Contrastive Unsupervised Learning of World Model with Invariant Causal Features

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper we present a *world model*, which learns causal features using the invariance principle. In particular, we use contrastive unsupervised learning to learn the invariant causal features, which enforces invariance across augmentations of irrelevant parts or styles of the observation. The world-model-based reinforcement learning methods independently optimize representation learning and the policy. Thus naïve contrastive loss implementation collapses due to a lack of supervisory signals to the representation learning module. We propose an intervention invariant auxiliary task to mitigate this issue. Specifically, we utilize depth prediction to explicitly enforce the invariance and use data augmentation as style intervention on the RGB observation space. Our design leverages unsupervised representation learning to learn the world model with invariant causal features. Our proposed method significantly outperforms current state-of-the-art model-based and model-free reinforcement learning methods on out-of-distribution point navigation tasks on the iGibson dataset. Moreover, our proposed model excels at the sim-to-real transfer of our perception learning module.

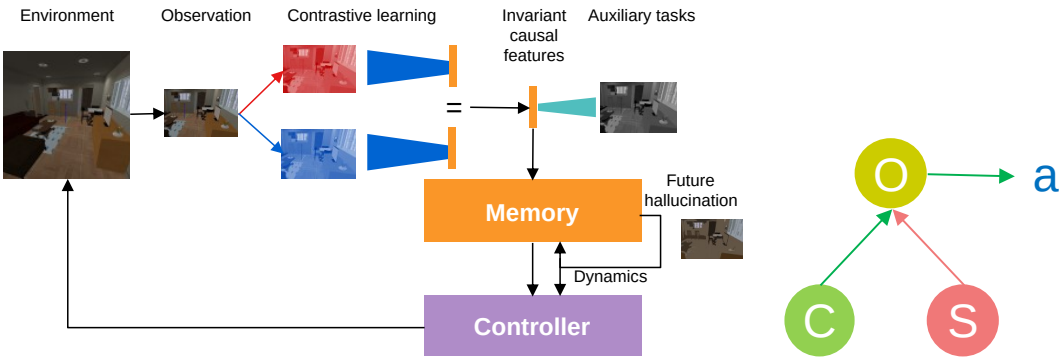
1 Introduction

An important branching point in reinforcement learning (RL) methods is whether the agent learns with or without a predictive environment model. In model-based methods, an explicit predictive model of the world is learned, enabling the agent to plan by thinking ahead [1–3]. The alternative model-free methods do not learn the predictive model of the environment explicitly as the control policy is learned end-to-end. As a consequence, model-free methods learn a greedy state representation for the task at hand and do not consider future downstream tasks. Therefore, we consider model-based methods more attractive for continuous learning, out-of-distribution (OoD) generalization and sim-to-real transfer.

A model-based approach has to learn the model of the environment purely from experience, which poses several challenges. The main problem is the training bias in the model, which can be exploited by an agent and lead to poor performance during testing [2]. Further, model-based RL methods learn the representation using observation reconstruction loss, for example variational autoencoders (VAE) [4]. The downside of such a state abstraction method is that it is not suited to separate relevant states from irrelevant ones, resulting in performance degradation of the control policy, even for slight task-irrelevant style changes. Hence, relevant state abstraction is essential for robust policy learning, which is the aim of this paper.

Causality is the study of learning cause and effect relationships. Learning causality in pixel-based control involves two tasks. The first is a causal variable abstraction from images, and the second is learning the causal structure. Causal inference uses graphical modelling [5], structural equation modelling [6], or counterfactuals [7]. Pearl [8] provided an excellent overview of those methods.

Submitted to the NeurIPS Causal ML for Real-World Impact Workshop (CML4Impact 2022). Do not distribute.



(a) Flow diagram of WMC. (b) Causal graph of WMC.

Figure 1: Components of the proposed *World Model with invariant Causal features* (WMC).

However, in complex visual control tasks the number of state variables involved is high, so inference of the underlining causal structure of the model becomes intractable. Causal discovery using the invariance principle tries to overcome this issue and is therefore gaining attention in the literature [9–12]. Arjovsky et al. [10] learns robust classifiers based on invariant causal associations between variables from different environments. Zhang et al. [11] uses multiple environments to learn the invariant causal features using a common encoder. Here spurious or irrelevant features are learnt using environment specific encoders. However, these methods need multiple sources of environments with specific interventions or variations. In contrast, we propose using data augmentation as a source of intervention, where samples can come from as little as a single environment, and we use contrastive learning for invariant feature abstraction. Related to our work, Mitrovic et al. [12] proposed a regularizer for self-supervised contrastive learning. On the other hand, we propose an intervention invariant auxiliary task for robust feature learning.

Model-based RL methods do not optimize feature learning and control policy end-to-end to prevent greedy feature learning. The aim is that such features will be more useful for various downstream tasks. Hence, state abstraction uses reward prediction, reconstruction loss or both [2, 3, 11]. On the other hand contrastive learning does not use the reconstruction of the inputs and applies the loss at the embedding space. Hence, we propose a causally invariant auxiliary task for invariant causal features learning. Specifically, we utilize depth predictions to extract the geometrical features needed for navigation, which are not dependent on the texture. Further, most computer vision data augmentations on RGB image space are invariant to depth. Finally, we emphasize that depth is not required for deployment, enabling wider applicability of the proposed model. Importantly, our setup allows us to use popular contrastive learning on model-based RL methods and improves the sample efficiency and the OoD generalization.

In summary, we propose a *World Model with invariant Causal features* (WMC), which can extract and predict the causal features (Figure 1a). Our WMC is verified on the *point goal* navigation task from Gibson [13] and iGibson 1.0 [14]. Our main contributions are:

1. Show that the world model can benefit from contrastive unsupervised representation learning.
2. Propose a world model with invariant causal features, which outperforms state-of-the-art RL models on out-of-distribution generalization and sim-to-real transfer of learned features.

2 Proposed Model

The data flow diagram of our proposed world model is shown in Figure 1a. We consider the visual control task as a finite-horizon partially observable Markov decision process (POMDP). We denote observation space, action space and time horizon as $\mathcal{O}$, $\mathcal{A}$ and $\mathcal{T}$ respectively. An agent performs continuous actions $a_t \sim p(a_t|o_{\leq t}, a_{<t})$, and receives observations and scalar rewards $o_t, r_t \sim p(o_t, r_t|o_{<t}, a_{<t})$ from the unknown environment. We use the deep neural networks for the state abstractions from observation $s_t \sim p(s_t|s_{t-1}, a_{t-1}, o_t)$, predictive transition model

$s_t \sim q(s_t|s_{t-1}, a_{t-1})$ and reward model $r_t \sim q(r_t|s_t)$. The goal of an agent is to maximize the expected total rewards $E_p(\sum_{t=1}^{T} r_t)$. In the following sections we describe our proposed model.

2.1 Invariant Causal Features Learning

Learning the pixel-based controller involves two main tasks, first abstraction of the environment state and second maximizing the expected total reward of the policy. Since visual control is a complex task, the number of causal variables involves are high, making the causal structure discovery a difficult task as it requires fitting the graphical model or structural equation. That is why we choose invariant prediction as our method of choice for causal feature learning. The key idea is that if we consider all *causes* of an *effect*, then the conditional distribution of the *effect* given the *causes* will not change when we change all the other remaining variables of the system [9]. Such experimental changes are known as interventions in the causality literature. We have used data augmentation as our mechanism of intervention, since we do not have access to the causal or spurious variables or both of the environment. We have also explored texture randomization as an intervention in our experiments, which we call action replay.

We have shown the high-level idea of the proposed causal features extraction technique in Figure 1b. The main idea is observation is made of content (C), causal variables, and style (S), spurious variables. We want our representation learning to extract the content variables only, which are the true cause of the action. In other words we want our control policy to learn $P(a|c = invariant_encoder(o))$ but not $P(a|(c, s) = encoder(o))$ as causal variables are sample efficient, and robust to OoD generalization and sim-to-real transfer. We have chosen the contrastive learning technique [15, 16] to learn the invariant causal features, which means embedding features of a sample and its data augmented version, intervention on style or spurious variables, should be the same. We use spatial jitter (crop and translation), Gaussian blur, color jitter, grayscale and cutout data augmentation techniques [15, 17, 18] for contrastive learning. Since the intervention of the style is performed at the image level, we do not need to know the location of the interventions. Further, the data also can come from observation of the different environments with different environment-level interventions. The theoretical guarantee of causal inference using invariant prediction is discussed in Peters et al. [9] and Mitrovic et al. [12]. However, our proposed method does not consider the hidden confounding variables that influence the target effect variable.

2.2 World Model

The proposed *World Model with invariant Causal features* (WMC) consists of three main components: i) unsupervised causal representation learning, ii) memory, also know as the world model, and iii) controller. The representation learning module uses the contrastive learning for invariant causal state learning. The memory module uses a recurrent neural network. It outputs the parameters of a categorical distribution, discrete and multinomial distribution, which is used to sample the future states of the model. Finally, the controller maximizes the action probability using an actor critic approach [3, 19]. We have adopted DreamerV2 [3] to test our proposed hypothesis, which we describe below.

Unsupervised causal features learning. Invariant causal feature extraction from the RGB image observation is a key component of our model. As described previously, we learn invariant causal features by maximizing agreement between different style interventions of the same observation via a contrastive loss in the latent feature space. Since the world model optimizes feature learning and controller separately to learn better representative features for downstream tasks, our early result with only rewards prediction was poor, which we verified in our experiments. Another reason for separate training of the world model and controller is that most of the complexity resides in the world model (features extraction and memory), so that controller training with RL methods will be easier. Hence, we need a stronger loss function to learn a good state representation of the environment. That is why we propose depth reconstruction as an auxiliary task and do not use image resize data augmentation to keep the relation of object size and distance intact. We used InfoNCE [20] style loss to learn the invariant causal feature. Hence, our encoder takes RGB observation and task specific information as inputs and depth reconstruction and reward prediction as targets. The invariant state abstraction is enforced by contrastive loss. The proposed invariant causal features learning technique has the following three major components:

- A *style intervention* module that uses data augmentation techniques. We use spatial jitter, Gaussian blur, color jitter, grayscale and cutout data augmentation techniques for style intervention. Spatial jitter is implemented by first padding and then performing random crop. Given any observation o_t, our style intervention module randomly transforms in two correlated views of the same observations. All the hyperparameters are provided in the appendix.
- We use an *encoder* network that extracts representation from augmented observations. We follow the same configurations as DreamerV2 for a fair comparison, and only contrastive loss and depth reconstruction tasks are added. We obtain $\tilde{s_t} = invariant_encoder(o_t)$, then we follow the DreamerV2 to obtain the final state $s_t \sim p(s_t|s_{t-1}, a_{t-1}, \tilde{s_t})$. We use the contrastive loss immediately after the encoder $\tilde{s_t}$.
- *Contrastive loss* is defined for a contrastive prediction task, which can be explained as a differentiable dictionary lookup task. Given a query observation q and a set $K = \{k_0, k_1, ...k_{2B}\}$ with known positive $\{k_+\}$ and negative $\{k_-\}$ keys. The aim of contrastive loss is to learn a representation in which positive sample pairs stay close to each other while negative ones are far apart. In contrastive learning literature q, K, k_+ and k_- are also referred as *anchors*, *targets*, *positive* and *negative* samples. We use bilinear products for *projection head* and InfoNCE loss for contrastive learning [20], which enforces the desired similarity in the embedding space:

$$\ell_t^q = \log \frac{\exp(q^T W k_+)}{\exp(q^T W k_+) + \sum_{i=0}^{2(B-1)} \exp(q^T W k_i)} \tag{1}$$

  This loss function can be seen as the log-loss of a $2B$-way softmax classifier whose label is k_+. Where B is a batch size, which becomes $2B$ after the style intervention module randomly transforms in two correlated views of the same observation. The quality of features using contrastive loss depends on the quality of the negative sample mining, which is a difficult task in an unsupervised setting. We slice the sample observations from an episode at each 5th time-step to reduce the similarity between neighbouring negative observations. In summary, causal feature learning has the following component and is optimized by Equation 1.

$$\text{Invariant causal model:} \quad p_\theta(\tilde{s_t}|o_t) \tag{2}$$

Future predictive memory model. The representation learning model extracts what the agent sees at each time frame but we also want our agent to remember the important events from the past. This is achieved with the memory model and implemented with a recurrent neural network. Further, the transition model learns to predict the future state using the current state and action in the latent space only, which enables future imagination without knowing the future observation since we can obtain the future action from the policy if we know the future state. Hence, this module is called a future predictive model and enables efficient latent imagination for planning [2, 21]. In summary, dynamic memory and representation learning modules are tightly integrated and have the following components,

$$\begin{aligned} &\text{Representation model:} && p_\theta(s_t|s_{t-1}, a_{t-1}, \tilde{s_t}) \\ &\text{Depth prediction model:} && q_\theta(o_t^d|s_t) \\ &\text{Reward model:} && q_\theta(r_t|s_t) \\ &\text{Predictive memory model:} && q_\theta(s_t|s_{t-1}, a_{t-1}). \end{aligned} \tag{3}$$

All the world model and representation losses were optimized jointly, which includes contrastive, depth prediction, reward and future predictive KL regularizer losses respectively,

$$\mathcal{L}_{WM} = E_p \left(\sum_t \left(\ell_t^q + \ln q(o_t^d|s_t) + \ln q(r_t|s_t) - \beta \ell_t^{KL} \right) \right) \tag{4}$$

where, $\ell_t^{KL} = KL(p(s_t|s_{t-1}, a_{t-1}, \tilde{s_t})||q(s_t|s_{t-1}, a_{t-1}))$

Table 1: Experiment results on PointGoal navigation task from iGibson 1.0 dataset.

		Ihlen_0_int		Ihlen_1_int		Rs_int		**Total**	
Models	Steps	SR	SPL	SR	SPL	SR	SPL	SR	SPL
RAD	100k	0.5	0.01	0.1	0.00	0.8	0.01	0.5	0.01
CURL	100k	8.0	0.07	0.6	0.00	5.4	0.05	4.6	0.04
DreamerV2	100k	1.7	0.01	0.5	0.00	1.6	0.01	1.3	0.01
DreamerV2 + DA	100k	7.2	0.05	1.5	0.01	7.7	0.05	5.5	0.03
DreamerV2 - I + D	100k	1.9	0.01	0.9	0.00	2.8	0.01	1.9	0.01
DreamerV2 - I + D + DA	100k	8.3	0.05	2.1	0.01	10.8	0.07	7.0	0.04
WMC	100k	**28.9**	**0.22**	**7.9**	**0.05**	**30.2**	**0.22**	**22.3**	**0.16**
WMC - AR	100k	15.4	0.11	4.2	0.02	19.7	0.12	13.1	0.08
WMC - AR - D	100k	0.0	0.0	0.0	0.0	0.0	0.0	0.03	0.00
WMC - D + I	100k	15.4	0.11	4.6	0.03	17.0	0.12	12.3	0.09
RAD	500k	48.7	**0.44**	11.6	**0.11**	48.5	0.44	36.3	0.32
CURL	500k	40.8	0.36	11.4	0.09	41.9	0.36	31.4	0.27
DreamerV2	500k	1.3	0.01	0.8	0.00	2.2	0.01	1.4	0.01
DreamerV2 + DA	500k	7.2	0.04	1.1	0.01	9.7	0.05	13.0	0.08
DreamerV2 - I + D	500k	3.3	0.02	0.9	0.00	4.1	0.02	2.7	0.01
Dreamer - I + D + DA	500k	38.0	0.25	9.0	0.06	52.7	0.35	33.2	0.22
WMC	500k	**58.6**	**0.44**	**15.7**	**0.11**	**67.4**	**0.51**	**47.26**	**0.36**
WMC - AR	500k	45.1	0.28	11.5	0.07	26.8	0.18	27.8	0.17
WMC - AR - D	500k	0.9	0.0	0.2	0	1.2	0.01	0.7	0.00
WMC - D + I	500k	28.6	0.12	6.6	0.04	22.3	0.14	19.1	0.10

Controller. The objective of the controller is to optimize the expected rewards of the action, which is optimized using an actor critic approach. The actor critic approach considers the rewards beyond the horizon. Since we follow the DreamerV2, an action model and a value model are learnt in the imagined latent space of the world model. The action model implements a policy that aims to predict future actions that maximizes the total expected rewards in the imagined environment. Given H as the imagination horizon length, γ the discount factor for the future rewards, action and policy model are defined as follows:

$$\begin{aligned} &\text{Action model:} \quad q_\phi(a_t|s_t) \\ &\text{Value model:} \quad E_{q(\cdot|s_\tau)} \sum_{\tau=t}^{t+H} \gamma^{\tau-t} r_\tau. \end{aligned} \tag{5}$$

2.3 Implementation Details

We have used the publicly available code of DreamerV2 [3] and added the contrastive loss on top of that. We have used default hyperparameters of the continuous control task. Here, we have explained the necessary changes for our experiments. Following MoCo [22] and BYOL [16] we have used the moving average version of the query encoder to encode the keys K with a momentum value of 0.999. The contrastive loss is jointly optimized with the world model using Adam [23]. To encode the task observations we used two dense layers of size 32 with ELU activations [24]. The features from RGB image observation and task observation are concatenated before sending to the representation module of the DreamerV2. Replay buffer capacity is $3e^5$ for both 100k and 500k steps experiments. All architectural details and hyperparameters are provided in the appendix. Further, the training time of WMC is almost twice than that of DreamerV2 but inference time is the same.

3 Experiments

Evaluation. We evaluate the OoD generalization, sim-to-real transfer of perception learning and sample-efficiency of our model and baselines at 100k and 500k environment steps. Sample efficiency test on 100k and 500k steps is a common practice [3, 17, 18, 25]. Following [3], we update the model parameters on every fifth interactive step. We used default hyperparameter values of DreamerV2 for our experiments. Similarly, we used official code for RAD and CURL experiments.

Table 2: iGibson-to-Gibson dataset: sim-to-real perception transfer results on navigation task.

		Ihlen		Muleshoe		Uvalda		Noxapater		McDade	
Models	Steps	SR	SPL	SR	SPL	SR	SPL	SR	SPL	SR	SPL
RAD	100k	0.0	0.00	0.0	0.00	0.0	0.00	0.0	0.00	0.0	0.00
CURL	100k	5.9	0.05	3.8	0.03	5.1	0.04	5.9	0.05	12.8	0.11
WMC	100k	**24.4**	**0.18**	**20.4**	**0.15**	**24.3**	**0.18**	**27.3**	**0.21**	**40.9**	**0.31**
RAD	500k	26.4	0.23	27.5	0.24	28.5	0.25	28.6	0.25	40.0	0.34
CURL	500k	36.8	0.33	29.3	0.27	33.7	0.30	35.2	0.32	53.8	0.50
WMC	500k	**50.0**	**0.38**	**50.3**	**0.38**	**49.7**	**0.37**	**45.5**	**0.34**	**50.7**	**0.38**

iGibson Dataset. We have tested our proposed WMC on a random *PointGoal* task from iGibson 1.0 environment [14] for OoD generalization. It contains 15 floor scenes with 108 rooms. The scenes are replicas of real-world homes with artist designed textures and materials. We have used RGB, depth and task related observation only. Depth is only used during the training phase. The task related observation includes goal location, current location, and linear and angular velocities of the robot. Action includes rotation in radians and forward distance in meters for the Turtlebot. Since iGibson 1.0 does not provide dataset splits for OoD generalization, we have chosen five scenes for training and tested on the held-out three scenes and visual textures both. The details are provided in the appendix.

We have trained all models three times with random seeds and report the average *Success Rate* (SR) and *Success weighted by (normalized inverse) Path Length* (SPL) on held-out scenes as well as visual textures in the Table 1. Our proposed WMC outperforms state-of-the-art model-based RL method DreamerV2 and model-free method RAD and CURL on 100k and 500k interactive steps. Even though depth reconstruction and data augmentation improve the DreamerV2, proposed invariant causal features learning with contrastive loss further improves the results. All methods perform poorly on the difficult Ihlen_1_int scene. Further, our experimental results confirm that, similar to the model-free RL methods [17], data augmentation improves the performance of the model-based RL.

The experimental results of ablation study, which show the effects of input image decoding (I), depth prediction (D), data augmentation (DA) and action replay (AR) are also shown in Table 1. These results clearly shows the advantage of using proposed intervention invariant auxiliary task and contrastive learning together.

iGibson-to-Gibson Dataset. We use the Gibson dataset [13] for sim-to-real transfer experiments of the perception module, representation learning module of the world model; however please note that the robot controller is still a part of the simulator. Gibson scenes are created by 3D scanning of the real scenes, and it uses a neural network to fill the pathological geometric and occlusion errors only. We have trained all models on the artist created textures of iGibson and tested on five scenes from the Gibson. The results are shown in Table 2. Our proposed WMC outperforms RAD and CURL on 100k and 500k interactive steps, which shows that WMC learns more stable features and is better suited for sim-to-real transfer.

4 Conclusion

In this work we proposed a method to learn *World Model with invariant Causal features* (WMC). These invariant causal features are learnt by minimizing contrastive loss between content invariance interventions of the observation. Since the world model learns the representation learning and policy of the agent independently, without providing the better supervisory signal for the representation learning module, the contrastive loss collapses. Hence, we proposed depth reconstruction as an auxiliary task, which is invariant to the proposed data augmentation techniques. Further, given an intervention in the observation space, WMC can *extract* as well as *predict* the related causal features.

Our proposed WMC significantly outperforms the state-of-the-art models on out-of-distribution generalization, sim-to-real transfer of perception module and sample efficiency measures. Further, our method works on a sample-level intervention and does not need data from different environments to learn the invariant causal features.

References

[1] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018. ISSN 0036-8075.

[2] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, 2018.

[3] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

[4] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

[5] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):157–194, 1988.

[6] Kenneth A Bollen. *Structural equations with latent variables*, volume 210. John Wiley & Sons, 1989.

[7] A. P. Dawid. Causal inference without counterfactuals. *Journal of the American Statistical Association*, 95(450):407–424, 2000.

[8] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2 edition, 2009.

[9] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

[10] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2020.

[11] Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. Invariant causal prediction for block MDPs. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[12] Jovana Mitrovic, Brian McWilliams, Jacob C Walker, Lars Holger Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. In *International Conference on Learning Representations*, 2021.

[13] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: real-world perception for embodied agents. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[14] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne Tchapmi, Micael Tchapmi, Kent Vainio, Josiah Wong, Li Fei-Fei, and Silvio Savarese. igibson 1.0: A simulation environment for interactive tasks in large realistic scenes. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[15] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020.

[17] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. In *Advances in Neural Information Processing Systems*, 2020.

[18] Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021.

[19] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[20] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018.

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

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[25] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, PMLR 119*, 2020. arXiv:2004.04136.

A Qualitative Results of WMC

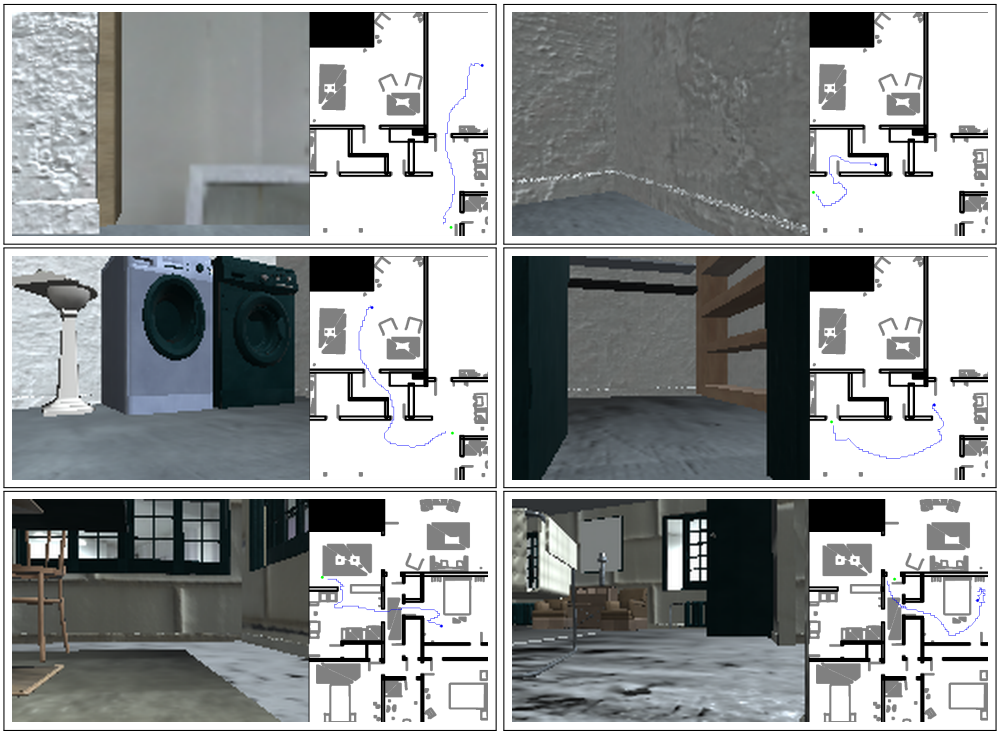


Figure 2: The out-of-distribution generalization tests of proposed WMC on held-out scenes and visual textures from iGibson 1.0 environment. Green circle is a random *PointGoal*, blue circle is a random starting point and blue line represents the travel path of the Turtlebot robot.

B Hyper Parameters

Table 3: Hyper parameters of proposed WMC.

Name	**Symbol**	**Value**
World Model		
Dataset size (FIFO)	—	$3 \cdot 10^5$
iGibson input image size	o	120×160
Batch size	B	50
Sequence length	L	50
Discrete latent dimensions	—	32
Discrete latent classes	—	32
RSSM number of units	—	1024
KL loss scale	β	1.0
World model learning rate	—	$3 \cdot 10^{-4}$
Key encoder exponential moving average	—	0.999
Behavior		
Imagination horizon	H	15
Actor learning rate	—	$1 \cdot 10^{-4}$
Critic learning rate	—	$1 \cdot 10^{-4}$
Slow critic update interval	—	100
Common		
Policy steps per gradient step	—	4
Policy and reward MPL number of layers	—	4
Policy and reward MPL number of units	—	400
Gradient clipping	—	100
Adam epsilon	ϵ	10^{-5}
Encoder and Decoder		
MLP encoder sizes of task obs	—	32, 32
Encoder kernels sizes	—	4, 4, 4, 4, 4
Decoder kernels sizes	—	5, 5, 4, 5, 4
Encoder and decoder feature maps	—	32, 64, 128, 256, 512
Encoder and decoder strides	—	2, 2, 2, 2, 2
Decoder padding	—	none, 0-1, none, none, none
Data Augmentation		
Padding range	—	10
Hue delta	—	0.1
Brightness delta	—	0.4
Contrast delta	—	0.4
Saturation delta	—	0.2
Gaussina blur sigma min, max	—	0.1, 2.0
Cutout min, max	—	30, 50

C iGibson 1.0 Training and Evaluation Splits

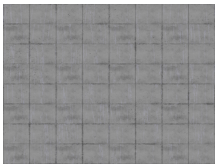
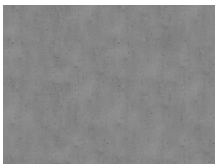
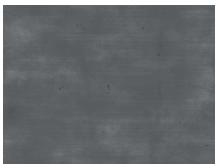
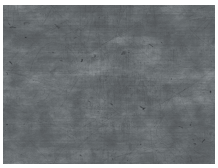
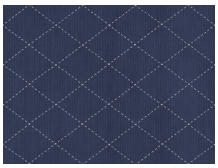
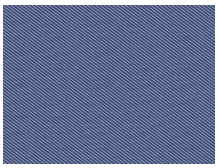
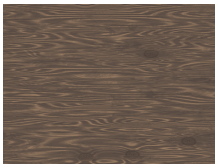
Table 4: Train-test scenes splits for iGibsion 1.0 dataset [14].

Phase	**Scene names**
Training	Beechwood_0_int, Beechwood_1_int, Benevolence_0_int, Benevolence_1_int, Benevolence_2_int Merom_0_int, Merom_1_int, Pomaria_0_int, Pomaria_1_int, Pomaria_2_int, Wainscott_0_int, Wainscott_1_int
Testing	Ihlen_0_int, Ihlen_1_int, Rs_int

Table 5: iGibson 1.0 environment [14] held out texture ids for test.

Material category	**Held-out texture ids for test**
asphalt	06, 15
bricks	08, 19
concrete	06, 15, 17
fabric	01, 02, 28
fabric_carpet	02, 05, 13
ground	13, 19
leather	03, 12
marble	02, 03
metal	10, 19
metal_diamond_plate	04
moss	01, 03
paint	05
paving_stones	24, 38
planks	07, 09, 16
plaster	03
plastic	04, 05
porcelain	02, 04
rocks	04
terrazzo	06, 08
tiles	43, 49
wood	02, 05, 16, 22, 32
wood_floor	06, 10, 17, 28

Table 6: Examples of textures split for training and testing from the proposed split in Table 5.

concrete	Train			
	Test			
fabric	Train			
	Test			
planks	Train			
	Test			
wood	Train			
	Test