments, explaining the considerably lower overall Evaluation Returns. Nevertheless, while FO- and O-CROP-trained policies show indications of overfitting to the training environment in their Evaluation Return, R-CROP and A-CROP trained policies generalize to a behavior at least reaching the goal within unseen maze configurations. FO-A2C- and RAD-PPO-trained polices merely reach the target of the test environments at all, which is unsurprising given their training performance.

Finally, Figure 5e shows the Validation Return training in 100 random Maze-11 configurations. In contrast to the previous results, only Radius CROP-trained policies reach performances above -50, with the final return -10. Remarking the theoretical worst case maximum return for generated Maze-11 configurations of 2, however, this translates to an 90% optimal behavior. Furthermore, as for all previous experiments, the agent is not able to observe the target from the initial position using R-CROP, making this high performance even more remarkable. Moreover, this also translates to the Evaluation Return shown in Figure 5f, where a near optimal performance of around 20 is reached. Generally, due to their increased complexity, larger mazes have shown to be less prune to overfitting, especially when training with a pool of random configurations. However, still, CROP trained policies, especially using the proposed Radius method show the fastest training convergence and the best generalizing capabilities to unseen configurations.

8 Conclusion

Overall, we formalized a method for Compact Reshaped Observation Processing (CROP) and proposed three concrete CROPs applicable to fully observable discrete environments: Radius CROP, compressing the observation w.r.t. close positional proximity, Action CROP, compressing the observation w.r.t. interactability, and Object CROP, compressing the observation w.r.t. relations to surrounding objects. Furthermore, we showed the improvement using any of the proposed CROP over the full observation regarding both the training speed (steps until an optimal policy is found) and, more importantly, the robustness to a distributional shift in the environment in two holey safety environments. Finally, benchmark comparisons to the state-of-the-art using augmented data in procedural generated mazes further confirmed the advantages of CROPed observations, showing improved generalization to unseen maze configurations. Overall, Radius CROP has shown most beneficial, outperforming the full observation in all tested configurations.

Overall, we believe that CROPing observations to improve their information relevance is a promising approach for improving both the robustness and reliability of reinforcement learning algorithms.

Future work should therefore consider methods and applicability to further observation spaces. Also, further CROP methods could be developed with attention to different important features in the observation space. Furthermore, the proposed concepts could be applied for training a CROP mechanism for automatic state reduction. For example, they could serve as an inspiration for a target of the latent space of an autoencoder, that could be applied to CROP continuous or even partially observable observation spaces. Generally, various meta learning concepts may be applied to extend CROP for a more universal applicability.

References

[Agarwal *et al.*, 2021] Rishabh Agarwal, Marlos C Machado, Pablo Samuel Castro, and Marc G Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. *arXiv preprint arXiv:2101.05265*, 2021.

[Amodei *et al.*, 2016] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[Bartlett *et al.*, 2020] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

[Bishop and Nasrabadi, 2006] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[Brockman *et al.*, 2016] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[Chen *et al.*, 2014] Bor-Chun Chen, Chu-Song Chen, and Winston H Hsu. Cross-age reference coding for age-invariant face recognition and retrieval. In *European conference on computer vision*, pages 768–783. Springer, 2014.

[Chen *et al.*, 2020] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Gridmask data augmentation. *arXiv preprint arXiv:2001.04086*, 2020.

[Choi *et al.*, 2019] Jinyoung Choi, Kyungsik Park, Minsu Kim, and Sangok Seok. Deep reinforcement learning of navigation in a complex and crowded environment with a limited field of view. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5993–6000. IEEE, 2019.

[Cobbe *et al.*, 2020] Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pages 2048–2056. PMLR, 2020.

[Ghosh *et al.*, 2021] Dibya Ghosh, Jad Rahme, Aviral Kumar, Amy Zhang, Ryan P Adams, and Sergey Levine. Why generalization in rl is difficult: Epistemic pomdps and implicit partial observability. *Advances in Neural Information Processing Systems*, 34:25502–25515, 2021.

[Gisslén *et al.*, 2021] Linus Gisslén, Andy Eakins, Camilo Gordillo, Joakim Bergdahl, and Konrad Tollmar. Adversarial reinforcement learning for procedural content generation. In *2021 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2021.

[Hendrycks *et al.*, 2021] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021.

[Jaakkola *et al.*, 1994] Tommi Jaakkola, Satinder Singh, and Michael Jordan. Reinforcement learning algorithm for partially observable markov decision problems. *Advances in neural information processing systems*, 7, 1994.

[Kaelbling *et al.*, 1998] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[Karystinos and Pados, 2000] George N Karystinos and Dimitrios A Pados. On overfitting, generalization, and randomly expanded training sets. *IEEE Transactions on Neural Networks*, 11(5):1050–1057, 2000.

[Kostrikov *et al.*, 2020] Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

[Laskin *et al.*, 2020] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020.

[Leike *et al.*, 2017] Jan Leike, Miljan Martic, Victoria Krakovna, Pedro A Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. Ai safety gridworlds. *arXiv preprint arXiv:1711.09883*, 2017.

[Malinin *et al.*, 2021] Andrey Malinin, Neil Band, German Chesnokov, Yarin Gal, Mark JF Gales, Alexey Noskov, Andrey Ploskonosov, Liudmila Prokhorenkova, Ivan Provilkov, Vatsal Raina, et al. Shifts: A dataset of real distributional shift across multiple large-scale tasks. *arXiv preprint arXiv:2107.07455*, 2021.

[Mazoure *et al.*, 2021] Bogdan Mazoure, Ahmed M Ahmed, Patrick MacAlpine, R Devon Hjelm, and Andrey Kolobov. Cross-trajectory representation learning for zero-shot generalization in rl. *arXiv preprint arXiv:2106.02193*, 2021.

[Mnih *et al.*, 2016] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. In *International Conference on Machine Learning*, 2016.

[Najar and Chetouani, 2021] Anis Najar and Mohamed Chetouani. Reinforcement learning with human advice: a survey. *Frontiers in Robotics and AI*, 8:584075, 2021.

[Pimentel *et al.*, 2014] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal processing*, 99:215–249, 2014.

[Pinto *et al.*, 2017] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, pages 2817–2826. PMLR, 2017.

[Puterman, 1990] Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

[Quinonero-Candela *et al.*, 2008] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

[Raffin *et al.*, 2021] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[Raileanu *et al.*, 2020] Roberta Raileanu, Max Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in deep reinforcement learning. *arXiv preprint arXiv:2006.12862*, 2020.

[Ramanan *et al.*, 2021] Nandini Ramanan, Rasool Tahmasbi, Marjorie Sayer, Deokwoo Jung, Shalini Hemachandran, and Claudionor Nunes Coelho Jr. Real-time drift detection on time-series data. *arXiv preprint arXiv:2110.06383*, 2021.

[Raskutti *et al.*, 2014] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Early stopping and non-parametric regression: an optimal data-dependent stopping rule. *The Journal of Machine Learning Research*, 15(1):335–366, 2014.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Shorten and Khoshgoftaar, 2019] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[Spaan, 2012] Matthijs TJ Spaan. Partially observable markov decision processes. In *Reinforcement Learning*, pages 387–414. Springer, 2012.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Sutton *et al.*, 2000] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 1057–1063. MIT Press, 2000.

[Thulasidasan *et al.*, 2021] Sunil Thulasidasan, Sushil Thapa, Sayera Dhaubhadel, Gopinath Chennupati, Tanmoy Bhattacharya, and Jeff Bilmes. An effective baseline for robustness to distributional shift. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 278–285. IEEE, 2021.

[Tobin *et al.*, 2017] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[Vlassis *et al.*, 2012] Nikos Vlassis, Michael L Littman, and David Barber. On the computational complexity of stochastic controller optimization in pomdps. *ACM Transactions on Computation Theory (TOCT)*, 4(4):1–8, 2012.

[Wang *et al.*, 2020] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

[Yarats *et al.*, 2021] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021.

[Ying, 2019] Xue Ying. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168(2):022022, 2019.

[Zhang *et al.*, 2020] Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.

[Zhao *et al.*, 2020] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744. IEEE, 2020.