Contextual Imagined Goals for Self-Supervised Robotic Learning

Ashvin Nair[*] **Shikhar Bahl**[*] **Alexander Khazatsky**[*]
Vitchyr Pong **Glen Berseth** **Sergey Levine**
University of California, Berkeley

Abstract: While reinforcement learning provides an appealing formalism for learning individual skills, a general-purpose robotic system must be able to master an extensive repertoire of behaviors. Instead of learning a large collection of skills individually, can we instead enable a robot to propose and practice its own behaviors automatically, learning about the affordances and behaviors that it can perform in its environment, such that it can then repurpose this knowledge once a new task is commanded by the user? In this paper, we study this question in the context of self-supervised goal-conditioned reinforcement learning. A central challenge in this learning regime is the problem of goal setting: in order to practice useful skills, the robot must be able to autonomously set goals that are feasible but diverse. When the robot's environment and available objects vary, as they do in most open-world settings, the robot must propose to itself only those goals that it can accomplish in its present setting with the objects that are at hand. Previous work only studies self-supervised goal-conditioned RL in a single-environment setting, where goal proposals come from the robot's past experience or a generative model are sufficient. In more diverse settings, this frequently leads to impossible goals and, as we show experimentally, prevents effective learning. We propose a conditional goal-setting model that aims to propose goals that are feasible from the robot's current state. We demonstrate that this enables self-supervised goal-conditioned off-policy learning with raw image observations in the real world, enabling a robot to manipulate a variety of objects and generalize to new objects that were not seen during training.

Keywords: Robotics, Deep Reinforcement Learning, Self-Supervision

1 Introduction

In order for robots to truly become *generalists*, they must be readily taskable by humans, handle raw sensory inputs without instrumentation, and be equipped with a range of skills that generalize effectively to new situations. Reinforcement learning autonomously learns policies that maximize a reward function and is a promising approach towards such generalist robots. However, in a general setting involving diverse objects and tasks, prior information about what tasks to learn is hard to come by without manually designing object detectors and reward functions. How can a robot explore the world in order to learn and fine-tune useful skills on diverse objects, only from acting and observing how its actions affect its sensory stream?

Prior methods have proposed to let an agent learn from its sensor stream by automatically generating plausible goals during an unsupervised training phase, and then learning policies that reach those goals [1, 2, 3, 4]. Such goals can be defined in a variety of ways, but a simple choice is to use goal observations, such that each proposed task requires reaching a different observation. When the robot observes the world via raw camera images, this corresponds to using images as goals. At test time, a user then provides the robot with a new goal image.

While such methods have been demonstrated in both simulated and real-world settings, they are typically used to learn behaviors in domains with relatively little visual diversity. In the real world, a robot might interact with highly diverse scenes and objects, and the tasks that it can perform from each of the many possible initial states will be different. If the robot is presented with an object, it
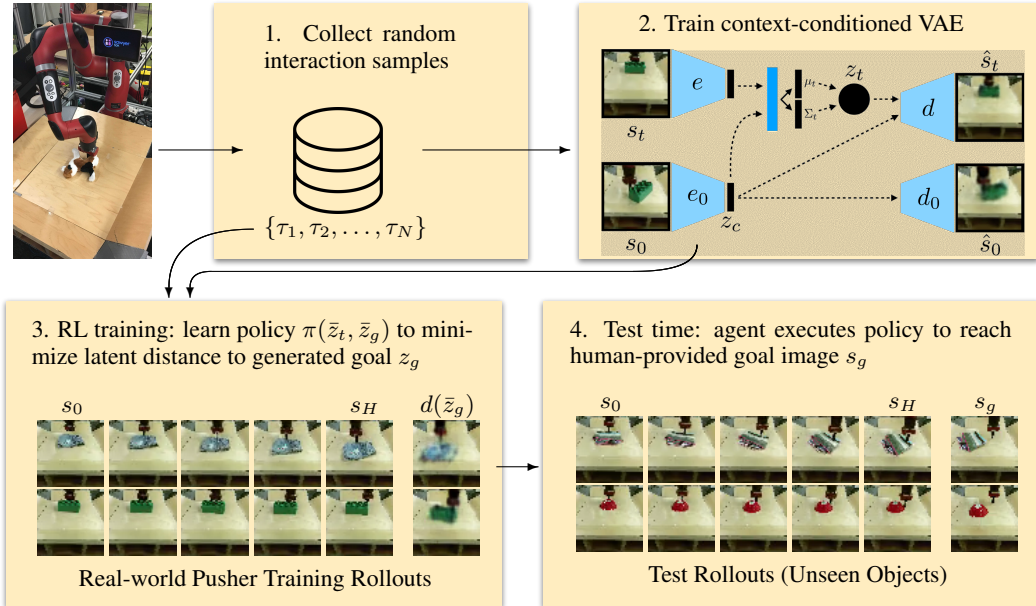
Figure 1: System overview of our self-supervised learning algorithm. (1) The agent collects random interaction data, to be used for both representation learning and as additional off-policy data for RL. (2) We propose a context-conditioned generative model (CC-VAE) to learn generalizable skills. In order to improve the generation of plausible goal states that result from a starting state s_0, our model allows information to flow freely through the context z_c while an information bottleneck on z_t gives us the ability to generate samples by re-sampling only z_t. This architecture provides a compact representation of the scene disentangling information that changes within a rollout (z_t) and information that changes between rollouts (z_c). We then use $\bar{z}_t = (z_t, z_c)$ as the representation for RL. (3) Our proposed CC-RIG algorithm samples latent goals, using the above representation, and learns a policy to minimize the latent distance to the goal with off-policy RL. Rollouts are shown on the real-world Sawyer robot pusher environment with visual variation. We include the initial image s_0, selected frames from the rollout, final image s_H, and the decoded goal latent $d(\bar{z}_g)$. (4) At test time, the agent is given a goal image s_g and executes the policy to reach it. Our method successfully handles pushing novel objects that were unseen at training time. Example rollouts can be found at https://ccrig.github.io/

can learn to pick up or grasp it, and when it is presented with a door, it can learn to open it. However, it must generate and practice goals that are suitable for each scene. In this paper, we propose and evaluate a self-supervised policy learning method that learns to propose goals that are suitable to the current scene via a conditional goal generation model, allowing it to learn in visually varied settings that prove challenging for prior algorithms.

The key idea in our work is that representing every element in a visually complex scene is often not necessary for control. A scene is a visual form of context that can be factored out, while only the controllable entities in the environment need to be captured for goal setting and representing the state. To this end, we propose learning a context-conditioned generative model that learns a smooth, compressed latent variable with an information bottleneck, while allowing the context, in the form of the initial state image, to be used freely to reconstruct other images during the task. This context-conditioned generative model architecture is shown in Figure 1.

The main contribution in this paper builds on this context-conditioned generative model to devise a complete self-supervised goal-conditioned reinforcement learning algorithm, which can handle visual variability in the scene via context-conditioned goal setting. Our method can learn policies that reach visually indicated goals without any additional supervision during training, using the context-conditioned generative model to set goals that are appropriate to the current scene. We show that our approach learns coherent representations of visually varied environments, capturing controllable dimensions of variation while ignoring dimensions that vary but cannot be influenced by the agent, such as lighting and object appearance. We further show that our approach can learn policies to solve tasks in visually varied environments, including in a real-world robotic pushing task with a wide variety of distinct objects in Figure 4.

2 Related Work

While many practical robots today perform tasks by executing hand-engineered sequences of motor commands, machine learning is opening up a new avenue to train a wide variety of robotic tasks from interaction. This body of work includes grasping [5, 6, 7], and general tasks [8, 9], multi-task learning [10], baseball [11], ping-pong [12], and various other tasks [13]. More recently, using expressive function approximators such as neural networks has reduced manual feature engineering and has increased task complexity and diversity, finding use in decision-making domains, such as solving Atari games [14] and Go [15]. Deep learning for robotics has proved to be difficult due to a host of challenges including noisy state estimation, specifying reward functions, and handling continuous action spaces, but has been used to investigate grasping [16], pushing [17], manipulation of 3D object models [18], active learning [19] and pouring liquids [20]. Deep reinforcement learning, which autonomously maximizes a given reward function, has been used to solve precise manipulation tasks [21], grasping [22, 23], door opening [24], and navigation [25]. These methods have succeeded on specific tasks, often with hard-coded reward functions. However, to scale task generalization robots may need to learn methods that can handle significant environment variation and require relatively little external supervision.

Several works have investigated self-supervised robotic interaction with varied objects in the deep learning setting with the goal of generalizing between objects. For example, in the domain of robotic grasping, several works have studied autonomous data collection to learn to grasp from a hand-specified grasping reward [16, 23]. However, hand-specifying such rewards in general settings and for arbitrary manipulation skills is very cumbersome. Other work has focused on self-supervised learning with visual forward models, either by enforcing a simplified dynamical structure [26, 27] or with pixel transformer architectures [28, 29, 30, 31, 32]. However, these methods rely on accurate visual forward modelling, which is itself a very challenging problem. Instead, we build on self-supervised model-free approaches, which allow the agent to efficiently reach visual goals without planning with a visual forward model.

Prior work has also sought to perform self-supervised learning with model-free approaches. Using visual inverse models [17] is one such approach, but may not work well for complex interaction dynamics or longer horizon planning. Most closely related to our approach are prior methods on goal-conditioned reinforcement learning [33, 34, 35]. The methods have been extended to frame self-supervised RL as learning goal reaching with automatically proposed goals, including visually-specified goals [1, 4, 3, 36, 37]. However, they generally focus on learning in narrow environments with little between-trial variability. In this setting, any previously visited state represents a valid goal. However, in the general case, this is no longer true: when the robot is presented with a different scene or different objects on each trial, it must only set those goals that can be accomplished in the current scene. In contrast, we focus on enabling self-supervised learning from off-policy data in heterogeneous environments with increased factors of variability.

3 Background

In this section, we provide an overview of relevant prior work necessary to understand our method, including goal-conditioned reinforcement learning and representation learning with variational auto-encoders.

3.1 Goal-Conditioned Reinforcement Learning

In an MDP consisting of states $s_t \in \mathcal{S}$, actions $a_t \in \mathcal{A}$, dynamics $p(s_{t+1}|s_t, a_t)$, rewards r_t, horizon H, and discount factor γ, reinforcement learning addresses optimizing a policy $a_t = \pi_\theta(s_t)$ to maximize expected return $\mathbb{E}[\sum_{t=0}^{H} \gamma^t r_t]$. To learn a variety of skills, it is instead convenient to optimize over a family of reward functions parametrized by goals, as in the framework of goal-conditioned RL [33]. A variety of RL algorithms exist, but as we are primarily interested in sample-efficient off-policy learning, we consider goal-conditioned Q-learning algorithms [34]. These algorithms learn a parametrized Q-function $Q_w(s, a, g)$ that estimates the expected return of taking action a from state s with goal g. Q-learning methods rely on minimizing the Bellman error:

$$\mathcal{L} = |Q_w(s, a, g) - (r + \max_{a'} Q_w(s', a', g))| \tag{1}$$

This objective can be optimized using standard actor-critic algorithms using a set of transitions (s, a, s', g, r) which can be collected off-policy [38]. In practice, a target network is often used for the second Q-function.

3.2 Variational Auto-Encoders

Above, the goal description can take many forms. To handle high-dimensional goals, in this work, we learn a latent representation of the state using a variational auto-encoder (VAE). A VAE is a probabilistic generative model that has been shown to learn structured representations of high-dimensional data [39] successfully. It is trained by reconstructing states s with a parametrized encoder that converts the state into a normal random distribution $q_\phi(z|s)$ with a parametrized decoder that converts the latent variable z into a predicted state distribution $p_\psi(s|z)$, while keeping the latent z close (in KL divergence) to its prior $p(z)$, a standard normal distribution. To train the encoder and decoder parameters ϕ and ψ, we jointly optimize both objectives when minimizing the negative evidence lower bound:

$$\mathcal{L}_{\text{VAE}} = -\mathbb{E}_{q_\phi(z|s)}[\log p(s|z)] + \beta D_{KL}(q_\phi(z|s)||p(z)). \tag{2}$$

3.3 Conditional Variational Auto-Encoders

Instead of a generative model that learns to generate the dataset distribution, one might instead desire a more structured generative model that can generate samples based on structured input. One example of this is a conditional variational auto-encoder (CVAE) that conditions the output on some input variable c and samples from $p(x|c)$ [40]. For example, to train a model that generates images of digits given the desired digit, the input variable c might be a one-hot encoded vector of the desired digit.

A CVAE trains $q_\phi(z|s, c)$ and $q_\psi(s|z, c)$, where both the encoder and decoder has access to the input variable c. The CVAE then minimizes:

$$\mathcal{L}_{\text{CVAE}} = -\mathbb{E}_{q_\phi(z|s,c)}[\log p(s|z, c)] + \beta D_{KL}(q_\phi(z|s, c)||p(z)). \tag{3}$$

Samples are generated by first sampling a latent $z \sim p(z)$. Based on c, we can then decode z with $q_\psi(s|z, c)$ and visualize the output, which is in our case an image. In our framework $c = s_0$.

3.4 Reinforcement Learning with Imagined Goals

To learn skills from raw observations in a self-supervised manner, reinforcement learning with imagined goals (RIG) proposed to use representation learning combined with goal-conditioned RL [1]. The aim of RIG is to choose actions in order to make the state s_t reach a goal image s_g at test time. RIG first collects an interaction dataset $\{s_t\}$ and learns a latent representation by training a VAE on this data. Then, a goal-conditioned policy is trained to act in the environment in order to reach a given goal z_g. Exploration data is collected by rolling out the policy with goals that are "imagined" from the VAE prior; at test time, the policy can take in a goal image as input, encode the image to the latent space, and act to reach the goal latent. A key limitation of this method is that sampling goals from the VAE prior during training time assumes that every state in the dataset is reachable at any time. However, in general, this assumption may not be true.

4 Self-Supervised Learning with Context-Conditioned Representations

In this work, our goal is to enable the learning of flexible goal-conditioned policies that can be used to successfully perform a variety of tasks in a variety of contexts – e.g., with different objects in the scene. Such policies must learn from large amounts of experience, and although it is in principle possible to use random exploration to collect this experience, this quickly becomes impractical in the real world. It is, therefore, necessary for the robot to set its own goals during self-supervised training, to collect meaningful experience. However, in diverse settings, many randomly generated goals may not be feasible – e.g., the robot cannot push a red puck if the red puck is not present in the scene. We propose to extend off-policy goal-conditioned reinforcement learning with a conditional goal setting model, which proposes only those goals that are currently feasible. This enables a learning regime with imagined goals that is more realistic for real-world robotic systems that must generalize effectively to a range of objects and settings.

4.1 Context-Conditioned VAEs

To train a generative model that can improve the generation of feasible goals in varied scenes, we use a modified CVAE that uses the initial state s_0 in a rollout as the input c, which we call the "context" for that rollout. The modified CVAE, which we call a context-conditioned VAE (CC-VAE), is shown in Figure 1. While most CVAE applications use a one-hot vector as the input, we use an image s_0. This image is encoded with a convolutional encoder e_0 into a compact representation z_c. Note that by design, e and e_0 do not share weights, as they are intended to encode different factors of variation in the images. The context z_c is used to output the latent representation z_t, as well as the reconstruction of the state $\hat{s}_t$. In addition, z_c is used alone to (deterministically) decode $\hat{s}_0 = d_0(z_c)$. The objective is given by

$$\mathcal{L}_{\text{CC-VAE}} = \mathcal{L}_{\text{CVAE}} + \log p(s_0|z_c). \tag{4}$$

Due to the information bottleneck on z_t, this loss function penalizes information passing through z_t but allows for unrestricted information flow from z_c. Therefore, the optimal solution would encode as much information as possible in z_c, while only including the state information that changes within a trajectory in the latent variable z_t. These are precisely the features of most interest for control.

4.2 Context-Conditioned Reinforcement Learning with Imagined Goals

We propose to use our context-conditioned VAE in the RIG framework to learn policies over environments with visual diversity, where each episode might involve interacting with a different scene and different objects. We first collect a dataset of trajectories $\mathcal{D} = \{\tau^{(i)}\}$ by executing random actions in the environment. We then learn a CC-VAE, as detailed in Section 4.1, to learn a factored representation of the image observations. To use the CC-VAE for self-supervised learning, we save the first image s_0 when starting a rollout. We compute the encoding of s_0, $z_c = e_0(s_0)$. Let $\bar{z}$ denote the context concatenated vector (z, z_c), and let $\mu(s, s_0)$ denote the mean of $q_\phi(z|s, s_0)$. We then use RIG in the $\bar{z}$ latent space by encoding observations with μ, meaning that we train a goal-conditioned policy $\pi(\bar{z}, \bar{z}_g)$ and a goal-conditioned Q-function $Q(\bar{z}, \bar{z}_g)$.

Algorithm 1 Context-Conditioned RIG

Require: Encoders $\mu(s_t, s_0)$, $e_0(s_0)$, policy $\pi_\theta(\bar{z}, \bar{z}_g)$, goal-conditioned value function $Q_w(\bar{z}, \bar{z}_g)$, dataset $\mathcal{D} = \{\tau^{(i)}\}$ of trajectories.
1: Train CC-VAE on $\mathcal{D}$ by optimizing (4).
2: **for** $n = 0, ..., N-1$ episodes **do**
3:     Sample latent goal $z_g \sim p(z)$, $\bar{z}_g = (z_g, z_c)$.
4:     Sample initial state $s_0 \sim p(s_0)$.
5:     Encode $z_c = e_0(s_0)$
6:     **for** $t = 0, ..., H-1$ steps **do**
7:         Observe s_t and encode $\bar{z}_t = (\mu(s_t, s_0), z_c)$
8:         Get action $a_t = \pi_\theta(\bar{z}_t, \bar{z}_g) +$ noise.
9:         Get next state $s_{t+1} \sim p(\cdot \mid s_t, a_t)$.
10:         Store $(\bar{z}_t, a_t, \bar{z}_{t+1}, \bar{z}_g)$ into replay buffer $\mathcal{R}$.
11:         Sample transition $(\bar{z}, a, \bar{z}', \bar{z}_g) \sim \mathcal{R}$.
12:         Compute new reward $r = -||\bar{z}' - \bar{z}_g||$.
13:         Minimize (1) using $(\bar{z}, a, \bar{z}', \bar{z}_g, r)$.
14:     **end for**
15: **end for**

To collect data, we sample a latent goal for each rollout from the prior $z_g \sim N(0, I)$, as in RIG. For every observation s_t, we compute the mean encoding $\mu_t(s_t, s_0)$. We then obtain a rollout of the policy by executing $\pi(\bar{z}, \bar{z}_g)$. The reward at each timestep is the latent distance $||\bar{\mu}_t - \bar{z}_g||$.

The policy and Q-function can be trained with any off-policy reinforcement learning algorithm. We use TD3 in our implementation [41]. Our policy and Q-function are goal-conditioned, and we take advantage of being able to relabel the goals for each transition to improve sample efficiency [35, 1, 4]. However, when relabeling a goal $\bar{z}_g$ with a random goal from the environment, the context-conditioning is still preserved. That is, if $z_g' \sim N(0, 1)$ is the new sampled goal, we use $\bar{z}_g' = (z_g', z_c)$. This ensures that the relabeled goal is compatible with the scene for the corresponding transition.

After training, we can use the learned policy π to reach a visually indicated goal. Given a goal image s_g, we encode it into a latent goal $z_g = \mu(s_g, s_0)$. Then, we execute the policy with the latent goal $\bar{z}_g$, just as during the training phase. The complete algorithm is presented in Algorithm 1.

5 Experiments

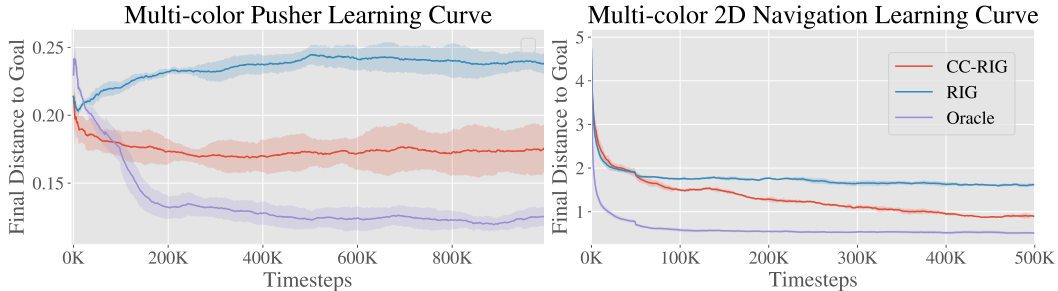In our experiments, we aim to answer the following questions:

Figure 2: Self-supervised learning results in visually varied simulated environments. CC-RIG significantly outperforms RIG and is competitive with the oracle method that has direct access to ground truth states.

1. How does our method compare to prior work at learning self-supervised skills in visually diverse environments?
2. Do context-conditioned VAEs learn an image representation that produces coherent and diverse goals that are suitable for the current scene?
3. Can our proposed context-conditioned RIG method handle diverse real-world data and learn effective policies under visual variation in the real world?

5.1 Self-Supervised Learning in Simulation

In simulation, we can conduct controlled experiments and evaluate against known underlying state values to measure the performance of our approach and prior methods. As a simulation test-bed, we use a multi-color pusher environment simulated in MuJoCo [42]. In this environment (Figure 3(left)), a simulated Sawyer arm is tasked with pushing a circular puck to a target position, specified by a goal image at test time. On each rollout, the puck color is set to a random RGB value. Therefore, the goal proposals for each method must adequately account for the color of the puck – a goal that requires moving a red puck to a given location is impossible if only a blue puck is present in the scene.

We compare the following algorithms: **CC-RIG.** Our method using a CC-VAE for representation learning, as described in Section 4.2. **RIG.** Reinforcement learning with imagined goals [1] using a standard VAE, as described in Section 3.4. **Oracle.** The oracle agent runs goal-conditioned RL with direct access to state information. Achieving performance similar to the oracle indicates that an algorithm loses little from using raw image observations over ground truth state.

Learning curves comparing these methods are presented in the plot on the left in Figure 2. CC-RIG outperforms RIG significantly, and standard RIG is not able to improve beyond the initial random policy. The performance of CC-RIG approaches that of the oracle policy, which has access to the true state. This suggests that, in visually varied environments, self-supervised learning is possible so long as the visual complexity is factored out with representation learning, and the proposed goals are consistent with the appearance of the current scene.

5.2 Generalizing to Varying Appearance and Dynamics with Self-Supervised Learning

In this experiment, to study changing both visual appearance and physical dynamics, we study how well our method can generalize when the environment dynamics change. We use a simulated 2D navigation task, where the goal is to navigate a point robot around an obstacle. The arrangement of the obstacles is chosen from a set of 15 possible configurations, and the color of the circle indicating the position of the point robot is generated from a random RGB value. Learning curves obtained by training the different methods above in this environment are presented in Figure 2. CC-RIG requires more samples to learn, but eventually approaches the oracle performance. RIG, in comparison, plateaus with poor performance in comparison.

5.3 Context-Conditioned VAE Goal Sampling

To better understand why CC-RIG outperforms RIG, we compare the samples from our CC-VAE to a standard VAE. Samples from both models are shown in Figure 3. The quality of the samples reveals why the CC-VAE provides better goal setting for self-supervised learning. In all environments, the
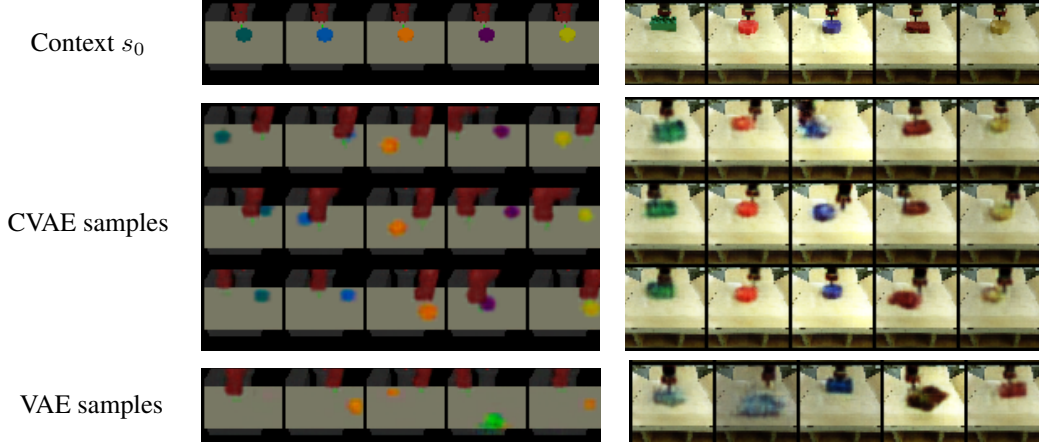
Figure 3: Comparing samples from our CC-VAE model to a standard VAE. The initial image s_0 is shown on the top row, and samples conditioned on s_0 are shown below. Our model coherently maintains object color and geometry in its samples, suggesting that the context conditioned model can successfully factor out the scene-specific object identity from the variable object position. This enables the use of the CC-VAE for goal proposals in visually diverse scenes.

samples from the CC-VAE maintain the background, object shape, and object color from the initial state. Therefore, the goals are more meaningful in the CC-VAE latent space.

This kind of visualization is a good indicator for the suitability of the representation for self-supervised learning. Diverse, coherent samples indicate that the latent space captures the appropriate factors of change in the environment and can be useful for self-supervised policy learning. Good samples also suggest that the latent space is well-structured, and therefore distances in the latent space should provide a good reward function for goal-reaching. In practice, we also look at the quality of the reconstructions. Good reconstructions confirm that the latent variables capture sufficient information about the image to be used in place of the image itself as a state representation.

5.4 Real-World Robotic Evaluation

In this experiment, we evaluate whether our method can handle manipulating visually varied objects in the real world. To this end, we used CC-RIG to train a Sawyer robot to manipulate a variety of objects, placed one at a time in a bin in front of the robot. As before, the training phase is self-supervised, and the robot must match a given goal image at test time.

We first collect a large dataset with random actions and train a CC-VAE on the data. Samples from the model are shown in Figure 4. The CC-VAE learns to generate goals with the correct object. To handle varying brightness at different times of the day, we added data augmentation by applying a color jitter filter to (s_0, s_t) pairs. As seen in the figure, the model is robust to this factor of variation. Each sample contains the same type of object, brightness level, and background as the initial state that it is conditioned on. However, crucially, these factors of variation are not present in z_t, as evidenced by the fact they do not vary within each column of Figure 4, but the object position does.

Next, we run CC-RIG with the trained CC-VAE to learn to reach visually indicated goals in a self-supervised manner. We first conduct fully off-policy training using the same dataset as was used to train the CC-VAE, and then collect a small amount of additional on-policy data to finetune the policy, analogously to recent work on large-scale vision-based robotic reinforcement learning [43]. The robot learns to push objects to target locations, indicated by a goal image. The real-world results are presented in Figure 4. Because it is difficult to automatically detect the positions of objects, we show some representative rollout examples, and we compute several distance metrics between the final state of a rollout and the goal: **CVAE distance.** CC-VAE latent space distance between final image and goal. **VAE distance.** VAE latent space distance between final image and goal. **Pixel distance.** We manually label the center of mass of the object in the final image and goal image, and compute the distance between them. **Object distance.** We measure the distance between the physical goal position of the object and the final position. In each metric, CC-RIG outperforms RIG.

Real-World Pushing Results

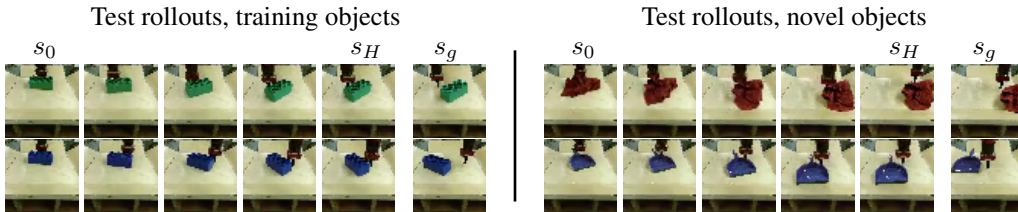|	CVAE distance	VAE distance	Pixel distance	Object distance (cm)
RIG	2.37 ± 0.97	2.41 ± 0.93	93.9 ± 41.7	17.1 ± 8.2
CC-RIG	1.66 ± 0.63	2.17 ± 0.88	56.8 ± 34.5	14.0 ± 6.9
CC-RIG, novel	1.51 ± 0.71	1.91 ± 0.87	53.1 ± 24.9	11.5 ± 2.9

Figure 4: The table above shows the performance of our method in the real-world, evaluated with four different evaluation metrics[1]. CC-RIG outperforms RIG in each one, even when tested on novel objects that it has not been trained on. Test rollouts of our method are shown on training objects on the left and unseen novel objects on the right. Successful rollouts where the object is pushed to the goal location are shown in top row, and failure modes are shown in the bottom row.

At training time, the dataset consists of interaction with 20 objects. The result of running CC-RIG on novel objects that were not included in the dataset are shown in the table as "CC-RIG, novel" and in the rollouts in Figure 4. These results show that our method can also generalize its experience to push novel objects it has not seen before.

6 Conclusion

We present a method for sample-efficient, flexible self-supervised task learning for environments with visual diversity. Our method can learn effective behavior without external supervision in simulated environments with randomized colors and layout, and in a real-world pushing task with differently colored pucks. Each environment contains an axis of visual variation that requires our algorithm to utilize an intelligent goal-setting strategy, to ensure that the self-proposed goals are consistent with the tasks and feasible in the current scene.

The main idea behind our method is to devise a context-conditioned goal proposal mechanism, allowing our self-supervised reinforcement learning algorithm to propose goals for itself that are feasible to reach. This context-conditioned VAE model factors out the unchanging context of a rollout, such as which objects are present in the scene, from the controllable aspects, such as the object positions to construct a more generalizable goal proposal model.

We believe this contribution will enable scalable learning in the real world. An agent manipulating objects in the real world must handle many forms of variation: different manipulation skills to learn, objects to manipulate, as well as variation in lighting, textures, etc. Methods that learn from data must be able to represent these variations while at the same time taking advantage of common structure across objects and tasks in order to achieve practical sample efficiency. Future work will address the remaining challenges to achieve this vision. This includes minimizing repeated human supervision even further by allowing reset-free data collection.

---

[1]The first three metrics are computed on 40 trajectories per method, and we report mean $\pm$ standard deviation. Object distance is computed on 10 trajectories per method, and we report median $\pm$ standard deviation.

References

[1] A. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine, "Visual Reinforcement Learning with Imagined Goals," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[2] O. Nachum, G. Brain, S. Gu, H. Lee, and S. Levine, "Data-Efficient Hierarchical Reinforcement Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[3] D. Warde-Farley, T. Van De Wiele, T. Kulkarni, C. Ionescu, S. Hansen, &. Volodymyr, and M. Deepmind, "Unsupervised Control Through Non-Parametric Discriminative Rewards," in *International Conference on Learning Representations (ICLR)*, 2019.

[4] V. H. Pong, M. Dalal, S. Lin, A. Nair, S. Bahl, and S. Levine, "Skew-fit: State-covering self-supervised reinforcement learning," *CoRR*, vol. abs/1903.03698, 2019.

[5] S. Ekvall and D. Kragic, "Interactive grasp learning based on human demonstration," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 4.   IEEE, 2004, pp. 3519–3524.

[6] O. Kroemer, R. Detry, J. Piater, and J. Peters, "Combining Active Learning and Reactive Control for Robot Grasping," *Robotics and Autonomous systems*, vol. 58, no. 9, pp. 1105–1116, 2010.

[7] J. Bohg and D. Kragic, "Learning grasping points with shape context," *Robotics and Autonomous Systems*, vol. 58, no. 4, pp. 362–377, 2010.

[8] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural Networks*, vol. 21, no. 4, pp. 682 – 697, 2008, robotics and Neuroscience.

[9] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[10] M. P. Deisenroth, P. Englert, J. Peters, and D. Fox, "Multi-task policy search for robotics," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 3876–3881.

[11] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural Networks*, vol. 21, no. 4, pp. 682–697, 2008.

[12] J. Peters, K. Mülling, and Y. Altün, "Relative Entropy Policy Search," in *AAAI Conference on Artificial Intelligence*, 2010, pp. 1607–1612.

[13] M. P. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *International Conference on Machine Learning (ICML)*, 2011, pp. 465–472.

[14] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," in *NIPS Workshop on Deep Learning*, 2013, pp. 1–9.

[15] D. Silver, A. Huang, C. J. Maddison *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, jan 2016.

[16] L. Pinto and A. Gupta, "Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours," *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[17] P. Agrawal, A. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to Poke by Poking: Experiential Learning of Intuitive Physics," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[18] M. Krainin, B. Curless, and D. Fox, "Autonomous generation of complete 3d object models using next best view manipulation planning," in *2011 IEEE International Conference on Robotics and Automation*.   IEEE, 2011, pp. 5031–5037.

[19] D. Martinez, G. Alenya, P. Jimenez, C. Torras, J. Rossmann, N. Wantia, E. E. Aksoy, S. Haller, and J. Piater, "Active learning of manipulation sequences," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*.   IEEE, 2014, pp. 5671–5678.

[20] C. Schenck and D. Fox, "Visual closed-loop control for pouring liquids," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*.   IEEE, 2017, pp. 2629–2636.

[21] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-End Training of Deep Visuomotor Policies," *Journal of Machine Learning Research (JMLR)*, vol. 17, no. 1, pp. 1334–1373, 2016.

[22] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust Adversarial Reinforcement Learning," in *International Conference on Machine Learning (ICML)*, 2017.

[23] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection," *International Journal of Robotics Research*, 2017.

[24] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous Deep Q-Learning with Model-based Acceleration," in *International Conference on Machine Learning (ICML)*, 2016.

[25] G. Kahn, A. Villaflor, B. Ding, P. Abbeel, S. Levine, and B. A. Research, "Self-supervised Deep Reinforcement Learning with Generalized Computation Graphs for Robot Navigation," in *International Conference on Robotics and Automation*, 2018.

[26] M. Watter, J. T. Springenberg, J. Boedecker, and M. Riedmiller, "Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images," in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 2728–2736.

[27] M. Zhang, S. Vikram, L. Smith, P. Abbeel, M. J. Johnson, and S. Levine, "SOLAR: Deep Structured Representations for Model-Based Reinforcement Learning," in *International Conference on Machine Learning (ICML)*, aug 2019.

[28] C. Finn and S. Levine, "Deep Visual Foresight for Planning Robot Motion," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[29] F. Ebert, C. Finn, A. X. Lee, and S. Levine, "Self-Supervised Visual Planning with Temporal Skip Connections," in *Conference on Robot Learning (CoRL)*, 2017.

[30] F. Ebert, S. Dasari, A. X. Lee, S. Levine, and C. Finn, "Robustness via Retrying: Closed-Loop Robotic Manipulation with Self-Supervised Learning," in *Conference on Robot Learning (CoRL)*, 2018.

[31] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic Adversarial Video Prediction," 2018.

[32] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual Foresight: Model-Based Deep Reinforcement Learning for Vision-Based Robotic Control."

[33] L. P. Kaelbling, "Learning to achieve goals," in *International Joint Conference on Artificial Intelligence (IJCAI)*, vol. vol.2, 1993, pp. 1094 – 8.

[34] T. Schaul, D. Horgan, K. Gregor, and D. Silver, "Universal Value Function Approximators," in *International Conference on Machine Learning (ICML)*, 2015, pp. 1312–1320.

[35] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. Mcgrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight Experience Replay," in *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[36] C. Florensa, J. Degrave, N. Heess, J. T. Springenberg, and M. Riedmiller, "Self-supervised learning of image embedding for continuous control," *arXiv preprint arXiv:1901.00943*, 2019.

[37] X. Lin, H. Singh Baweja, and D. Held, "Reinforcement Learning without Ground-Truth State," *arXiv preprint arXiv:1905.07866*, 2019.

[38] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2016.

[39] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[40] K. Sohn, X. Yan, and H. Lee, "Learning Structured Output Representation using Deep Conditional Generative Models," in *Advances in Neural Information Processing Systems*, 2015.

[41] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," *International Conference on Machine Learning (ICML)*, 2018.

[42] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 5026–5033.

[43] D. Kalashnikov, A. Irpan, P. Pastor *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Proceedings of The 2nd Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 87.   PMLR, 29–31 Oct 2018, pp. 651–673.