Predicting Diverse and Plausible State Foresight For Robotic Pushing Tasks

Lingzhi Zhang*
University of Pennsylvania
zlz@seas.upenn.edu

Shenghao Zhou*
University of Pennsylvania
shzhou2@seas.upenn.edu

Jianbo Shi
University of Pennsylvania
jshi@seas.upenn.edu

***Abstract*—Given an environment, humans are able to hallucinate diverse and plausible locations for an object to exist. Is it possible to let a robot learn such hallucination ability as well? If this can be carried out reliably, the robot could use this ability to generate the plausible state foresight just by observing the environment, and potentially leverage the state foresight for planning. In this paper, we study this problem of predicting diverse and plausible state foresight given an environment, which can be categorized as a conditional multimodal prediction task. Many existing approaches leverages Variational Auto-Encoder (VAE) [7] in various vision applications. However, we notice that even though these previous methods have shown they can generate multimodal results, none of them has shown that they can provide a good coverage of solution space for different conditional input, which is necessary for our application. Thus, we propose a novel two-stage model that first learns to unfold solution space of a canonical conditional environment input, and then learns to deform the solution space into an arbitrary environment. Our experiments show that our propose method outperform existing strong baselines in terms of mode coverage and plausibility. Finally, we demonstrate that our predicted state foresight can be used for planning robotic manipulation successfully.**

I. Introduction

Humans are able to hallucinate diverse and plausible locations for objects to be placed in an environment. This gives human the ability to decide what to do as subgoals in imagination. We ask the question: can an intelligent machine also hallucinate diverse and plausible states just by observing the environment visually? If this can be carried out reliably, can we then use the hallucinated state foresight to generate a trajectory plan for robotic pushing tasks?

To tackle this problem, the naive idea is to learn a conditional generative model that predicts multimodal states conditioned on the environment. Variational Auto-Encoders [7] is a popular choice for this purpose, where the Gaussian latent space parameterizes the plausible solutions given the conditional input. This technique has been widely applied to many conditional multimodal prediction tasks, such as image-to-image translation [22, 10, 6, 20], image-to-flow predictions [4, 15], human pose/location hallucinations [18, 16, 12, 9, 21], and so on.

As a starting point, we use several variants of VAE, such as cVAE-GAN [8] and NDiv [13]. The results show that even though these models can produce multimodal states, they can only cover a small portion of the plausible solutions given conditional input. Though previous works demonstrated multimodal modeling capability, they have never shown good coverage of the solution space, which is vital to planning the trajectory of robot arms.

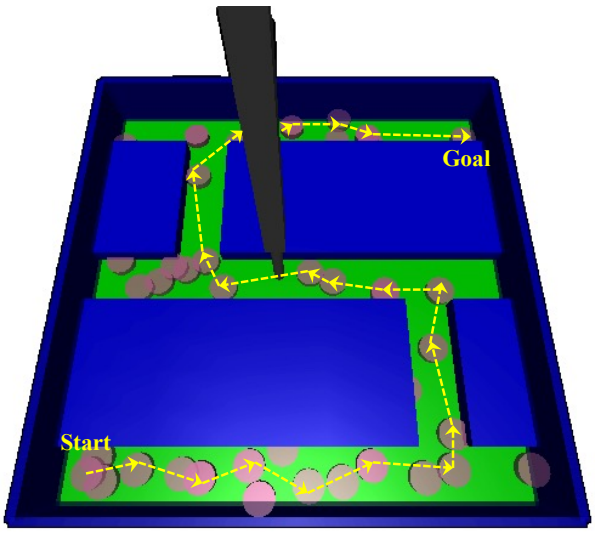

Fig. 1. Given the observation of the environment, our proposed method densely predicts the plausible state foresight (shaded pink disk), and use them for planning between any start and goal states using A-Star shortest path search.

This observation motivates us to think why it is so hard to produce states with good coverage rate in conditional generation. In unconditional generation, deep generative models, such as VAE [7] or GAN [5], are commonly used to unfold a data manifold using a compact latent space. A common issue in these models is mode collapse, which means multiple sampled latent codes correspond to the same output. As a result, only part of output space is covered.

For unconditional generative models, more recent works propose techniques to alleviate mode collapse and thus provide better coverage of the output data distribution. In conditional generation, however, solution space is dependent on the conditional input, and thus the topology of the data distribution varies with the conditional input. Since there are usually many unique conditional inputs in the data, the learning complexity of unfolding multiple data manifolds with different topology simultaneously in a single network could lead to big challenges

*These two authors contributed equally.

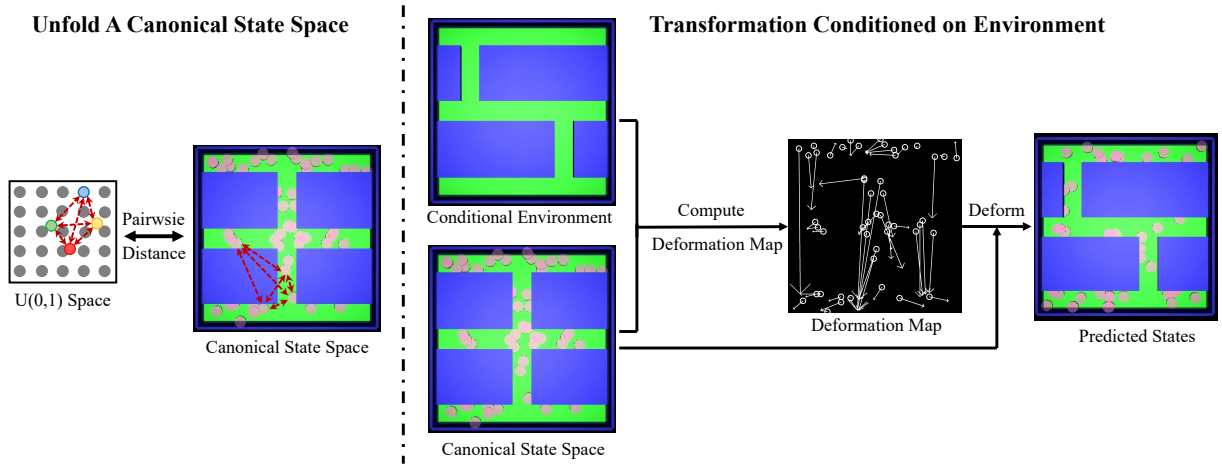

Fig. 2. A schematic diagram of our proposed method. In the **left**, we first learn a generative model to unfold the data manifold of a canonical state space. In the **right**, given a conditioned environment, we learn another network to predict a point-wise deformation map that can transform states from the canonical space into the conditioned environment space.

for the optimization as well as the model capacity.

In this work, our insight is that we can first learn an unconditional generative model to unfold a data distribution for a canonical conditional input, and then learn another network to deform the canonical solutions into many different sets of solutions given different conditional inputs. In this way, we explicitly separate the conditional generative process in two steps, where each step is modeled by an independent network. In our experiment, this simple but effective model design provides sufficiently good coverage of data distribution while maintaining the plausibility. With the ability to produce reasonably good coverage of solutions in various conditional environments, we finally demonstrate that our proposed model can be used to densely sample diverse state foresight given an unseen environment, where the predicted state foresight can be further used to generate plans for the robot arm pushing task.

II. Methods

In this section, we discuss the details about how to learn a generative model to unfold a canonical space, deform the canonical solution to adapt to different environments, and finally do planning using the predicted state foresight.

A. Unfolding A Canonical State Space

Our first step is to learn a generative model to unfold the state space for a canonical environment, as shown in the left of Figure 2. In the canonical environment, two passages in the vertical direction lie in the middle, the average positions of all possible locations. Specifically, we use the normalized diversification (NDiv) [13] to achieve this goal, since it has a nice property to actively explore data manifold with unknown topology and provides good coverage of data distribution. In the NDiv training, the first objective goal is to preserve the normalized pairwise distance of different predicted states with respect to the normalized pairwise distance of the latent variables. Let us denote the latent variable as z, the predict state as $\hat{s}$, and the generative model as $g(\cdot)$. Then, the forward pass can be represented as $\hat{s} = g(z)$ The distance between latent variables is defined as $d_z(z_i, z_j) = ||z_i - z_j||$, and the distance between the predicted states is defined as $d_s(\hat{s}_i, \hat{s}_j) = ||\hat{s}_i - \hat{s}_j||$, where i and j indicate two different samples. Then, the normalized pairwise distance matrices are defined as $D^z_{ij} = \frac{d_z(z_i,z_j)}{\sum_j d_z(z_i,z_j)}$ and $D^s_{ij} = \frac{d_s(\hat{s}_i,\hat{s}_j)}{\sum_j d_s(\hat{s}_i,\hat{s}_j)}$ for latent variable and states, respectively. Finally, the normalized diversification loss is represented as follows.

$$\mathcal{L}_{ndiv}(\hat{s}, z) = \frac{1}{N^2 - N} \sum_{i=1}^{N} \sum_{i \neq j}^{N} max(0, \alpha D^z_{ij} - D^s_{ij}) \quad (1)$$

where α is a hyperparameter. The pairwise distance matrices D^z_{ij} and D^s_{ij} have zero diagonal elements, and thus we average the error by $\frac{1}{N^2-N}$.

Besides normalized diversification loss, an adversarial loss is used to ensure that the predicted outputs fall into the training data distribution. In this case, the adversarial loss can be written as: $\mathcal{L}_{adv} = E_{s \sim p_{\text{data}}(s)} [log(d(s))] + E_{z \sim p(z)} [log(1 - d(g(z)))]$, where we denote discriminator as $d(\cdot)$. Overall, these two losses $\mathcal{L}_{ndiv}$ and $\mathcal{L}_{adv}$ enable us to learn a generative model to actively and safely unfold the canonical state space.

B. Transforming Solutions to Different Environments

Now we have a generative model that can unfold a data manifold, the next step is to learn a model that can transform the solutions from the canonical environment to an arbitrary environment. Let us denote the environment as E, the canonical state space as S_{cano}, and the predicted and ground truth state space for each environment as $\hat{S}_{env}$ and S_{env}, respectively. Note that the state space $S_{cano}/\hat{S}_{env}/S_{env}$ here denotes all states in the environment, so it's different from the single state s or $\hat{s}$ mentioned in the last section. In this case, the single state s is in $\mathcal{R}^2$, and the state space is in $\mathcal{R}^{H \times W}$, where each

single state is encoded by the binary value in the $H \times W$ spatial state space map.

Let us denote the transformation network as $f(\cdot)$. We aim to learn a mapping function $f(E, S_{cano}) \rightarrow \hat{S}_{env}$, where each state in S_{cano} and $\hat{S}_{env}$ has a one-to-one correspondence. During the implementation, we sample 100 states in both S_{cano} and S_{env}, and construct the point-wise deformation flow field for each state from S_{cano} to every S_{env}. Our intuition is that such deformation field should minimize the overall deformation distance. Thus, we use Bipartite matching with Hungarian algorithm to find the one-to-one correspondence between S_{cano} and every S_{env} by minimizing the total Manhattan distance. With the matching correspondence, we construct a 2D deformation flow field F with $H \times W \times 2$ dimension to represent the horizontal and vertical movement at each location to from the canonical space to an arbitrary space.

During training, we use the deformation flow field F computed from Bipartite matching as supervision, and learn this network $f(E, S_{cano}) = \hat{F}$ to predict the deformation flow field using MSE loss. Finally, the predicted state space $\hat{S}_{env}$ is obtained by displacing each state in S_{cano} using $\hat{F}$ in an point-wise manner. The overall schematic diagram is shown in Figure 2.

C. Pushing Through the State Foresight

Given an environment, we now have a whole pipeline to densely sample sufficiently diverse and plausible states. The last step is to generate a plan using the sampled state foresight. In our design, we first construct a graph by treating each sampled state as nodes and constructing edges between nodes with distance below a threshold. With a connected graph, we then run the A* search algorithm to find the shortest path as our planning trajectory between the arbitrary initial and goal state. Finally, we adopt the Cross-Entropy Method (CEM) [1] to search an action between the current state and the next coming up state in the planning trajectory to execute the push manipulation. A visual demonstration of connected graph and search path is shown in Figure 3.

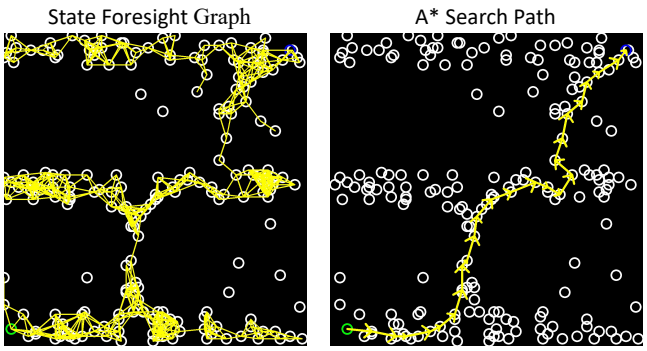

Fig. 3. A visual demonstration of graph construction and search using predicted state foresight.

In our implementation, we notice that the distance threshold for linking the edges between nodes is a slightly tricky hyperparameter to choose. If the threshold is too low, the A* search on the graph with too many edges could possibly generate a path that goes through obstacle. If the threshold is too high, then the edges could be sparse and the graph could be broken into multiple connected components, which prevent searching path from two arbitrary states. Thus, our solution is to incrementally increase the threshold until the A* search can find a path between the initial and goal state. This design ensures there is always a path , while minimizes the risk of having a path going through obstacle.

III. Experiments

In this section, we first discuss the comparison between baselines and our method in the conditional state unfolding. After that, we show how our predicted state foresight can be used for planning robotic push manipulation.

A. Unfolding Conditional State Space

In the conditional state prediction, the goal is to predict diverse and plausible states given an environment input. We compare with two baseline generative models: cVAE-GAN [8] and conditional Normalized Diversification (cNDiv) [13]. The cVAE-GAN consists of a VAE as generator and a discriminator, where the Gaussian latent space in VAE models the plausible solutions conditioned an environment. The cVAE-GAN takes the environment image as input, encode it in a Gaussian latent space, reparametrize the code, and decode to a plausible state. Similarly, the cNDiv first encodes the environment into a latent code, then sample a random variable from uniform space, and finally concatenate them to decode to a plausible state. In the cNDiv training, a diversity loss is applied to enforce the generator actively explore the diverse solution spaces while a discriminator is trained to check the plausibility of predictions in the meantime.

We make comparison with the baselines both qualitatively and quantitatively. The qualitative comparison is shown in Figure 4. Given the environment input, we sample 200 predicted states during inference. Ideally, the sampled points should spread out over the valid regions. However, we observe that the sampled predictions from cVAE-GAN collapse into a compact region. The cNDiv model produce better diversity, but the sampled states also have mode collapse and thus only cover a partial solution space in the given environment. In contrast, our method can produce much more diverse states in the valid region, as shown in the third column of Figure 4.

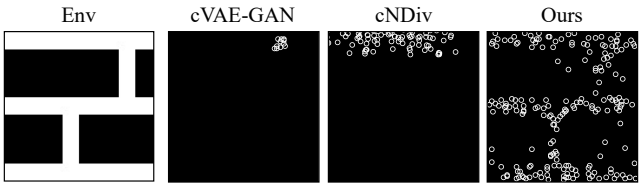

Fig. 4. A visual comparison of different methods on conditional manifold unfolding. The leftmost image is the given environment input, where the white indicates valid region and black indicates the obstacle region.

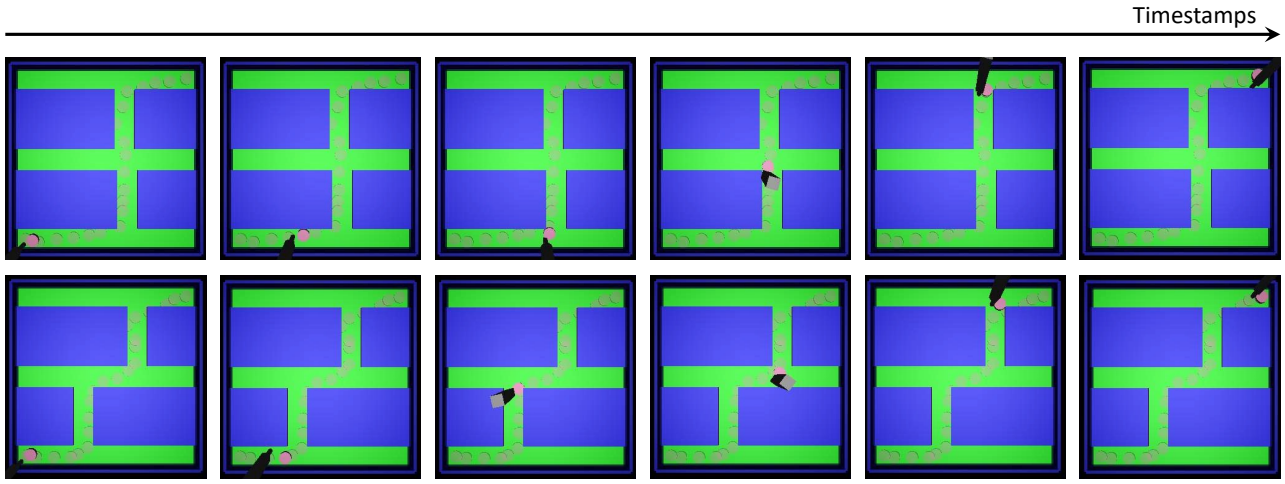

Fig. 5. The first row and second row show two examples of successfully pushing the object from the initial state to the goal state using our predicted state foresight. From left to right, we illustrate a few intermediate timestamps to show the pushing process.

Methods	Coverage (Unique States) ↑	Plausibility ↑
cVAE-GAN [8]	10.25	96.75 %
cNDiv [13]	86.90	51.75 %
Ours	170.65	85.95 %

TABLE I
A QUANTITATIVE COMPARISON. **COVERAGE** SHOWS THE NUMBER OF UNIQUE PREDICTED STATES IN THE VALID REGION BY SAMPLING 200 STATES. **PLAUSIBILITY** INDICATES THE PERCENTAGE OF VALID STATE PREDICTIONS.

For the training data, we render 100 environments with 100 states in each environment. During evaluation, we randomly render 20 new environments and compute the averaged performance among them. In Table I, we show quantitative comparison under two metrics: coverage and plausibility. The Coverage shows the number of unique predicted states within valid region among 200 sampled points in each environment. The Plausibility indicates the percentage of valid state predictions. Note that the valid state is defined as the state inside valid region (non-obstacle region). Note that even though cVAE-GAN has higher plausibility score than ours, it suffers from severe mode collapse and has worst coverage. Overall, we believe that our method demonstrate the best balance in terms of diversity and plausibility in this conditional multimodal prediction task.

B. Robotic Push task

With sufficiently diverse and plausible state foresight predictions, we first construct a graph using the predicted states as nodes and then run A-Star search algorithm to generate the trajectory between the initial and goal states. In Figure 5, we show two successfully robotic pushing cases using our pipeline in the first and second rows. The shaded disks indicate the state foresight on the planning trajectory, where we also denote them as state waypoint. We use CEM to search the low-level robot arm control parameters between current state and the closest future state waypoint for pushing execution. As shown in the figure, even though a few predicted state foresight are not perfectly accurate and are centered slightly inside the obstacle, this has limited effect on executing in this pushing scenario.

IV. RELATED WORK

Our work is related to two lines of research: conditional multimodal prediction and visual foresight for robotics. Recent work have demonstrated conditional multimodal reasoning on various vision tasks, such as conditional multimodal prediction tasks, such as image-to-image translation [22, 10, 6, 20], image-to-flow predictions [4, 15], human pose/location hallucinations [18, 16, 12, 9, 21], and so on. On the other hand, Visual foresight predictions have been used for various robotic tasks. For example, [3] develop a method for combining deep action-conditioned video prediction models with model-predictive control for robotic pushing manipulation. [2] propose a self-supervised model-based approach, which leverages a predictive model that directly predicts the future from raw images, for various robotic pushing scenarios. Visual foresight is not only limited to future state, but also could be affordance or action. For example, [19] propose to learn a neural embedding to predict diverse actions given imagery state input. [17] introduce a new affordance representation that enables the robot to reason about the long-term effects of actions through modeling what actions are afforded in the future, and demonstrate effective performance on robotic grasping tasks. [14] propose to predict affordance segmentation from visual input, and shows that such affordance can effectively boost the RL sample efficiency in indoor navigation and manipulation tasks.

V. CONCLUSION

In this work, we formulate the problem of predicting diverse and plausible state foresight for robotic push manipulation. We propose a novel two-stage model that first unfolds solution space of a canonical environment, and then transform solutions

into an arbitrary environment. In the experiment, the propose method shows the power to sample sufficient diverse state foresight so that they can be used for planning a push trajectory. In the future, we aim to test out our algorithm in more complicated robotic pushing scenarios. In addition, we also plan to study how set transformer [11] can be leveraged for the solution transform in the second step of our model.

REFERENCES

[1] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

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

[3] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017.

[4] Ruohan Gao, Bo Xiong, and Kristen Grauman. Im2flow: Motion hallucination from static images for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5937–5947, 2018.

[5] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[6] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018.

[7] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[8] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016.

[9] Donghoon Lee, Sifei Liu, Jinwei Gu, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Context-aware synthesis and placement of object instances. *arXiv preprint arXiv:1812.02350*, 2018.

[10] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 35–51, 2018.

[11] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pages 3744–3753. PMLR, 2019.

[12] Xueting Li, Sifei Liu, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Putting humans in a scene: Learning affordance in 3d indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12368–12376, 2019.

[13] Shaohui Liu, Xiao Zhang, Jianqiao Wangni, and Jianbo Shi. Normalized diversification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10306–10315, 2019.

[14] Tushar Nagarajan and Kristen Grauman. Learning affordance landscapes forinteraction exploration in 3d environments. *arXiv preprint arXiv:2008.09241*, 2020.

[15] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pages 835–851. Springer, 2016.

[16] Xiaolong Wang, Rohit Girdhar, and Abhinav Gupta. Binge watching: Scaling affordance learning from sitcoms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2596–2605, 2017.

[17] Danfei Xu, Ajay Mandlekar, Roberto Martín-Martín, Yuke Zhu, Silvio Savarese, and Li Fei-Fei. Deep affordance foresight: Planning through what can be done in the future. *arXiv preprint arXiv:2011.08424*, 2020.

[18] Tianfan Xue, Jiajun Wu, Katherine L Bouman, and William T Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. *arXiv preprint arXiv:1607.02586*, 2016.

[19] Lingzhi Zhang, Andong Cao, Rui Li, and Jianbo Shi. Neural embedding for physical manipulations. *arXiv preprint arXiv:1907.06143*, 2019.

[20] Lingzhi Zhang, Jiancong Wang, Yinshuang Xu, Jie Min, Tarmily Wen, James C Gee, and Jianbo Shi. Nested scale-editing for conditional image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5477–5487, 2020.

[21] Lingzhi Zhang, Tarmily Wen, Jie Min, Jiancong Wang, David Han, and Jianbo Shi. Learning object placement by inpainting for compositional data augmentation. In *Computer Vision – ECCV 2020*, 2020.

[22] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *arXiv preprint arXiv:1711.11586*, 2017.