Discrete Predictive Representation for Long-horizon Planning

Thanard Kurutach[*1] **Julia Peng**[*1] **Yang Gao** [1] **Stuart Russell** [1] **Pieter Abbeel** [1]

1 Introduction

In future, we hope that robots will be able to operate in unstructured environments such as homes and hospitals, and endowed with long-horizon planning ability. Despite successes in deep reinforcement learning (RL) from raw observations, much progress relies on the availability of shaped reward to guide the learning [31, 34]. On the other hand, over past decades, task and motion planning has been shown to solve much longer-horizon goal-directed tasks such as making a cup of coffee from torque control [20, 39, 40, 43]. However, these methods often require pre-specified discrete abstract states, task representations and transition models, e.g., whether the robot is holding a cup and what actions (or perturbations) change such an abstract state. In this paper, we aim to learn discrete representations for high-level abstract planning from video interaction data, combined with a learned short-horizon controller.

In this work, we propose Discrete Object-factorized Representations for Planning (DORP) – a novel framework for visual planning and control by learning discrete representations and a low-level controller. DORP learns discrete representations from images that change slowly overtime, such as whether or not the agent holds a key or which room the agent is in, along with a low-level predictive model for control. These slow features enable the agent to plan at a low frequency in longer-horizon tasks. More specifically, DORP represents an abstract state as a set of one-hot vectors, and optimizes its encoder by maximizing a mutual information lower bound between the current representations to future observations [36]. In order to train through the discrete layer, we apply the Gumbel-Softmax reparametrization trick [19, 30]. Using abstract states as nodes, we build an approximate feasibility graph based on observed transition data. When provided with new start and goal images, the agent plans the shortest abstract path. Using the next abstractions as waypoints, model-predictive control maximizes the objective that is 1 if it reaches the target abstraction and 0 otherwise with a trained video prediction model. Unlike other subgoal planning works [26, 27, 33, 38], which follows abstract waypoints, DORP avoids unnecessary steps to match exact waypoint states.

In a set of experiments, we demonstrate that DORP learns temporally-consistent and object-factorized representations suitable for planning. We show that these representations enable DORP to handle unseen long-horizon tasks more successfully compared to the states-of-the-art in visual planning. Interestingly, we observe that latent representations show object-level factorization such as key-and-door.

2 Discrete Object-factorized Representation for Planning

Our objective is to derive representation properties such as *object-factorization* and *temporal-consistency* from an unsupervised learning objective to facilitate long-horizon planning. When object representations are factorized, the agent can escape the need of combinatorial data configurations and can quickly generalize to unseen tasks. The connectivity graph can also be memory-efficiently represented and combined with a more powerful planning algorithm. Another important property for a representation is temporal consistency, meaning any two state observations in the same abstraction should be reachable from one-another by a short sequence of actions. When this property holds in the latent space, a high-level plan can be successfully executed by a low-level controller.

[1]Berkeley AI Research, University of California, Berkeley

NeurIPS 2020 Workshop on Object Representations for Learning and Reasoning, Vancouver, Canada.

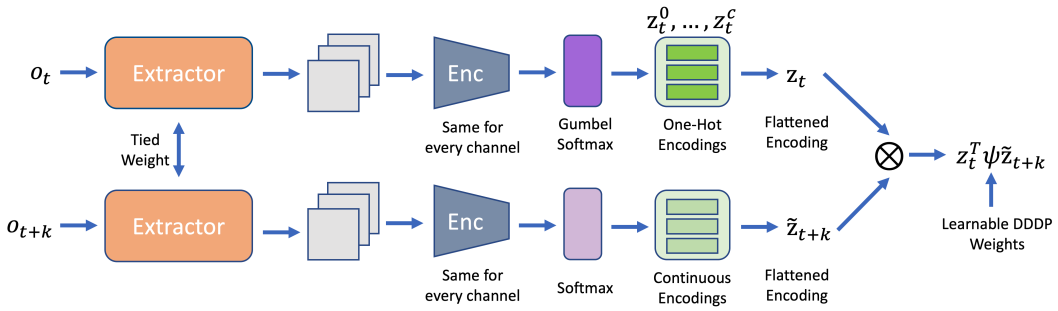


Figure 1: DORP architecture. DORP learns a set of one-hot encodings as the latent representation for each observation using CPC loss [36]. For a query observation o_t, its neighbours o_{t+k} for some small k are treated as positives, and random other observations are negatives. We propose a learnable weight matrix to be diagonally dominant and diagonally positive (DDDP). An object extractor architecture is applied to learn to extract objects and a shared encoder is applied per channel. Finally, Gumbel softmax and softmax are used to increase the gradient signal for backpropagation.

Problem Statement We define an unknown, fully-observable, stochastic dynamical system f which maps input observation $o_t \in \mathcal{O}$ and action $a_t \in \mathcal{A}$ to the next observation o_{t+1}. Under this dynamical system, we assume a simple exploration policy π_{rand} which can collect data that characterizes the dynamics of the system. This is known as self-supervised data or play data [1, 28]. We consider high-dimensional observation such as images.

Our goal is to learn discrete abstraction $z_t \in \mathcal{Z}$ given observation o_t, an abstract feasibility model, and a low-level local controller. At test time, given start o_s and goal o_g observations, we can plan a sequence of abstract states $z_s, z_1, ..., z_g$ where z_s and z_g are the representation of o_s and o_g, and apply the low-level controller to reach these abstract waypoints and finally to o_g.

Discrete Representation Learning Our starting point is contrastive predictive coding (CPC) [36] which learns low-dimensional representations that are most predictive of future high-dimensional sequential data. A non-linear encoder $q_\theta : \mathcal{O} \to R^l$, parametrized by θ, encodes the observation $o_t \in \mathcal{O}$ to a latent l-dimensional vector representation z_t. Let's define a similarity score $f_k(z_t, o_{t+k}) = \exp(z_t \psi q_\theta(o_{t+k}))$ where ψ is a trainable l-by-l similarity matrix and o_{t+k} is a future observation k steps ahead of o_t. Given the *query* observation o_t, we aim to classify the *key* observations – o_{t+k} as positive and other sample o from the dataset as negative. Formally, we optimize the loss function $\mathcal{L}_{CPC} = -E_{o_t, o_{t+k}} \left[\log f_k(z_t, o_{t+k}) - \log \sum_{o_j \in X} f_k(z_t, o_j) \right]$ with respect to θ and ψ. This also corresponds to maximizing a lowerbound of the mutual information between the latent representation z_t and the future observation o_{t+k}.

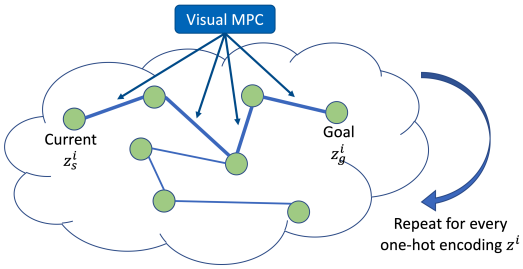


Figure 2: DORP planning and control. We build a graph on each one-hot encoding z^i at a time, and find the shortest path from current z_s^i to the goal z_g^i. We then apply visual MPC to follow it.

We depart from it by implementing our DORP architecture to encourage object-factorization and capture discrete representations. The extractor passes an anchor observation through a convolutional neural network and outputs a c-channel feature map. Each feature map is inputted to a shared encoder followed by a Gumbel softmax. For a positive and negative pair of images, however, we opt for a continuous embedding by swapping Gumbel softmax for softmax. Empirically, without this asymmetry trick, the optimization tends to converge to a poor local optima. Together these one-hots are flattened into long vectors, and their bilinear product is the similarity score between the query-key pair (see Figure 1).

We propose an inductive bias in the similarity score function to encourage temporal consistency. Instead of a fully trainable matrix, we parameterize the similarity matrix ψ to be diagonally dominant

and diagonally positive (DDDP). This property biases the similarity score to be high when the key embedding z_{t+k} is close to the query embedding z_t and maximized when $z_{t+k} = z_t$. In other words, this incentivizes the representations of positive pairs to share as many one-hots as possible. We reparameterize the weight matrix as $e^{\psi_0}(-(\psi_1) + \alpha I_{l\times l})$ where $I_{l\times l}$ is an identity matrix, α is a positive constant, is a sigmoid function, ψ_0 is a trainable scalar, and ψ_1 is a trainable l-by-l matrix.

Abstract Planning A critical component in planning is a connectivity graph that decides whether two discrete representations are connected. One naive solution is to build such graph from data – all observed representations as nodes and all observed transitions as edges. However as the number of independent entities increases, the amount of data required to generate the nodes and edges in order to cover the state space grows exponentially. To solve this, we propose to reduce the planning problem by exploiting factorization: 1. Embed the current image o_s and goal o_g as $z_s = q_\theta(o_s)$ and $z_g = q_\theta(o_g)$. 2. Pick a random one-hot index i s.t. the current state's one-hot z_s^i differs from the goal's one-hot z_g^i. 3. Build a graph based on how this one-hot transits in the data (ignoring all other one-hots). 4. Plan a path $z_1^i, ..., z_g^i$ from z_s^i to z_g^i. If this does not exists, then we fail the task. 5. Execute the low-level controller following the next target z_k^i, $k = 1, \cdots, g$, while keeping other z^j the same. 6. If it doesn't reach z_k^i, remove the edge from the graph and redo the planning in step 4. 7. If it succeeds, follow the next target until it reaches z_g^i. 8. Repeat step 2 until all the one-hots match. 9. Finally, execute the low-level controller to the goal image o_g. (see Figure 2)

To incorporate interactions between one-hots, we can extend this algorithm by building a graph on a random set of one-hot indices at a time in step 2. When this set contains all the one-hots, the method is equivalent to the full graph search (complete but expensive). Thus, our extended planning method trades efficiency for completeness. We demonstrate this further in our experiments.

Low-level Control To achieve different waypoints and goals, we deploy visual model-predictive controller (MPC) which models the dynamics of the world from past data and minimizes with the total predicted cost of horizon T at run-time: $a_t^*, ..., a_{t+T-1}^* = _{a_t,...,a_{t+T-1}} E\left[\sum_{i=1}^{T} \gamma^t \hat{c}_{t+i}\right]$. The predicted cost is computed by applying a known cost function on the predicted outcome of a T-step action sequence from the current observation o_t. Specifically, we implement a stochastic video prediction model [4], and define two cost functions for reaching abstract goals and for reaching the final goal observation. The first cost function is defined to be 1 if the current embedding exactly matches the goal embedding (all the one-hots match) and 0 otherwise: $c_{lat}(o_t, z') = \mathbb{1}[q_\theta(o_t) == z']$. The second cost function is defined by its L2 loss in the observation space to the goal: $c_{obs}(o_t, o_g) = ||o_t - o_g||$. This cost is versatile for reaching nearby observations particularly to reach the goal observation once it has reached the goal code.

3 Experiments

We perform experiments aimed at answering the following questions: (1) Are DORP representations temporally-consistent for high-level planning? (3) Can DORP representations factorize objects that translate or change in appearances over time? (3) How does DORP compare to the visual planning SOTA in solving unseen long-horizon goal-directed tasks?

Environments We evaluate DORP on two main environments – object-rearrangement and key-room. All the observations are provided as color images of size $16 \times 16 \times 3$.

k-object-rearrangement Multiple (k) 2-by-4 blocks of different colors can each be manipulated independently by an agent. The tasks require the agent to manipulate these objects from a start configuration to a goal configuration. In each step, it can manipulate each block by one unit in one of the 4 directions. The agent collects data by randomly interacting with one object at each timestep in a purely exploratory manner, without any goal in mind. In this environment our trajectory has length one. That is, the configuration is randomly reset after every step.

key-room Two variations of key-room depend on the number of rooms – key-corridor (6 rooms and a corridor) and key-wall (2 rooms). A key and agent are represented by 1 pixel. A key is placed in a fixed location in a room. If the agent steps on the key, a door will be removed, allowing the agent to enter a locked room. At each step, the agent can move in one of the four directions by 1 unit. The agent collects data by a long random walk – 1500 steps for key-corridor and 1000 steps for key-wall. After each reset the agent will be placed randomly in one of the unlocked rooms. This environment is inspired by MiniGrid [8].

Algorithms	1 object		2 objects		3 objects		5 objects	
	SR	# steps	SR	# steps	SR	# steps	SR	# steps
DORP	**1.00**	$25_{\pm 15}$	**1.00**	$49_{\pm 22}$	**1.00**	$75_{\pm 28}$	**0.87**	$200_{\pm 69}$
– arbitary weight [36]	0.92	$26_{\pm 19}$	0.58	$52_{\pm 16}$	0.26	$52_{\pm 16}$	0.10	$262_{\pm 61}$
– identity weight [18]	**1.00**	$34_{\pm 12}$	0.94	$18_{\pm 7}$	0.48	$83_{\pm 23}$	0.05	224
– full graph	**1.00**	$25_{\pm 15}$	**1.00**	$56_{\pm 17}$	**1.00**	$76_{\pm 26}$	0.00	-
VF-RS [9]	0.83	$19_{\pm 19}$	0.58	$53_{\pm 25}$	0.25	$93_{\pm 24}$	0.00	-
VF-CEM [9]	1.00	$12_{\pm 15}$	0.60	$47_{\pm 25}$	0.30	$89_{\pm 27}$	0.09	150

Table 1: Success rates in k-object-rearrangement across 50 unseen tasks. DORP is able to successfully solve most of the tasks as we increase the number of objects while other methods' performances degrade rapidly. We terminate when the agent has not reached the goal within the step limits. The number of steps are averaged only over the successful tasks. See Appendix B for baseline details.

Long-horizon planning We test the agent by its ability to solve unseen goal-directed tasks. In k-object-rearrangement, we can study the effect of increasing the length and complexity of the tasks by increasing the number of objects. In Table 1, we demonstrate that even as the number of objects increases, DORP is able to succeed in most of the tasks while other methods' performances degrade quickly. Note that when faced with 5 objects DORP employs the extended planning version by grouping the 5 one-hots into two groups of 2 and 3 one-hots. By considering more that one object at a time, the agent is able to perform non-myopic planning and achieve 87% success rate. We see similar trend in key-room, as longer planning horizon is required in key-corridor, visual foresight methods perform poorly compared to DORP (Table 2).

Algorithms	key-wall		key-corridor	
	SR	# steps	SR	# steps
DORP	**.91**	$74_{\pm 22}$	**0.98**	$64_{\pm 24}$
VF-RS	0.66	$75_{\pm 34}$	0.1	$101_{\pm 21}$
VF-CEM	0.06	$104_{\pm 18}$	0.02	$106_{\pm 14}$

Table 2: Success rates in two key-room environments across 50 sample tasks. For each task, the key is presented in the start image, but not goal. DORP is able to successfully solve most of tasks, while VF-RS and VF-CEM sometimes navigate to the nearby key but not the goal.

Temporal consistency We investigate DORP representations by color-coding the discrete embeddings on the configuration map of the environments. For visualization simplicity, we choose 1-object-rearrangement to visualize the learned code map. In this environment we have a single one-hot code for representing the object. We demonstrate that DORP learns temporally consistent representations in which two states from the same discrete code are connected by a short-horizon controller (see Figure 4(a)). Similarly, we observe the same property in the key-wall environment (see Figure 5).

Object factorization We evaluate the behavior of the one-hot codes by changing one property of the agent at a time such as positions or whether the agent has a key. In Figure 3(c) we observe that by moving one object at a time only at most one one-hot code changes its value k-object-rearrangement. In key-room, we formulate the discrete representation as follows: of the two output one-hot latents, the first is set to a size z_0 and the other is set to size 2. Our aim is for each latent to encode the key and agents' abstract states, respectively. In Figure 5, we demonstrate that the learned abstract representations are factorized as desired by observing (1) the size z_0 one-hot changes when the agent moves with no key interactions and (2) the binary one-hot changes when removing and adding the key while maintaining the agent position.

Weight Ablation We study the effect of the proposed similarity matrix against commonly-used similarity matrices [18, 36] in unsupervised representation learning. In Figure 4, we demonstrate that DORP is able to exploit more available latent codes, therefore helping the latent space be more temporally consistent.

4 Conclusion and Future Directions

In this paper, we propose an unsupervised discrete representation learning method for long term planning, which can extract high level abstract states that are good for planning in an purely unsupervised manner. We demonstrate DORP's effectiveness over other methods on challenging long horizon tasks. We note that our method generate approximate plans more tractably and hence trades off optimality for efficiency, like other state abstraction works [10, 27, 38]. Those approximate solutions can be used to initialize a model-free policy which can be later fine-tuned to reach optimality. We leave this as our future work.

References

[1] Agrawal, Pulkit, Nair, Ashvin V, Abbeel, Pieter, Malik, Jitendra, and Levine, Sergey. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in Neural Information Processing Systems*, pp. 5074–5082, 2016.

[2] Aloimonos, John, Weiss, Isaac, and Bandyopadhyay, Amit. Active vision. *International journal of computer vision*, 1(4):333–356, 1988.

[3] Asai, Masataro and Fukunaga, Alex. Classical planning in deep latent space: Bridging the subsymbolic-symbolic boundary. *arXiv preprint arXiv:1705.00154*, 2017.

[4] Babaeizadeh, Mohammad, Finn, Chelsea, Erhan, Dumitru, Campbell, Roy H, and Levine, Sergey. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

[5] Bajcsy, Ruzena. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

[6] Bajcsy, Ruzena, Aloimonos, Yiannis, and Tsotsos, John K. Revisiting active perception. *Autonomous Robots*, 42(2):177–196, 2018.

[7] Burgess, Christopher P, Matthey, Loic, Watters, Nicholas, Kabra, Rishabh, Higgins, Irina, Botvinick, Matt, and Lerchner, Alexander. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

[8] Chevalier-Boisvert, Maxime, Willems, Lucas, and Pal, Suman. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018.

[9] Ebert, Frederik, Finn, Chelsea, Dasari, Sudeep, Xie, Annie, Lee, Alex, and Levine, Sergey. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

[10] Eysenbach, Ben, Salakhutdinov, Russ R, and Levine, Sergey. Search on the replay buffer: Bridging planning and reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 15246–15257, 2019.

[11] Finn, Chelsea, Goodfellow, Ian, and Levine, Sergey. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems*, pp. 64–72, 2016.

[12] Fitzpatrick, Paul. First contact: an active vision approach to segmentation. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pp. 2161–2166. IEEE, 2003.

[13] Gibson, James J. *The ecological approach to visual perception: classic edition*. Psychology Press, 2014.

[14] Greff, Klaus, Kaufman, Raphaël Lopez, Kabra, Rishabh, Watters, Nick, Burgess, Chris, Zoran, Daniel, Matthey, Loic, Botvinick, Matthew, and Lerchner, Alexander. Multi-object representation learning with iterative variational inference. *arXiv preprint arXiv:1903.00450*, 2019.

[15] Ha, David and Schmidhuber, Jürgen. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[16] Hafner, Danijar, Lillicrap, Timothy, Fischer, Ian, Villegas, Ruben, Ha, David, Lee, Honglak, and Davidson, James. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pp. 2555–2565. PMLR, 2019.

[17] Hausman, Karol, Pangercic, Dejan, Márton, Zoltán-Csaba, Bálint-Benczédi, Ferenc, Bersch, Christian, Gupta, Megha, Sukhatme, Gaurav, and Beetz, Michael. Interactive segmentation of textured and textureless objects. In *Handling Uncertainty and Networked Structure in Robot Control*, pp. 237–262. Springer, 2015.

[18] He, Kaiming, Fan, Haoqi, Wu, Yuxin, Xie, Saining, and Girshick, Ross. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

[19] Jang, Eric, Gu, Shixiang, and Poole, Ben. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[20] Kaelbling, Leslie Pack and Lozano-Pérez, Tomás. Hierarchical task and motion planning in the now. In *2011 IEEE International Conference on Robotics and Automation*, pp. 1470–1477. IEEE, 2011.

[21] Kansky, Ken, Silver, Tom, Mély, David A, Eldawy, Mohamed, Lázaro-Gredilla, Miguel, Lou, Xinghua, Dorfman, Nimrod, Sidor, Szymon, Phoenix, Scott, and George, Dileep. Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. *arXiv preprint arXiv:1706.04317*, 2017.

[22] Kenney, Jacqueline, Buckley, Thomas, and Brock, Oliver. Interactive segmentation for manipulation in unstructured environments. In *2009 IEEE International Conference on Robotics and Automation*, pp. 1377–1382. IEEE, 2009.

[23] Kipf, Thomas, van der Pol, Elise, and Welling, Max. Contrastive learning of structured world models. *arXiv preprint arXiv:1911.12247*, 2019.

[24] Kurutach, Thanard, Tamar, Aviv, Yang, Ge, Russell, Stuart J, and Abbeel, Pieter. Learning plannable representations with causal infogan. In *Advances in Neural Information Processing Systems*, pp. 8733–8744, 2018.

[25] Kwak, Suha, Cho, Minsu, Laptev, Ivan, Ponce, Jean, and Schmid, Cordelia. Unsupervised object discovery and tracking in video collections. In *Proceedings of the IEEE international conference on computer vision*, pp. 3173–3181, 2015.

[26] Laskin, Michael, Emmons, Scott, Jain, Ajay, Kurutach, Thanard, Abbeel, Pieter, and Pathak, Deepak. Sparse graphical memory for robust planning. *arXiv preprint arXiv:2003.06417*, 2020.

[27] Liu, Kara, Kurutach, Thanard, Tung, Christine, Abbeel, Pieter, and Tamar, Aviv. Hallucinative topological memory for zero-shot visual planning. *arXiv preprint arXiv:2002.12336*, 2020.

[28] Lynch, Corey, Khansari, Mohi, Xiao, Ted, Kumar, Vikash, Tompson, Jonathan, Levine, Sergey, and Sermanet, Pierre. Learning latent plans from play. In *Conference on Robot Learning*, pp. 1113–1132, 2020.

[29] Ma, Xiao, Chen, Siwei, Hsu, David, and Lee, Wee Sun. Contrastive variational model-based reinforcement learning for complex observations. *arXiv preprint arXiv:2008.02430*, 2020.

[30] Maddison, Chris J, Mnih, Andriy, and Teh, Yee Whye. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[31] Mirza, Mehdi, Jaegle, Andrew, Hunt, Jonathan J, Guez, Arthur, Tunyasuvunakool, Saran, Muldal, Alistair, Weber, Théophane, Karkus, Peter, Racanière, Sébastien, Buesing, Lars, et al. Physically embedded planning problems: New challenges for reinforcement learning. *arXiv preprint arXiv:2009.05524*, 2020.

[32] Nair, Suraj and Finn, Chelsea. Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. *arXiv preprint arXiv:1909.05829*, 2019.

[33] Nasiriany, Soroush, Pong, Vitchyr, Lin, Steven, and Levine, Sergey. Planning with goal-conditioned policies. In *Advances in Neural Information Processing Systems*, pp. 14843–14854, 2019.

[34] Ng, Andrew Y, Harada, Daishi, and Russell, Stuart. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pp. 278–287, 1999.

[35] Okada, Masashi and Taniguchi, Tadahiro. Dreaming: Model-based reinforcement learning by latent imagination without reconstruction. *arXiv preprint arXiv:2007.14535*, 2020.

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

[37] Pertsch, Karl, Rybkin, Oleh, Ebert, Frederik, Finn, Chelsea, Jayaraman, Dinesh, and Levine, Sergey. Long-horizon visual planning with goal-conditioned hierarchical predictors. *arXiv preprint arXiv:2006.13205*, 2020.

[38] Savinov, Nikolay, Dosovitskiy, Alexey, and Koltun, Vladlen. Semi-parametric topological memory for navigation. *arXiv preprint arXiv:1803.00653*, 2018.

[39] Srivastava, Siddharth, Fang, Eugene, Riano, Lorenzo, Chitnis, Rohan, Russell, Stuart, and Abbeel, Pieter. Combined task and motion planning through an extensible planner-independent interface layer. In *2014 IEEE international conference on robotics and automation (ICRA)*, pp. 639–646. IEEE, 2014.

[40] Toussaint, Marc. Logic-geometric programming: An optimization-based approach to combined task and motion planning. In *IJCAI*, pp. 1930–1936, 2015.

[41] Veerapaneni, Rishi, Co-Reyes, John D, Chang, Michael, Janner, Michael, Finn, Chelsea, Wu, Jiajun, Tenenbaum, Joshua, and Levine, Sergey. Entity abstraction in visual model-based reinforcement learning. In *Conference on Robot Learning*, pp. 1439–1456. PMLR, 2020.

[42] Wang, Ning, Song, Yibing, Ma, Chao, Zhou, Wengang, Liu, Wei, and Li, Houqiang. Unsupervised deep tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1308–1317, 2019.

[43] Wang, Zi, Garrett, Caelan Reed, Kaelbling, Leslie Pack, and Lozano-Perez, Tomas. Active model learning and diverse action sampling for task and motion planning. In *International Conference on Intelligent Robots and Systems (IROS)*, 2018. URL http://lis.csail.mit.edu/pubs/wang-iros18.pdf.

[44] Watter, Manuel, Springenberg, Jost, Boedecker, Joschka, and Riedmiller, Martin. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in neural information processing systems*, pp. 2746–2754, 2015.

[45] Watters, Nicholas, Matthey, Loic, Bosnjak, Matko, Burgess, Christopher P, and Lerchner, Alexander. Cobra: Data-efficient model-based rl through unsupervised object discovery and curiosity-driven exploration. *arXiv preprint arXiv:1905.09275*, 2019.

[46] Xiao, Fanyi and Jae Lee, Yong. Track and segment: An iterative unsupervised approach for video object proposals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 933–942, 2016.

[47] Yan, Wilson, Vangipuram, Ashwin, Abbeel, Pieter, and Pinto, Lerrel. Learning predictive representations for deformable objects using contrastive estimation. *arXiv preprint arXiv:2003.05436*, 2020.

A Related Work

Object Discovery Recent work has studied unsupervised object segmentations from visual inputs [7, 14], and entity-factorized representations and models for predictions [23]. However, these studies have not been demonstrated to directly solve the task. In this work, we take a step further to evaluate the representations in downstream tasks. While other work [41, 45] shows how the MPC agents benefit from object-factorized models, they require shaped rewards for the tasks. In contrast, we consider a long-horizon task in which the agent only receives its reward when it has reached the goal. A large body of work in computer vision has studied unsupervised video object segmentation [2, 5, 6, 12, 13, 17, 22] and unsupervised object detection [25, 42, 46]. However, semantic segmentation and object detection might not be the correct representation for the end-task. In this work, we learn representations that are more ready to use for planning without explicit segmentation or detection.

World models Much previous work has applied generative models of the world to visual control tasks [15, 16, 44]. Other work [29, 35, 47] leverages a contrastive objective to learn a latent world model. These methods however require dense reward signals for most tasks. Visual foresight methods [9, 11] demonstrate impressive results on real robots using unshaped reward such as pixel distance to the goal, but it still remains limited to short-horizon object pushing tasks. In our work, we borrow these methods for low-level controllers. Asai & Fukunaga [3] learn discrete abstraction of the system such as an 8-piece puzzle, but do not consider temporal abstraction. Kansky et al. [21] demonstrate that discrete object-factorized representations can be used to learn logic-based transitions. In combination with powerful planning, it improves generalization and data efficiency. However, it assumes supervised ground-truth labels for representation learning. In this work, we aim to learn this in an unsupervised manner.

Hierarchial RL Recent work has tried to approach long-horizon visual planning by breaking down tasks using skills [23, 28]. Our approach is orthogonal and may deploy such action abstractions as a low-level controller. Other methods propose to plan subgoals and attempt to follow them either in the latent space [24, 32, 33] or in the visual space [10, 27, 38]. However, such methods require labor-intensive engineering to tune the threshold on when to move on to pursue the next subgoal [27, 38] because the observations or the latent states can never exactly match. Imagine bringing a chair from one office to another. It would be time consuming to match all the positions and orientations of the chair along the way. Rather we should care about rough area the chair has to go through in order to reach the goal. In our work, by learning the discrete codes, our method do not require such threshold as it exactly knows when it has reached the target discrete representation. Additionally, it avoids aiming to match a specific waypoint observation. Instead of planning to the goal, Pertsch et al. [37] predict intermediate images iteratively to construct a subgoal tree. However in order to train such prediction model a long-sequential training data are required. Instead our work can be applied to short-horizon or long-horizon trajectories.

B Experimental Details

Baselines We choose the state-of-the-arts visual foresight (VF) [9] implemented using SV2P architecture [4] as our baseline for two reasons. First, VF only requires self-supervised data collection and is applicable to unseen tasks – thus aligning with goal-directed visual planning problems. Second, Visual Foresight is a low-level controller of DORP. By sharing the same video model as baseline, we show a direct improvement of abstract planning for temporally-extended tasks. Two optimization variations of VF – VF with random shooting (VF-RS) and VF with cross-entropy method (VF-CEM) – both share the same video prediction model. Their objective is to minimize its L2 distance from the current image to the goal image. While DORP deploys VF-RS for its low-level control, VF-CEM comparision are provided for an improved baseline. VF-RS randomizes 1000 trajectories and take the full action sequence that achieves the minimum cost. VF-CEM randomizes 1000 samples per iteration. We pick the top 2% to refit a distribution over sequence and repeat for 3 iterations.

To understand DORP representation learning improvement, we evaluate DORP when replacing its DDDP similarity matrix by an arbitrary weight [36] or an identity matrix [18] which have been used extensively in unsupervised representation learning. To understand the benefits of factorized planning, we replace it with the full graph planning with a maximum limit on the number of steps allowed.

C Additional Results

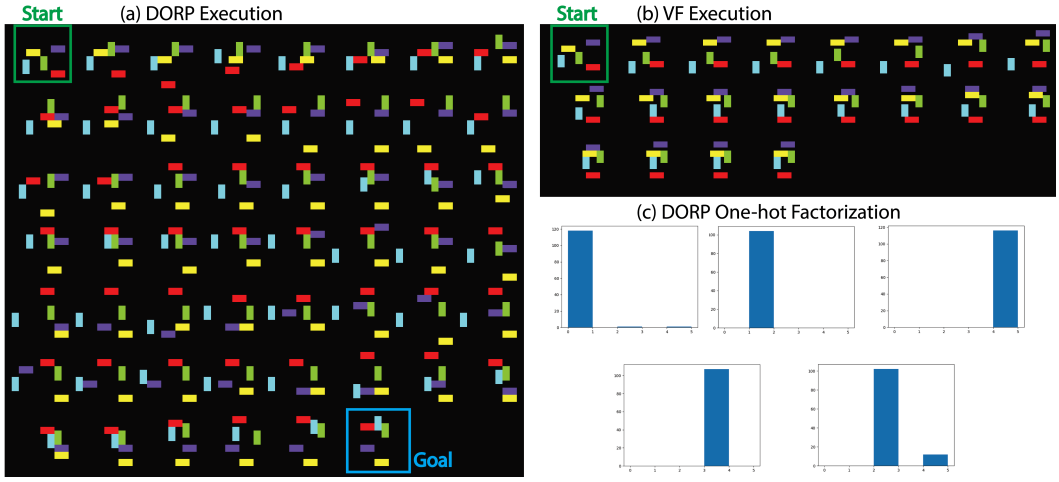


Figure 3: DORP in 5-object-rearrangement. In (a), we present DORP with random unseen start and goal images which require temporal-extended planning. In (b), when presenting the same task to VF-CEM we find that the objects are stuck in an awkward configuration where 4 blocks (except the purple) are blocking to reach their goal positions. Finally, in (c), we visualize the representation factorization by randomly moving one of the five object while maintaining the positions of the others. We plot a histogram of which one-hots have been changed per object. We find that with high probability only the one-hot that corresponds to the moving object is modified.

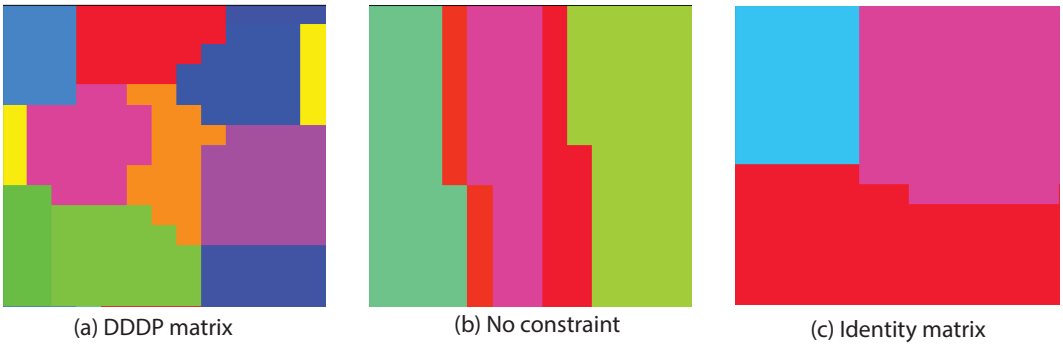
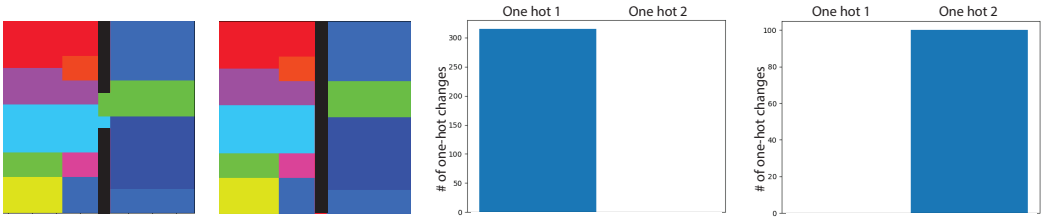


Figure 4: Discrete Embedding Comparison. We visualize the color codes of different object positions in 1-object-rearrangement per similarity matrix type. Each color shows different discrete code. The one-hot embedding has size 16 for all settings. In (a), we find that DORP discrete representation is temporally-consistent, i.e., two states that map to the same embedding are connected by a short sequence of actions. In (b), when using an arbitrary weight matrix the embedding is less temporally consistent. In (c), when using an identity matrix the, the embedding uses only 3 out of 16 available codes



(a) Representation map with key

(b) Representation map without key

(c) Onehot changes when moving only the agent without key interaction

(d) Onehot changes when adding and removing the key on the grid while fixing agent positions

Figure 5: Key-Wall Representations. In (a) and (b), we demonstrate the discrete code of the agent at different positions. Each color represents the same code. The grids in black are invalid states (the wall that blocks the agent and separates the two rooms). We demonstrate temporal consistency in the latent space both when the object has the key as not. In (c) and (d), we confirm that two one-hot codes are factorized by observing that only one one-hot changes when removing the key while maintaining the agent position and only the other one-hot changes when the agent moves without interacting with the key.