Concept2Robot 2.0: Improving Learning of Manipulation Concepts Using Enhanced Representations

Shivang Singh
University of Texas at Austin
Department of Computer Science
ssingh@cs.utexas.edu

Jie Hie Liao
University of Texas at Austin
Department of Computer Science
jhliao@utexas.edu

***Abstract*—In this work we aim to train a agent that is able to learn manipulation concepts that allow for the agent to map natural language instructions to motion trajectories. More specifically, our aim is to learn single task policies that are able to take language instructions as well as an initial scene image and output output trajectories. Our work builds on previous works in the domain by exploring enhanced visual representations that allow the agent to perform better on the single tasks. We introduce semantic pretraining, in which we pretrain a visual backbone on the semantically rich task of image captioning. We show that this leads to improvement in performance over previous work. We also show the effect of using other such enhanced representations for other domains.**

I. Introduction

Many of the verbs in the language we use today originate from everyday manipulation tasks that humans do on a day to day basis. These verbs or concepts serve as abstractions of complex and diverse set of skills that we have. Furthermore, these words and concepts allow us to generalize to different environments and unseen situations. However, currently robotic agents don't possess this ability.

Previous work in this domain, Concept2Robot, proposes a Robot Learning framework which learns these concepts [1] from human demonstration videos. Manipulation concepts are derived from natural language instructions and their relation to the corresponding human demonstration video. Using videos allows us to mimic how humans learn their manipulation concepts, which is through seeing the mapping language and visual stimulus. For example, a human might learn the correspondence between speech that another human says and visual output of another actor. In this way, the agent is able to generalize to unseen environments and unknown tasks. Furthermore, by using videos to learn manipulation concepts, training futures might be more efficient as they would not require expensive and time consuming demonstration data.

Concept2Robot uses a end to end model which encodes the scene image as well as the natural language instruction and maps it it to a motion trajectory. That motion trajectory is fed into a classifier, which is trained on demonstrations. The confidence of the video classifier is used as a reward function.

However, one of the key limitations of the previous work is that the visual representations that it uses to learn these concepts might be limiting. Previous work uses a ResNet-18 model to encode its environment and simply concatenates this with the input textual embedding in a late fusion approach. Furthermore, since the update in reinforcement learning models is relatively expensive compared to other learning paradigms, this means that the input image embedder doesn't learn low level semantic features that could allow it to reason about the image on the object level. [2]

In this work we propose using an image pretraining scheme that can allow the image backbone to learn representations that pay attention to important semantic and object low level details in the image. VirTex [3, 4] is a pretraining method in which an image model is used as the visual backbone used as the input to image captioning decoder model. The gold standard caption is used to generate loss. This pretraining task is semantically dense allowing it to improve downstream tasks (which are less semantically dense) such as object detection (when used as image backbone to Faster-RCNN) as well as image classification. Another embedding scheme we propose is CLIP [5], which are visual and textual embedding learned from image-text pair matching tasks.

We hypothesize that using this pretraining scheme allows for better performance in tasks which involve multiple objects and deep understanding of the scene image. Additionally, we hypothesize that using VirTex could allow for faster convergence as the image model encodes a lot of the semantically important information needed.

We show that using VirTex as well as other techniques that capture semantic information (CLIP) show improvement over the baseline Concept2Robot model trained with a ResNet-50 image encoder. However, this doesn't reach the reported performance of the baseline model, likely due to training constraints. The training and testing of our models can be found on our GitHub repository.[1]

II. Problem Statement

In our task, we are given an initial state (with the configurations of the robotic arm) and a natural language instruction,

[1] https://github.com/liaojh1998/cross-modal-concept2robot

which indicates how the arm should manipulate the environment. Given this input, our model returns intermediate goal poses (the positions that the robotic arm should take). We use the 20BN-smth-smth dataset, which are series of actions (done by humans in the real world) to guide the training of our model. We describe how we use this dataset in the technical approach.

III. Related Work

A. Concept2Robot

Concept2Robot [1] aims to learn manipulation concepts that link natural language instructions with motor skills. The goal is to learn an agent that can attain manipulation concepts that it can use to generate motion trajectories. As opposed to using simulated trajectories to learn concepts, model learns associations between natural language instructions and motor skills directly from real life demonstrations. The model consists of a language encoder, which takes the input language instruction and generates an output language encoding. Similarly, the model also has a visual encoder which generates a visual encoding. These encodings are concatenated and fed to a CNN which maps them to an output goal pose and a motion trajectory. This motion trajectory gets translated into a video, which then fed to a video classifier trained on demonstrations.[6] The confidence of the video classifier is used a reward function for the actor-critic network. However, one of the limitations of this work, that our work wishes to address, is that the baseline visual encoder doesn't fully exploit all of the visual information in the scene image. The visual embedding (which is gotten using a ResNet-18 model) doesn't encode much information about low level semantic features, which would allow the model to learn to reason about individual objects in the scene. Our approach attempts to address this problem by using a visual model, pretrained on a semantically dense task allowing the representation encode semantic differences within the scene. [3]

B. Visual and Language Grounding

Tasks involving that involve bridging the gap between vision and language have gained in popularity in recent years.

1) Image Captioning: One of the preeminent tasks in the multimodal vision and language domain is the class of image captioning. [7, 8] Image captioning involves a deep understanding of the semantic knowledge encoded within the baseline image. [3] shows that these models learn better image representations that allow for the model to reason about the underlying baseline model. They show that pretraining on the image captioning task can have great performance increase on unimodal image based tasks such as object detection when used as the image backbone. [9]

2) Image and Language Tasks: Other tasks in the visual and language domain include Visual Question Answering (VQA), Natural Language for Visual Reasoning (NLVR), Visual Entailment Dataset (SNLI-VE), and Visual Commonsense Reasoning (VCR). [10, 11, 12, 13] These tasks all involve a visual and text (multimodal) input. These tasks involve complex reasoning over the input in which the underlying model needs to understand how the two modalities relate to each other. The model must, for example, understand which part of the image to attend to based on the input text.

3) Robot Image and Language Learning: As opposed to other image and language learning tasks, robot image and language learning tasks involve a complex understanding of the image. Agents must be able to reason about depth and positioning to a great degree. Furthermore, the reward and the subsequent loss provided by the environment is relatively expensive to compute when compared to other image and language tasks. Furthermore, the agent must learn a more embodied representation of the image and language input and the action space is much larger than the traditional image and language tasks. For example, Jiang et al [14] examines the compositional nature of language and uses it to learn hierarchical manipulation task abstractions. It learns to break down compositional instructions into hierarchical instruction sets, which can be used to carry out a policy. Similarly, Shu et al [15] also looks at the skill acquisition in multi-task reinforcement learning settings. It learns modules that look at the compositional nature of language and outputs a hierarchy of instructions from a universe of instructions. Thus it learns the compositional nature of language as well as concepts regarding manipulation. However, this work focuses on a small discrete action space. Unlike these works, our work focuses on a continuous action space.

C. Learning from Demonstrations

Learning from Demonstration enables an agent to learn from expert demonstrations. [16] Learning from Demonstrations reduces the human effort in designing reward functions and motion trajectories. Jiang et al. and Shu et al.[15, 14] had to design specific reward functions for their works. Recent advances in large scale image datasets have lead to leverage visual data for models that learn from demonstrations. For example Fu et al [17] looks at how to do inverse reinforcement learning without expert demonstrations and labelling. They learn a policy when a large number of goal states are available. Gupta et al [18] on the other hand uses demonstration data to learn long horizon tasks, which require extensive planning. This work however focuses on learning from video demonstrations, which are inherently noisy.

IV. Data

A. Human Demonstration Data

For our work we use the 20bn-sth sth dataset of human video recordings. [6] It contains over 108,000 videos of 174 different tasks, which are roughly 3-6 seconds long. The labels of each of the videos consist of text based descriptions of the action taking place in the video. For example, one label is "Moving something closer to something", which a task that we focus on. We focus on a select number tasks from this dataset. We outline these tasks in our experiments section.

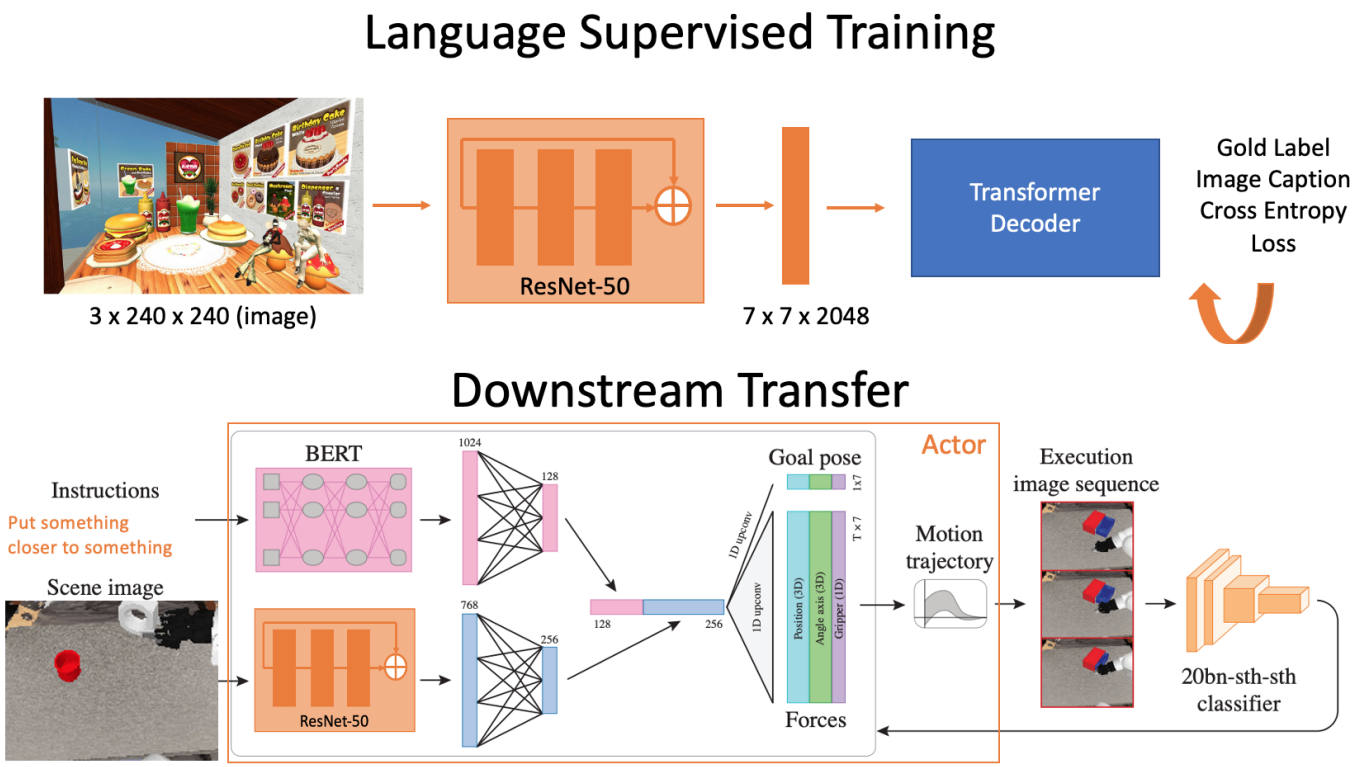

Fig. 1: This figure illustrates the architecture of our approach. In order to generate image feature representations that capture more semantically important and low level features of the image, the ResNet-50 model, pretrained on the image caption task, is used.

B. Environment

We use a PyBullet environment to simulate each environment associated with each of our task. [19] Our robot is a simulated 7-DOF Franka Panda arm with a 2F-285 gripper. A camera is mounted to capture the environment state and a RGB camera is used to capture a downsampled 120 x 160 image for each frame in our execution history.

C. Language Supervised Training Data

The language supervised pre-training of our modified vision model uses the COCO dataset which contains over 200,000 labeled images. These labeled images contain both images as well as their captions. [20] These are used as the input into the language supervised training methodology that we outline in the technical approach. The captions are used to generate loss functions allowing the visual encoder backbone to learn semantic meaning.

V. Technical Approach

A. Overview

We intend to improve the performance of the single-task approach for 4 tasks defined by the Concept2Robot project in the PyBullet environment. These tasks are:

1) Task 42: Moving something closer to something.
2) Task 104: Putting something behind something.
3) Task 93: Pushing something from left to right.
4) Task 86: Pulling something from left to right.

We build our approach on top of the single-task approach proposed in Concept2Robot, in which an agent is trained to solve a single task. An agent in Concept2Robot has 2 components, an actor and a critic. The actor describes the action to take for the robot arm while the critic aims to stabilize the training of the actor. The actor implements the policy network, which outputs the parameters of an open-loop motion trajectory that, when executed by the robot, achieves the manipulation task in the current environment. The motion trajectory is represented by a second-order dynamical system of the form:

$$\ddot{y}(t) = k_p(g - y(t)) - k_d\ \dot{y}(t) + f(t)$$

where $y, \dot{y}, \ddot{y}$ are the position, velocity, and acceleration of the end-effector, g is the goal pose, and f are the additional forces at each timestep that modify the shape and velocity profile of the trajectory. k_p and k_d are standard PD control gains. The motion trajectory operates in a 7-dimensional space: 3D position and orientation of the end effector and 1D movement of the two-fingered gripper. The actor will predict the goal pose and the forces of the trajectory in this 7-dimensional space for the robot arm to move into the pose required to complete the task.

The reward function for a task is substituted by a video classifier that is trained to classify actions in the *Sth Sth* videos.

The 4 tasks we chose are tasks that already exist in the *Sth Sth* dataset. The classifier uses 3D convolutions to learn spatio-temporal relationships in videos. During training of the policy network, the output trajectory is executed open-loop by the robot and rendered into a video. The video is then scored by the video classifier for the reward.

The actor and critic are neural networks that used ResNet18 to extract features from the visual modality and BERT to extract features from the text modality.[21] The features of each feature extractor are concatenated together as a late fusion representation of the overall state. In the actor, the representation is fed through an MLP to predict the goal pose and the forces. In the critic, the representation is fed through an MLP to approximate the Q-value of the task.

To solve the Contextual Bandit problem, Deep Deterministic Policy Gradients (DDPG) and the Cross Entropy Method (CEM) are combined to jointly optimize the actor and critic in the agent.[22, 23] During training, trajectories and their rewards are stored in a replay buffer. A sample of trajectories in the replay buffer is used to update the critic and the actor. The critic loss is defined as $\mathcal{L}_c = ||r - Q(s,a)||^2$ and the actor loss is defined as $\mathcal{L}_a = -Q(s, \text{Actor}(s))$, where $\text{Actor}(s)$ is the output of the actor network. Note that the weights of the critic network are not updated when optimizing the loss of the actor network. The critic network is trained to directly approximate the Q value. The critic network is updated first before updating the actor network.

In this project, we aim to improve the performance of the agent by changing the underlying architecture of the feature extractors. We changed the feature extractor of the visual modality from ResNet18 to ResNet50 for an experiment and to VirTex for another experiment.[24] We also used visual transformer feature extractors and text transformer feature extractors of the CLIP model instead of ResNet18 and BERT for another experiment. [21] The architecture of both actor and critic are changed in our case. So, when we say we changed from ResNet18 to ResNet50, both the actor and the critic used ResNet50 for feature extraction. An example of this change is shown in Figure 1. [24]

B. VirTex

Karan et. al [3] introduces an image representation learning methodology in which a ResNet-50 model is used to generate image representations useful for semantically dense tasks such as image captioning (VirTex). When the VirTex model is used to generate visual representations for downstream image tasks such as object detection and even image classification it leads to outperformance over using baseline models. For example, when VirTex is used as the visual backbone of the Faster-RCNN model, it is able to outperform the baseline ResNet model. [24] One of the limitations of the current Concept2Robot model is that it doesn't fully exploit the visual input. The visual features it generates are using a pretrained ResNet model that is trained on ImageNet. Furthermore, since the reinforcement learning model doesn't train for many iterations, it doesn't learn low level semantic features that might be important for encoding semantic meaning in the image. To overcome the limitations of the baseline, the VirTex ResNet-50 model was used to encode image features. The VirTex model as shown in Figure 1 was pretrained on a language supervised training task. [24] VirTex was used as the image backbone for the image captioning task, where it was used as the input embedding for a language decoder model. The gold label image caption was used to generate the resulting cross entropy loss. This pretrained model was trained on the COCO-image dataset. [20] The resulting ResNet-50 weights were transferred to the downstream model, which in this case is the architecture from Concept2Robot. [1, 24]

C. CLIP

CLIP [5], which stands for Connecting Text and Images, is a form of pre-training done on features extracted from a visual feature extractor and text feature extractor. The model aims to fine-tune both extractors through a image-text pair matching task: given an image, predict which out of a set of 32,768 randomly sampled text snippets, was actually paired with it in the dataset.

D. Evaluation Metric

The reward that the robot receives from the video classification is a proxy objective that may not directly reflect how successful a policy is with respect to its task. Therefore, the authors of the original Concept2Robot manually defined task-specific success metrics to evaluate the policies. For example, for the task "putting sth behind sth", the task is successfully completed if the robot managed to move the grasped object behind a stationary object. The original paper reported the average success rate over 100 episodes. We report the average success rate over 500 episodes.

VI. Experiments

A. Technical Setup

We extended upon the official repository of Concept2Robot [1].[2] In the original Concept2Robot repository, as shown in Figure 1, the actor and critic both used images of dimensions 120x160x3 and a BERT embedding of dimensions 1024 as input to the network. The head of a pre-trained ResNet18 is replaced with a 2D convolution that takes in 512 features and output 256 features. The visual feature extractor eventually outputs 256 features through an MLP. The text feature extractor eventually outputs 128 features through an MLP. These are then concatenated as a late fusion representation to be put through another MLP to output action information. Note that the actor and the critic both used ResNet18, but the weights of the ResNet18 is not shared between the two, and the weights of the pre-trained network are fine-tuned as the actor and critic are trained.

In this project, for ResNet50 and VirTex, we replaced the head of the pre-trained model with a 2D convolution that takes in 2048 latent features and output 256 features. The inputs to

[2]https://github.com/stanford-iprl-lab/Concept2Robot

the actor and critic are the same as prior work. All other MLP architectures are also the same as prior work. The pre-trained weights of the extractor are also not frozen, so they are also fine-tuned as the actor and critic are trained.

Since the CLIP model replaces both the visual feature extractor and the text feature extractor of Concept2Robot, the input to the actor and critic will be slightly different. The actor and critic takes in images of dimensions 120x160x3 and tokenized vocabulary indices of the task description that is tokenized according to the CLIP model. The CLIP model will take the images and output 512 image features and take the task description and output 512 text features. We then feed the 512 image features through an MLP to output 256 features and the 512 text features through an MLP to output 128 features. Again, these are concatenated as a late fusion representation that is put through another MLP to output action information. We used the ViT-B/32 CLIP model that used a transformer architecture for both visual feature extraction and text feature extraction. Since transformers are much bigger than ResNets, we froze the weights of the CLIP feature extractors and only trained the weights of the MLPs.

B. Hyperparameters

We used the Adam optimizer for training both actor and critic with learning rate of 1e-5 and 5e-5 respectively. We train each model on each task for 10,000 episodes. There are 49 timesteps, or 49 frames of the environment and the robot, in an episode. The first 1,000 episodes stored trajectories of random actions instead of actions of the actor into the replay buffer, and the latter episodes stored trajectories of the actor into the replay buffer. For every episode after the first 1,000, we randomly sampled 64 trajectories from the replay buffer to update the critic and actor. A trajectory is defined to have a maximum of 49 timesteps, so the actor is defined to output the goal pose and 49 timesteps of forces, which are a total of 50x7 values. We also experimented with actors that output only the goal pose (with forces) for Task 42. We evaluate the model for 100 validation episodes after every 1,000 train episodes, and chose the model with the highest validation success rate as the model to use for the final testing session. We tested for 500 episodes in the final testing session. All other hyperparameters we did not mention are the same as the hyperparameters in prior work. We trained all models on a NVIDIA Tesla K80, which has about 12 GB of graphics memory, and it generally took about 12 hours to train one model for one task.

C. Results

Our results are summarized in Table I. In general, we found features of ResNet50 to perform better than ResNet18, VirTex to perform better than ResNet50, and CLIP to perform better than VirTex.

In the original Concept2Robot, single-task agents are usually trained for weeks for 100,000 episodes, so we believe that it is expected for their agents that use ResNet18 to perform better than ours, which are only trained for 10,000 episodes. However, this is only the case for Task 42, and we found our proposed feature extractors to do much better in other tasks with much less iterations of training.

TABLE I: The success rate (over 500 episodes) of each different type of model architecture. ResNet18 results are from prior work. ResNet50, VirTex, and CLIP are tested by us. Each model is tested on the iteration they achieved the highest validation success rate on. An "(F)" indicates that the model outputted the goal pose and the forces for 49 timesteps. A model without the "(F)" indicates that the model only outputted the goal pose. The descriptions of these tasks can be found in section V-A.

Model	Task 42	Task 104	Task 93	Task 86
ResNet18	87	43	92	**89**
ResNet18 (F)	**88**	**51**	**95**	88
ResNet50	62.8	-	-	-
VirTex	**67.6**	-	-	-
ResNet50 (F)	49	55.6	98.6	**100**
VirTex (F)	64	59.8	91.8	80.4
CLIP (F)	**74.8**	**63.2**	**99.8**	99

For Task 42, we found our actors that output the goal pose only to do better than the actors that output the goal pose and the forces. We suspect that this is because it is much easier to train a model to output only 7 values than to train a model that output 50x7 values. If we trained our model for longer than 10,000 episodes, we will likely see an improvement from outputting goal pose with forces as observed from prior work.

As we expected from VirTex that used pre-training with Image Captioning, even though both VirTex had the same ResNet50 model architecture, VirTex features had stronger semantic information that allowed the actor to perform better for Task 42 and 104, which are "put sth closer to sth" and "put sth behind sth". We also found VirTex to have a faster convergence than ResNet50 through validation success rate over time. However, VirTex seemed to have adversary features that are bad for Tasks 93 and 86, which are "push sth from left to right" and "pull sth from left to right".

Lastly, we found CLIP to do better and converge faster in success rate than other models for all tasks. Success rate that generally took about 6000 to 7000 episodes of training to reach for VirTex and ResNet50 generally took about 3000 episodes of training to reach for CLIP. Thus, we believe features of the transformer architecture will usually perform better than ResNet features for other robot learning tasks as well.

VII. Conclusion

Learning manipulation concepts from demonstrations is a important task as it can allow for agents that are able to better generalize to unseen environments and tasks. Previous work established a model which took in input natural language instructions and input scene images and translated them into motion trajectories. In this work we proposed two methods to improve on this previous work. Firstly, we replaced the underlying image encoder (ResNet-18) with an image encoder that is pretrained on the semantically dense task of image captioning. Secondly, we sought to enhance the textual embeddings of the

input model, by using a transformer that is grounded using visual tasks. Using these two changes, we show significant improvement on a set of single tasks. There are many future directions that the current work could go. One drawback of our combined model is that it continues to rely on the late fusion approach used in the baseline. However, if a cross modal attention model was used, the model would be able to use the natural language instruction to focus on parts of the image instead of relying on a model that can encode semantic features. Another drawback of the approach is that the approach is limited to learning manipulation concepts through a video demonstration dataset. These means that there are a finite set of tasks and manipulation concepts that the agent can learn as we require labels. However, the pretrained classifier could be extended to instruction videos where nothing is labeled. The entire model could be pretrained in a self-supervised learning method in this way. The video and text (gotten from what the instructor is speaking) could be used a train a classifier. And a larger universe of environments and tasks could be done in this way. However, there could be large computational costs to train the self-supervised model. Looking at these future steps could improve concept acquisition for the field of Robot Learning.

ACKNOWLEDGEMENTS

We would like to thank Toki Mimigatsu for providing us with the missing "something something" video database and classifier. We would also like to thank Professor Yuke Zhu for supervising us on this project.

REFERENCES

[1] Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, July 2020.

[2] Weixin Liang, James Zou, and Zhou Yu. ALICE: active learning with contrastive natural language explanations. *CoRR*, abs/2009.10259, 2020. URL https://arxiv.org/abs/2009.10259.

[3] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. *CoRR*, abs/2006.06666, 2020. URL https://arxiv.org/abs/2006.06666.

[4] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai, 2021.

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[6] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense.

[7] Oriol Vinyals, Alexander Toshev, Samy Bengio, and D. Erhan. Show and tell: A neural image caption generator. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015.

[8] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. In *NeurIPS*, 2019.

[9] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015. URL http://arxiv.org/abs/1504.08083.

[10] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. Vqa: Visual question answering. *International Journal of Computer Vision*, 123:4–31, 2015.

[11] Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. A corpus of natural language for visual reasoning. In *ACL*, 2017.

[12] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *ArXiv*, abs/1901.06706, 2019.

[13] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6713–6724, 2019.

[14] Yiding Jiang, Shixiang Gu, Kevin Murphy, and Chelsea Finn. Language as an abstraction for hierarchical deep reinforcement learning. *CoRR*, abs/1906.07343, 2019. URL http://arxiv.org/abs/1906.07343.

[15] Tianmin Shu, Caiming Xiong, and Richard Socher. Hierarchical and interpretable skill acquisition in multi-task reinforcement learning. *CoRR*, abs/1712.07294, 2017. URL http://arxiv.org/abs/1712.07294.

[16] Brenna Argall, S. Chernova, Manuela M. Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics Auton. Syst.*, 57:469–483, 2009.

[17] Justin Fu, Avi Singh, Dibya Ghosh, Larry Yang, and Sergey Levine. Variational inverse control with events: A general framework for data-driven reward definition. *CoRR*, abs/1805.11686, 2018. URL http://arxiv.org/abs/1805.11686.

[18] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *CoRR*, abs/1910.11956, 2019. URL http://arxiv.org/abs/1910.11956.

[19] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021.

[20] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

[22] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Manfred Otto Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2016.

[23] Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-Carlo Simulation (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2004. ISBN 038721240X.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.