Hierarchical Foresight: Self-Supervised Learning of Long-Horizon Tasks via Visual Subgoal Generation

Suraj Nair[1,†], **Chelsea Finn**[1,2]
[1]Stanford University, [2]Google Brain

1 Introduction

Developing robotic systems that can complete long horizon visual manipulation tasks, while generalizing to novel scenes and objectives, remains an unsolved and challenging problem. Generalization to unseen objects and scenes requires robots to be trained across diverse environments, meaning that detailed supervision during data collection in not practical to provide. Furthermore, reasoning over long-horizon tasks introduces two additional major challenges. First, the robot must handle large amounts of uncertainty as the horizon increases. And second, the robot must identify how to reach distant goals when only provided with the final goal state, a sparse indication of the task, as opposed to a shaped cost that implicitly encodes how to get there. In this work, we aim to develop a method that can address these challenges, leveraging self-supervised models learned using only unlabeled data, to solve novel temporally-extended tasks.

The key insight that we leverage is that while model error and sparse cost signals can make long horizon planning difficult, we can mitigate these issues by *learning to break down long-horizon tasks into short horizon segments*. Consider, for example, the long horizon task of opening a drawer and putting a book in it, given supervision only in the form of the final image of the open drawer containing the book. The goal image provides nearly no useful cost signal until the last stage of the task, and model predictions are likely to become inaccurate beyond the first stage of the task. However, if we can generate good subgoals, such as (1) the robot arm grasping the drawer handle, (2) the open drawer, and (3) the robot arm reaching for the book, planning from the initial state to (1), from (1) to (2), from (2) to (3), and from (3) to the goal, the problem becomes substantially easier.

Our main contribution is a self-supervised hierarchical planning framework, hierarchical visual foresight (HVF), which combines generative models of images and model predictive control to decompose a long-horizon visual task into a sequence of subgoals. In particular, we propose optimizing over subgoals such that the resulting task subsegments have low expected planning cost. However, in the case of visual planning, optimizing over subgoals corresponds to *optimizing within the space of natural images*. To address this challenge, we train a generative latent variable model over images from the robot's environment and optimize over subgoals in the latent space of this model. This allows us to optimize over the manifold of images with only a small number of optimization variables. When combined with visual model predictive control, we observe that this subgoal optimization naturally identifies semantically meaningful states in a long horizon tasks as subgoals, and that when using these subgoals during planning, we achieve significantly higher success rates on long horizon, multi-stage visual tasks. Furthermore, since our method outputs subgoals conditioned on a goal image, we can use the same model and approach to plan to solve many different long horizon tasks, even with previously unseen objects. We first demonstrate our approach in simulation on a continuous control navigation task with tight bottlenecks, and then evaluate on a set of four different multi-stage object manipulation tasks in a simulated desk environment, which require interacting with up to 3 different objects. In the challenging desk environment, we find that our method yields nearly a 200% performance improvement over prior approaches. Finally, we show that our approach generates realistic subgoals on real robot manipulation data.

[†]Work completed at Google Brain

NeurIPS 2019 Workshop on Robot Learning: Control and Interaction in the Real World, Vancouver, Canada

2 Related Work

Developing robots that can execute complex behaviours from only pixel inputs has been a well studied problem, for example with visual servoing [1, 2, 3, 4, 5, 6, 7, 8]. Recently, reinforcement learning has shown promise in completing complex tasks from pixels [9, 10, 11, 12, 13, 14, 15, 16, 17]. While model-free RL approaches have illustrated the ability to generalize to new objects [11] and learn tasks such as grasping and pushing through self-supervision [18, 19], pure model-free approaches generally lack the ability to explicitly reason over temporally-extended plans, making them ill-suited for the problem of learning long-horizon tasks with limited supervision.

Within the space of goal-conditioned policy learning [20, 21, 22, 7, 8, 23], video prediction and planning have also shown promise in enabling robots to complete a diverse set of visuomotor tasks while generalizing to novel objects [24, 25, 26, 27]. Since then a number of video prediction frameworks have been developed specifically for robotics [28, 29, 30], which combined with planning have been used to complete diverse behaviors [31, 32, 33, 34]. However, these approaches still struggle with long horizon tasks, which we specifically focus on.

One approach to handle long horizon tasks is to add compositional structure to policies, either from demonstrations [35, 36], with manually-specified primitives [37, 38], learned temporal abstractions [39], or through model-free reinforcement learning [40, 41, 42, 43, 44]. These works have studied such hierarchy in grid worlds [42] and simulated control tasks [43, 45, 44] with known reward functions. In contrast, we study how to incorporate compositional structure in learned model-based planning with video prediction models. Our approach is *entirely self-supervised*, without motion primitives, demonstrations, or shaped rewards, and scales to vision-based manipulation tasks. Like our work, a number of recent works have explored reaching novel goals using only self-supervision [24, 46, 47, 48, 49, 23]. While [46] presents hierarchical planning in the domain of visual navigation, we focus on the problem of tabletop manipulation. Time-agnostic prediction (TAP) [49] aims to identify bottlenecks in long-horizon visual manipulation tasks, while [23, 24] reach novel goals using only self-supervision. We compare to all three of these methods in Section 4 and find that HVF significantly outperforms all of them.

3 Hierarchical Visual Foresight

Our key insight is that we can train a deep generative model, trained exclusively on self-supervised data, as a means to sample possible states. Once we can sample states, we also need to evaluate how easy it is to get from one sampled state to another, to determine if a state makes for a good subgoal. We can do so through planning: running visual MPC to get from one state to another and measuring the predicted cost of the planned action sequence. Thus by leveraging the low-dimensional space of a generative model and the cost acquired by visual MPC, we can optimize over a sequence of subgoals that lead to the goal image. In particular, we can explicitly search in latent image space for subgoals, such that no segment is too long-horizon, mitigating the issues around sparse costs and compounding model error.

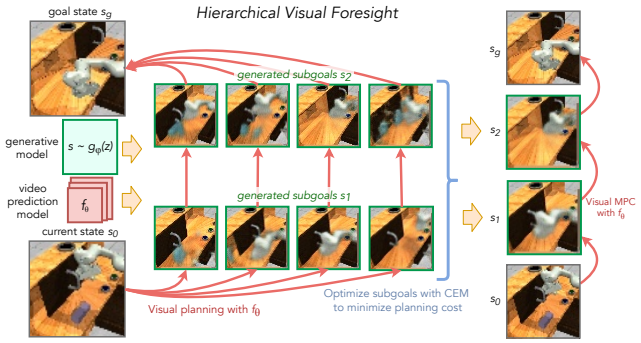

Figure 1: **Hierarchical visual foresight:** Our method takes as input the current image, goal image, an action conditioned video prediction model f_θ, and a generative model $g_\phi(z)$. Then, it samples sets of possible states from $g_\phi(z)$ as sub-goals. It then plans between each sub-goal, and iteratively optimizes the sub-goals to minimize the worst case planning cost between any segment. The final set of sub-goals that minimize planning cost are selected, and the agent completes the task via visual model predictive control with the sub-goals in sequence. In this example the task is to push a block off the table, and then slide the door shut. Given only the goal image, HVF produces sub-goals for (1) pushing the block off and reaching to the door and (2) sliding the door shut.

Hierarchical Visual Foresight: Formally, we assume the goal conditioned MDP where the agent has a current state s_0, goal state s_g, cost function $\mathcal{C}$, and dataset of environment interaction $\{(s_1, a_1, s_2, a_2, ..., s_T, a_T)\}$. This data can come from any exploration policy; in practice, we find that interaction from a uniformly random policy in

the continuous action space of the agent works well. From this data, we train both a dynamics model f_θ using maximum likelihood supervised learning, as well as a generative model $s \sim g_\phi$.

Now given s_0 and s_g, the objective is to find K subgoals $s_1, s_2, ..., s_K$ that enable easier completion of the task. Our hope is that the subgoals will identify steps in the task such that, for each subsegment, the planning problem is easier and the horizon is short. While one way to do this might be to find subgoals that minimize the total planning cost, we observe that this does not necessarily encourage splitting the task into shorter segments. Consider planning in a straight line: using any point on that line as a subgoal would equally minimize the total cost. Therefore, we instead optimize for subgoals that minimize the worst expected planning cost across any segment. This corresponds to the following objective:

$$\min_{s_1,...,s_K} \max\{\mathcal{C}_{plan}(s_0, s_1), \mathcal{C}_{plan}(s_1, s_2), ..., \mathcal{C}_{plan}(s_K, s_g)\} \tag{1}$$

where $\mathcal{C}_{plan}(s_i, s_j)$ is the cost achieved by the planner when planning from s_i to s_j, which we compute by planning a sequence of actions to reach s_j from s_i using f_θ and measuring the predicted cost achieved by that action sequence[1]. Once the subgoals are found, then the agent simply plans using visual MPC [24, 32] from each s_{k-1} to s_k until a cost threshold is reached or for a fixed, maximum number of timesteps, then from s_k^* to s_{k+1}, until planning to the goal state, where s_k^* is the actual state reached when running MPC to get to s_k. We describe the individual components and provide algorithm pseudo code in the supplementary material.

4 Experiments

In our experiments, we aim to evaluate **(1)** if, by using HVF, robots can perform challenging goal-conditioned long-horizon tasks from raw pixel observations more effectively than prior self-supervised approaches, **(2)** if HVF is capable of generating realistic and semantically significant subgoal images, and **(3)** if HVF can scale to real images of cluttered scenes. To do so, we test on three domains: simulated visual maze navigation (in supplement), simulated desk manipulation, and real robot manipulation of diverse objects. The simulation environments use the MuJoCo physics engine [50]. We compare against three prior methods: (a) visual foresight [24, 32], which uses no subgoals, (b) RIG [23] which trains a model-free policy to reach generated goals using latent space distance as cost, and (c) visual foresight with subgoals generated by time-agnostic prediction (TAP) [49], which generates subgoals by predicting the most likely frame between the current and goal state, a state-of-the-art method for self-supervised generation of visual subgoals. Lastly, we perform ablation studies to determine the effect of various design choices of HVF (in supplement)

4.1 Simulated Desk Manipulation

We study the performance improvement and subgoal quality of HVF in a challenging simulated robotic manipulation domain. Specifically, a simulated Franka Emika Panda robot arm is mounted in front of a desk (as used in [51]). The desk consists of 3 blocks, a sliding door, three buttons, and a drawer. We explore four tasks in this space: door closing, 2 block pushing, door closing + block pushing, and door closing + 2 block pushing. Example start and goal images for each task are visualized in Figure 3, and task details are in the supplementary material. The arm is controlled with 3D Cartesian velocity control of the end-effector position. Across the 4 different tasks in this environment, we use a single dynamics model f_θ and generative model $g_\phi(z)$. Experimental details are in the supplementary material.

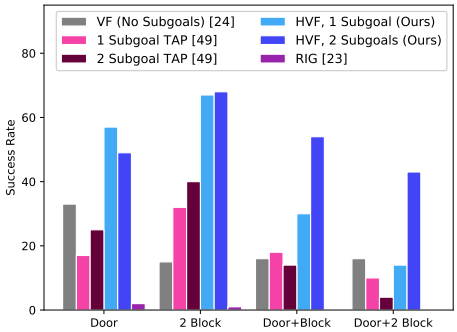

Figure 2: **Quantitative Results for Desk Manipulation:** For the challenging desk manipulation tasks, using the HVF subgoals drastically improves performance. Across all 4 tasks, HVF with two subgoals has over a 20% absolute performance improvement over visual foresight [24], TAP [49], and RIG [23]. Computed over 100 trials with random initial scenes.

[1]We compare max/mean cost in the supplement

Results: As seen in Figure 2, we find that using HVF subgoals dramatically improves performance, providing at least a 20% absolute improvement in success rate across the board. In the task with the longest horizon, closing the door and sliding two blocks off the table, we find that using no subgoals or 1 subgoal has approximately 15% performance, but 2 subgoals leads to over 42% success rate. We compare to subgoals generated by time agnostic prediction (TAP) [49] and find that while it does generate plausible subgoals, they are very close to the start or goal, leading to no benefit in planning. We also compare against RIG [23], where we train a model free policy in the latent space of the VAE to reach "imagined" goals, then try and reach the unseen goals. However, due to the complexity of the environment, we find that RIG struggles to reach even the sampled goals during training, and thus fails on the unseen goals. Qualitatively, in Figure 3, we also observe that HVF outputs meaningful subgoals on the desk manipulation tasks. For example, it often produces subgoals corresponding to grasping the door handle, sliding the door, or reaching to a block.

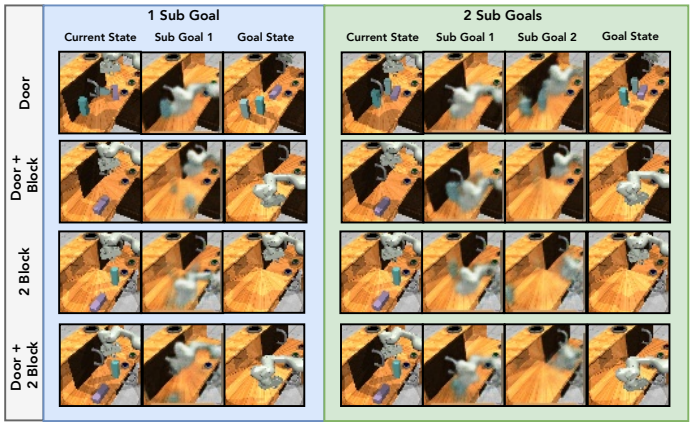

Figure 3: **Qualitative Results for Desk Manipulation.** Example generated subgoals from HVF for the desk manipulation tasks with one or two subgoal. We observe interesting behavior: for example in the Door Closing + Block Pushing task with one subgoal, the subgoal is to first push the block and then slide the door, while in the Door Closing + 2 Block Pushing the subgoal is to first push a block then grasp the door.

4.2 Real Robot Manipulation

Lastly, we aim to study whether HVF can extend to real images and cluttered scenes. To do so, we explore the qualitative performance of our method on the BAIR robot pushing dataset [52]. We train f_θ and $g_\phi(z, s_0)$ on the training set, and sample current and goal states from the beginning and end of test trajectories. We then qualitatively evaluate the subgoals outputted by HVF. Further implementation details are in the supplementary material.

Results: Our results are illustrated in Fig. 4. We observe that even in cluttered scenes, HVF produces meaningful subgoals, such generating grasping objects which need to be moved. Additionally, when grasping or interacting with objects, grasping the object is often selected as a subgoal (top left and bottom right examples in Fig. 4). For more examples, see supplementary material.

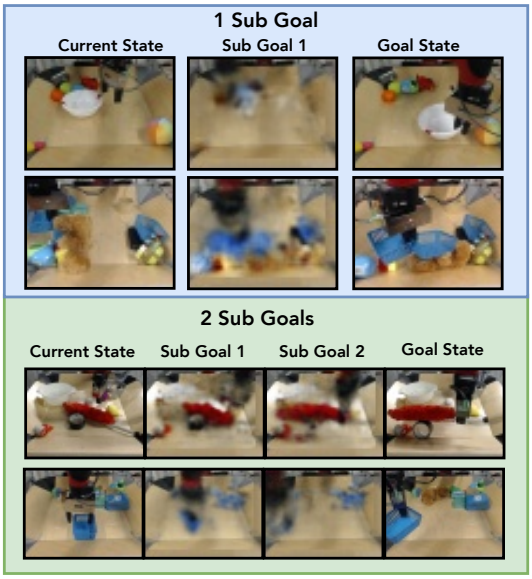

Figure 4: **BAIR Dataset Qualitative Results.** The subgoals generated by HVF on the BAIR robot data set, which we find correspond to meaningful states between the start and goal. For example, when moving objects we see subgoals of reaching/grasping the object.

5 Conclusion

We presented an approach for hierarchical planning with vision-based tasks, hierarchical visual foresight (HVF), which decomposes a visual goal into a sequence of subgoals. By explicitly optimizing for subgoals that minimize planning cost, HVF is able to discover semantically meaningful goals in visual space, and when using these goal for planning, perform a variety of challenging, long-horizon vision-based tasks. Further, HVF learns these tasks in an entirely self-supervised manner without rewards or demonstrations.

References

[1] K. Mohta, V. Kumar, and K. Daniilidis. Vision-based control of a quadrotor for perching on lines. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3130–3136, May 2014.

[2] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, June 1992.

[3] W. J. Wilson, C. C. Williams Hulls, and G. S. Bell. Relative end-effector control using cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation*, 12(5):684–696, Oct 1996.

[4] B. H. Yoshimi and P. K. Allen. Active, uncalibrated visual servoing. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 156–161 vol.1, May 1994.

[5] M. Jagersand, O. Fuentes, and R. Nelson. Experimental evaluation of uncalibrated visual servoing for precision manipulation. In *Proceedings of International Conference on Robotics and Automation*, volume 4, pages 2874–2880 vol.4, April 1997.

[6] Thomas Lampe and Martin Riedmiller. Acquiring visual servoing reaching and grasping skills using neural reinforcement learning. In *IEEE International Joint Conference on Neural Networks (IJCNN 2013)*, Dallas, TX, 2013.

[7] Fereshteh Sadeghi, Alexander Toshev, Eric Jang, and Sergey Levine. Sim2real viewpoint invariant visual servoing by recurrent control. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[8] Fereshteh Sadeghi. Divis: Domain invariant visual servoing for collision-free goal reaching. *CoRR*, abs/1902.05947, 2019.

[9] Ali Ghadirzadeh, Atsuto Maki, Danica Kragic, and Mårten Björkman. Deep predictive policy training using reinforcement learning. *CoRR*, abs/1703.00727, 2017.

[10] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *CoRR*, abs/1504.00702, 2015.

[11] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arxiv:Preprint*, 2018.

[12] S. Lange, M. Riedmiller, and A. Voigtländer. Autonomous reinforcement learning on raw visual input data in a real world application. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, June 2012.

[13] OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub W. Pachocki, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *CoRR*, abs/1808.00177, 2018.

[14] Connor Schenck and Dieter Fox. Visual closed-loop control for pouring liquids. *CoRR*, abs/1610.02610, 2016.

[15] Jan Matas, Stephen James, and Andrew J. Davison. Sim-to-real reinforcement learning for deformable object manipulation. *CoRR*, abs/1806.07851, 2018.

[16] Stephen James, Andrew J. Davison, and Edward Johns. Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task. *CoRR*, abs/1707.02267, 2017.

[17] Avi Singh, Larry Yang, Kristian Hartikainen, Chelsea Finn, and Sergey Levine. End-to-end robotic reinforcement learning without reward engineering. *CoRR*, abs/1904.07854, 2019.

[18] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. *CoRR*, abs/1509.06825, 2015.

[19] Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas A. Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. *CoRR*, abs/1803.09956, 2018.

[20] Leslie Pack Kaelbling. Learning to achieve goals. In *IN PROC. OF IJCAI-93*, pages 1094–1098. Morgan Kaufmann, 1993.

[21] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1312–1320, Lille, France, 07–09 Jul 2015. PMLR.

[22] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *CoRR*, abs/1707.01495, 2017.

[23] Ashvin Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. *CoRR*, abs/1807.04742, 2018.

[24] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. *CoRR*, abs/1610.00696, 2016.

[25] Nal Kalchbrenner, Aäron van den Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. *CoRR*, abs/1610.00527, 2016.

[26] B. Boots, A. Byravan, and D. Fox. Learning predictive models of a depth camera amp; manipulator from raw execution traces. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4021–4028, May 2014.

[27] Arunkumar Byravan and Dieter Fox. Se3-nets: Learning rigid body motion using deep neural networks. *CoRR*, abs/1606.02378, 2016.

[28] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. *CoRR*, abs/1710.11252, 2017.

[29] Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *CoRR*, abs/1804.01523, 2018.

[30] Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. *CoRR*, abs/1710.05268, 2017.

[31] Suraj Nair, Mohammad Babaeizadeh, Chelsea Finn, Sergey Levine, and Vikash Kumar. Time reversal as self-supervision. *CoRR*, abs/1810.01128, 2018.

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

[33] Chris Paxton, Yotam Barnoy, Kapil D. Katyal, Raman Arora, and Gregory D. Hager. Visual robot task planning. *CoRR*, abs/1804.00062, 2018.

[34] Annie Xie, Frederik Ebert, Sergey Levine, and Chelsea Finn. Improvisation through physical understanding: Using novel objects as tools with visual foresight. *CoRR*, abs/1904.05538, 2019.

[35] Sanjay Krishnan, Roy Fox, Ion Stoica, and Ken Goldberg. DDCO: discovery of deep continuous options forrobot learning from demonstrations. *CoRR*, abs/1710.05421, 2017.

[36] Roy Fox, Richard Shin, Sanjay Krishnan, Ken Goldberg, Dawn Song, and Ion Stoica. Parametrized hierarchical procedures for neural programming. In *International Conference on Learning Representations*, 2018.

[37] Danfei Xu, Suraj Nair, Yuke Zhu, Julian Gao, Animesh Garg, Li Fei-Fei, and Silvio Savarese. Neural task programming: Learning to generalize across hierarchical tasks. *CoRR*, abs/1710.01813, 2017.

[38] De-An Huang, Suraj Nair, Danfei Xu, Yuke Zhu, Animesh Garg, Li Fei-Fei, Silvio Savarese, and Juan Carlos Niebles. Neural task graphs: Generalizing to unseen tasks from a single video demonstration. *CoRR*, abs/1807.03480, 2018.

[39] Alexander Neitz, Giambattista Parascandolo, Stefan Bauer, and Bernhard Schölkopf. Adaptive skip intervals: Temporal abstraction for recurrent dynamical models. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9816–9826. Curran Associates, Inc., 2018.

[40] Richard Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

[41] Andrew G. Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(1-2):41–77, January 2003.

[42] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. *CoRR*, abs/1609.05140, 2016.

[43] Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *CoRR*, abs/1805.08296, 2018.

[44] Andrew Levy, Robert Platt, and Kate Saenko. Hierarchical reinforcement learning with hindsight. In *International Conference on Learning Representations*, 2019.

[45] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *CoRR*, abs/1802.06070, 2018.

[46] Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging planning and reinforcement learning. *CoRR*, abs/1906.05253, 2019.

[47] Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart J. Russell, and Pieter Abbeel. Learning plannable representations with causal infogan. *CoRR*, abs/1807.09341, 2018.

[48] Angelina Wang, Thanard Kurutach, Kara Liu, Pieter Abbeel, and Aviv Tamar. Learning robotic manipulation through visual planning and acting. *CoRR*, abs/1905.04411, 2019.

[49] Dinesh Jayaraman, Frederik Ebert, Alexei Efros, and Sergey Levine. Time-agnostic prediction: Predicting predictable video frames. In *International Conference on Learning Representations*, 2019.

[50] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*, pages 5026–5033. IEEE, 2012.

[51] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. *CoRR*, abs/1903.01973, 2019.

[52] Frederik Ebert, Sudeep Dasari, Alex X. Lee, Sergey Levine, and Chelsea Finn. Robustness via retrying: Closed-loop robotic manipulation with self-supervised learning. *CoRR*, abs/1810.03043, 2018.