Transformer Policy Study for Imitation Learning

Arjun Somayazulu*, Sharachchandra Bhat†,
*Department of Computer Science, University of Texas at Austin
†Department of Electrical & Computer Engineering, University of Texas at Austin

***Abstract*—Imitation Learning is a paradigm of robot learning in which agents learn behavior through observation. Behavioral Cloning (BC) is one popular form of Imitation Learning in which a stochastic policy is learned from labeled demonstration trajectories in a supervised fashion. Given the vast diversity of observation modalities that are involved in robotic tasks, encoding and combining these disparate sources as well as capturing temporal relationships in trajectories is key to learning a good policy in BC. We evaluate several methods for encoding these observation sources and capturing temporal relationships in the policy network for Behavioral Cloning based on a Transformer Encoder architecture that incorporate temporal self-attention mechanisms. We evaluate the models on three tasks in the Robomimic suite: Lift, Can, and Square. For each task, we consider both the low-dimensional and image datasets. Our Transformer Encoder policy network performs close to SOTA algorithms that use an LSTM policy, while only attending over only a fixed size history. Similarly, the Transformer-based observation featurization schemes fail to outperform existing approaches, highlighting the deficiency of Transformer-based vision models when used without pre-training and extensive hyper-parameter fine-tuning.**

I. Introduction

Traditional Reinforcement Learning (RL) approaches towards robot manipulation tasks require carefully specified reward functions. If the reward is misspecified or too sparse, it can be difficult for learning to proceed properly. Furthermore, RL approaches towards robot manipulation tasks involve extensive interaction with the environment, which is often expensive in both computation and time.

Imitation Learning is an alternative paradigm of robot learning in which an expert provides some form of supervision to the model. Behavioral Cloning is one main form of Imitation Learning in which this supervision comes in the form of labeled demonstrations. For episodic tasks we consider these expert demonstrations as labeled trajectories, where each trajectory is given as a sequence of observations and actions, $\{(o_t, a_t)\}$ for $1 \leq t \leq T$. These labeled trajectories permit fully-supervised approaches for learning a policy for an MDP. Furthermore, using an end-to-end model to perform this fully-supervised learning allows us great flexibility in choosing how best to represent the policy.

Robot learning problems are often multi-modal in nature, and several approaches for Behavioral Cloning have used Transformer-based models to represent the policy network in these multi-modal settings. However, there is little work detailing how different methods of encoding the sensor signals affects the ability of the network to learn a good policy. We plan to approach this problem by evaluating the performance of several Transformer Encoder based policy models on a suite of behavioral cloning robot manipulation tasks. In particular, because the tasks we are considering are more simplistic than prior problems that we discuss in Section II, we propose a simpler architecture based off a Transformer Encoder to represent our stochastic policy. We implement several modifications to this base architecture, including ensuring causality of prediction and fixed-length history, without significantly increasing the complexity of the model or approach.

II. Related Work

There are several important related works that are relevant to our general problem. For the problem of one-shot visual imitation, [3] develops a Transformer-based model which conditions action generation on a fixed-size featurized expert demonstration video context. For the problem of imitation learning from dual-arm robot manipulation, [5] uses extensive featurization of the signal prior to input into a Transformer encoder, which involves gaze estimation and positional embedding modules.

For the task of imitation learning with Transformer-based sensor fusion for autonomous driving, [2] and [10] use raw LiDAR point-cloud and RGB signals, and employ Transformers on several scales of intermediate feature maps to obtain a compact, global, fixed length featurized signal. [1] takes the opposite approach for natural-language guided robot trajectory reshaping, obtaining a highly compact feature representation *prior* to being input to a Transformer, which subsequently employs a cross-modal attention mechanism within the encoder module itself. These studies are relevant to our project as they demonstrate several different approaches to signal featurization given disparate data modalities (text, images, point clouds, proprioceptive information, etc.).

We also highlight that most approaches fall under two paradigms: In the first, each modality is encoded independently to its featurized representation, and these embeddings are fed to a Transformer which computes temporal attention over the trajectory sequence. In the second paradigm, Transformers are used both to featurize the input modalities as well as combine them using cross-modal attention mechanisms, prior to temporal attention over the entire trajectory sequence. Exploring the performance of these two approaches through empirical results on simple robot manipulation tasks can help us understand the advantages and drawbacks of each of these approaches.

Because the tasks we are considering are relatively more simplistic and shorter-horizon compared to the tasks for which

TABLE I: Task trajectory lengths

Task:	Lift	Square	Can
Min Trajectory Length:	36	82	107
Max Trajectory Length:	A	B	C

these approaches were designed, we implement a Transformer-Encoder stochastic policy network with a few modifications (as discussed in Section IV). In particular, our Transformer encoder based approach uses a fixed-horizon history over which it attends, compared to LSTM policy networks that have infinite-length history horizons.

III. Data

We will be performing our experiments in Robomimic [8], a large-scale framework and collection of datasets and algorithms designed for robot learning tasks. The framework uses the MuJoCo physics engine as its backend, and contains implementations of several Imitation Learning algorithms, including Behavioral Cloning.

For our Behavioral Cloning experiments, we consider the Lift, Can, and Square tasks from the ”robomimic v0.1” collection of datasets. In particular, we use the Proficient-Human dataset for each of these tasks, which consists of trajectories captured by one human operator via the RoboTurk platform [7] For each of these tasks, we consider both the ”low-dim” and ”image” datasets. The ”low-dim” dataset consists of a vector of proprioceptive information from the simulated arm. This proprioceptive information consists of end-effector position, end-effector quaternion values, gripper position, and object pose/position values.

The image dataset includes of visual observations from RGB cameras mounted on the robot arm. These consist of an ”eye-in-hand” image mounted on the end-effector, as well as an ”agent view” image that views the arm from a disembodied perspective. Each of these datasets consist of 200 labeled trajectories; we train on 180 trajectories and use 20 held out trajectories for validation. Table I shows the minimum and maximum lengths of all labeled trajectories for the three tasks we consider.

IV. Methods

For each task, we are given a dataset of trajectories $\mathcal{D} = \{(o_t^{(i)}, a_t^{(i)})\}$, where the observation at each timestep t in trajectory i $o_t^{(i)}$ consists of an RGB image captured from the over-the-shoulder and agent view cameras (only in the image dataset), as well as a low-dimensional vector of proprioceptive data from the robot arm's joints and end-effector. We discuss our methods for Transformer-based featurization of the visual modalities and Transformer-based stochastic policy representation below.

A. Observation Featurization

We evaluate several Transformer-encoder observation featurization schemes in order to evaluate the effect of self-attention on each visual modality. In particular, we evaluate

Fig. 1: Vision Transformer architecture

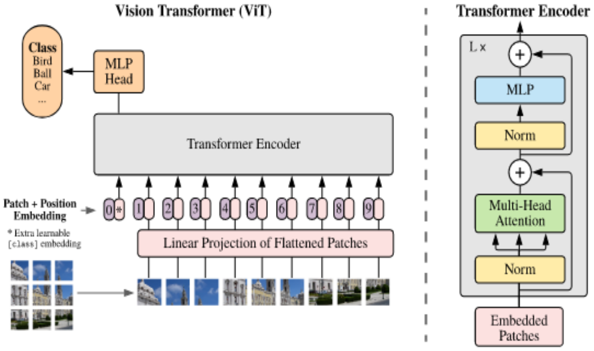
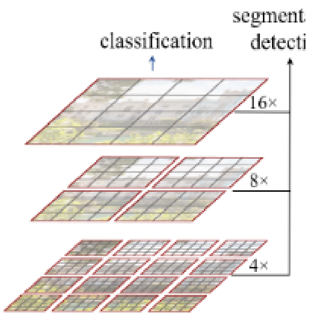


Fig. 2: Swin Transformer architecture



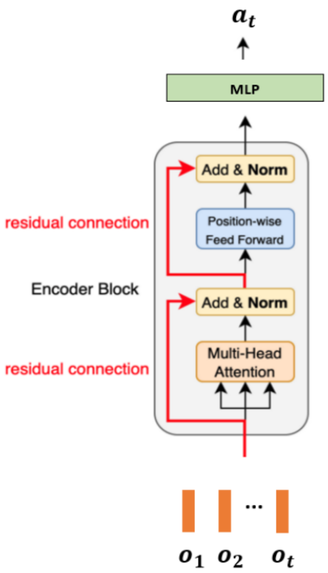
(a) Swin Transformer (ours)

the Vision Transformer architecture introduced in [4], and the Swin Transformer introduced in [6]. For both methods, we upsample the size of the RGB camera image to 224x224 pixels in order to ensure consistency in input.

1) Vision Transformer: The Vision Transformer (Figure 1) is based off the Transformer Encoder architecture proposed in [11]. The input image is divided into 16x16 patches, which are then flattened and fed through linear projections. This sequence is fed to a Transformer Encoder, and the final attended sequence is used as our encoded visual observation vector.

2) Swin Transformer: The Swin Transformer (Figure 2) is similar to the Vision Transformer architecture, but it incorporates spatial inductive bias. Self-attention is computed and aggregated over local pixel regions in the image, and this process is repeated at progressively coarser spatial scales until a final attended representation is computed, which is used in downstream tasks. We flatten the output feature at the last spatial scale and use that vector as our encoded visual observation feature. For both the Vision Transformer and the Swin Transformer, we use the PyTorch implementation [9] in our experiments.

Fig. 3: Transformer Policy Network (ours)



B. *Transformer-Encoder Policy Network*

We also evaluate several models based on the Transformer-Encoder as the stochastic policy network in Behavioral Cloning. Figure 3 shows a diagram of our model.

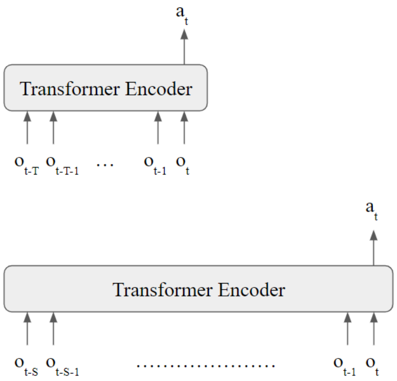
We compute featurized observations using a visual encoder (ResNet) and concatenate it with our low-dimensional proprioceptive observations at each time-step in the trajectory. The assembled featurized observation trajectory is masked to ensure causality of prediction, and input to a Transformer Encoder model. The Transformer Encoder architecture closely follows the proposed architecture in [11]. The Transformer Encoder produces the initial attended feature sequence. This sequence is fed through N Transformer Encoder blocks to produce the final attended sequence. An MLP following the final Encoder takes in the attended feature sequence and decodes the output action value.

We do not use the observation featurization schemes outlined in the previous section here in order to standardize comparison against the baseline models (see Section V). We instead use the default ResNet model available in Robomimic to serve as the feature encoder for our visual observation sources (the two camera views). The encoded output of the ResNet18 is flattened and concatenated with the proprioceptive information to form our raw featurized observations. This is then passed through an MLP to create a low-dimensional observation encoding, which also mixes information between the two modalities.

1) Transformer Input Sequence Length: An architecture choice that affects the performance of the Imitation Learning model is observed to be the maximum length of the input sequence to the transformer encoder. It is well known that access to a sequence of observations to a policy network provides an advantage over a singe time step observation. The intermediate layers of neural policy network act as a state-estimator that allows the later layers to devise a control action assuming the dynamics is Markovian. It can be imagined that the length of the input sequence required to accomplish a goal depends on the task and environment at hand. If the task involves information observed at the beginning, keeping hold of earlier observations is crucial for task success. If the environment layout causes occlusions during robot motion, then too a larger history maybe required.

On the other hand, having an unnecessarily large sequence might make the learning data hungry and with imitation learning demonstrations, training-data is often difficult to come by. Moreover the attention mechanism of the transformer encoder needs to learn to avoid utilizing tokens from earlier irrelevant time instances. This can make learning slower. Therefore we empirically analyse the effect of varying input sequence length on the success rates of different Imitation Learning tasks. We hope to shed light on the relation between complexity of tasks and richness in observations required to accomplish them.

Fig. 4: Difference in sequence length as a model architecture choice

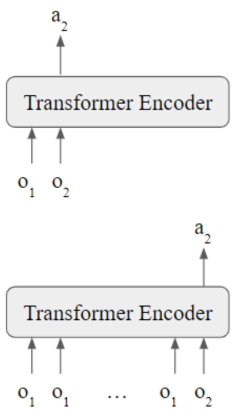

a_t
Transformer Encoder
o_t-T o_t-T-1 ... o_t-1 o_t
a_t
Transformer Encoder
o_t-S o_t-S-1 .... o_t-1 o_t

2) Transformer Input Format: LSTMs recursively processes inputs and hence length of input sequence as an architecture choice is a moot question. However retention of information from earlier parts of long sequences is not necessarily good. The Transformer encoder can work with arbitrary length inputs as the time dimension is just an extra dimension of the input tensor. Once a maximum length is fixed for the input, one could choose a shorter sequence as input particularly at the beginning of the task when fewer past observations are available.

The other option is to pad the input with the first observation multiple times to ensure that the maximum input sequence length is maintained. We aim to test if this architecture choice affects the performance of the policy network.

The two choices are depicted in Fig. 5

Fig. 5: Difference in input format as a model architecture choice



V. Experiments

A. *Observation Featurization*

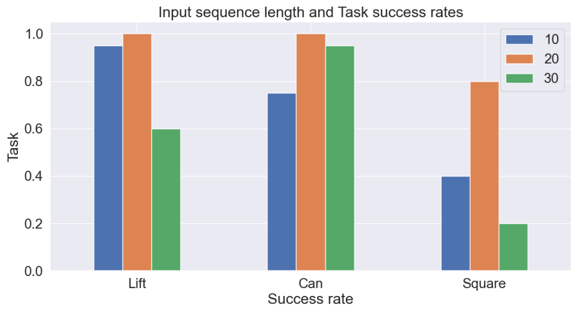
We evaluate the Swin Transformer and Vision Transformer for featurizing the agent view and eye-in-hand images on three tasks: Lift, Square, and Can. We compare the performance of these visual encoder models against a baseline visual encoder, the ResNet18 model. All models are trained from initialization (no pre-training). For consistency, all experiments use a policy network represented by an LSTM. The results on the image dataset for each of the three tasks is presented in Table III.

We can see that for all three tasks we considered, both Transformer models underperformed the ResNet18 encoder model. Table III displays the performance of Transformer-based models compared to the ResNet model. Though we tried adjusting the learning rate and schedulers to to guide the Transformer model's learning, the models we introduced consistently performed poorly on the Can and Square tasks. However, we can see from our validation loss plots that the models approach the same loss as the ResNet encoder for all three tasks. Furthermore, it should be noted that the Swin Transformer consistently outperforms the Vision Transformer (ViT) on all three tasks. This may be due to the spatial inductive bias that is present in the Swin Transformer, which better suits image modality data compared to the spatially invariant patch-based attention mechanism that is employed in the Vision Transformer.

B. *Transformer Policy Network*

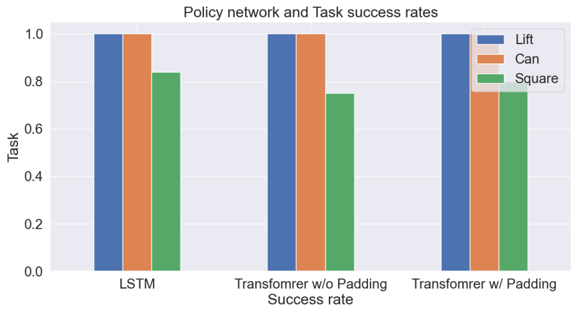
1) Transformer Input Sequence Length: The success rates of three imitation learning tasks Lift, Can and Square were compared by training the transformer policy network with input sequence lengths of 10, 20 and 30. All other parameters were fixed. It was observed that the highest success rate for all the tasks was achieved for the intermediate input sequence length. The decrease in performance with the longer sequences can be attributed to difficulty in learning with limited demonstration data.

Fig. 6: Effect of input sequence length on task performance



2) Transformer Input Format: Here we compare the task performance with and without padding of input sequence to the transformer policy network. They are benchmarked against an LSTM backbone [8]. We observe that the choice of padding does not affect the performance and compares to the benchmark.

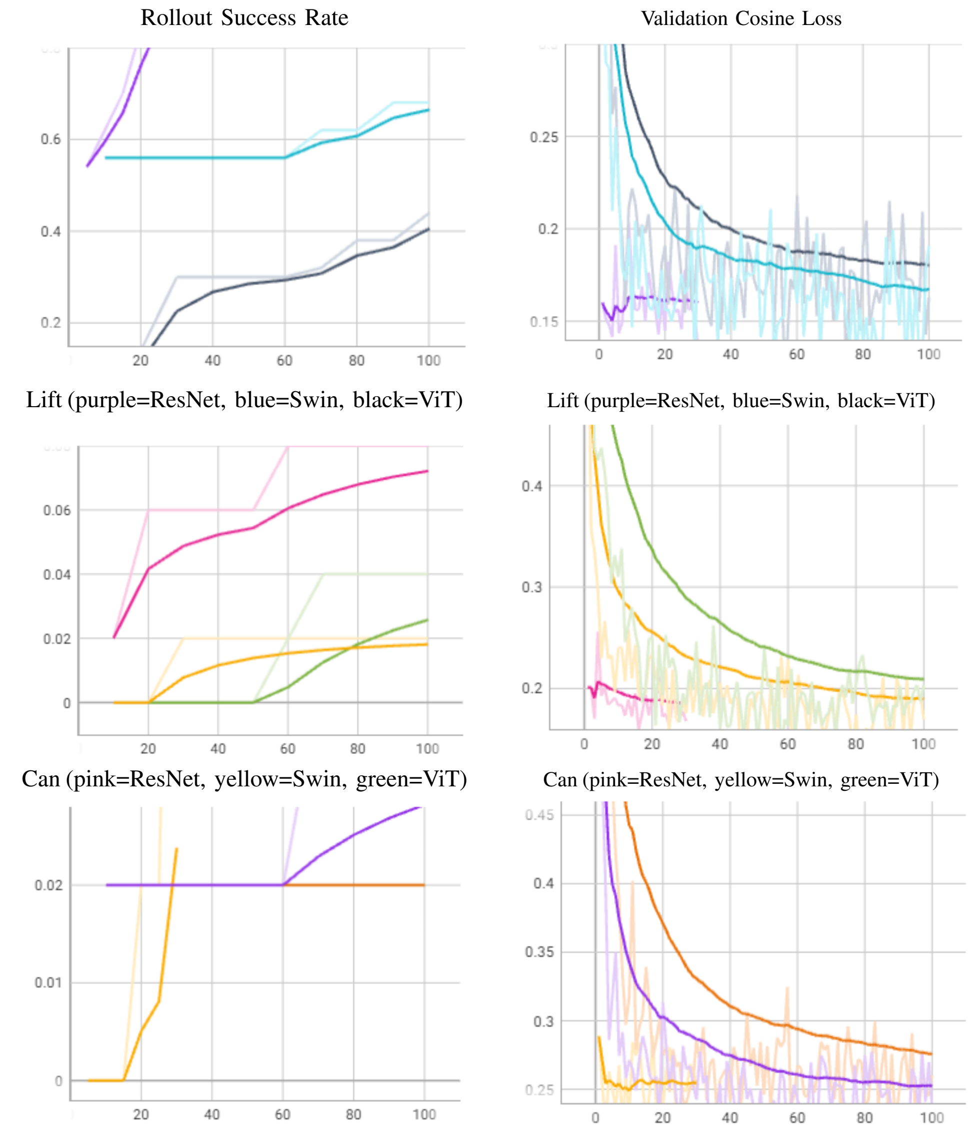
Fig. 7: Effect of policy network architecture on task performance



VI. Conclusion

From our observation featurization experiments, we can see that the Transformer-based visual encoders have difficulty outperforming the ResNet baseline. Because of the lack of spatial inductive bias in models like the Vision Transformer compared to a CNN like ResNet, these visual encoders may require extensive pre-training in order to outperform ResNet models. Furthermore, our experiments showed us that training Transformer models required extensive hyperparameter fine-tuning in order for learning to proceed.

From our experiments with Transformers for Behavioral Cloning policy networks, the Transformer policy network performed up to the levels of the LSTM model. However the expected increase in success rates was not observed. Several alterations on the architecture were tried to outperform the LSTM benchmark. This included varying the history size over which the model attends to. While we expected this increase in history size to help improve performance (as it approaches the infinite history size used by an LSTM), one



Lift (purple=ResNet, blue=Swin, black=ViT)

Lift (purple=ResNet, blue=Swin, black=ViT)

Can (pink=ResNet, yellow=Swin, green=ViT)

Can (pink=ResNet, yellow=Swin, green=ViT)

Can (yellow=ResNet, purple=Swin, orange=ViT)

Can (yellow=ResNet, purple=Swin, orange=ViT)

TABLE II: Observation featurization rollout success rate and validation plots for Lift, Can, and Square tasks

Model	Lift	Can	Square
LSTM (baseline)	0.92	0.08	0.1
Swin Transformer	0.68	0.04	0.04
ViT	0.44	0.02	0.02

TABLE III: Maximum rollout success rate for Behavioral Cloning under various featurization schemes

possible explanation for the results observed is that the tasks considered are so short-horizon in nature that increasing the history over which attention is computed has no beneficial effect on performance. The slight decrease can be attributed to difficulty in learning from a limited dataset.

Finally, our experiments comparing the added benefit of high-dimensional visual observations over simple proprioceptive state observations showed that our Transformer model was able to successfully take advantage of the visual modalities to learn a more robust policy. However, more work is needed to develop a mechanism by which these modalities can be better combined with the proprioceptive information in order to outperform the LSTM model.

A. *Future Work*

There are several future directions that we can take to explore performance improvements for our Transformer Encoder policy network.

One important future direction to explore is cross-modal attention within the observations at each time-step in the trajectory. Instead of mixing the visual embedding and proprioceptive embedding with an MLP, we could input the visual feature vector to a cross-modal mechanism in a Transformer Encoder. Given RGB visual feature sequence V and proprioceptive embedding Z, we can compute cross-modal attention as

$$A_{CM} = softmax\Big(\frac{Z \cdot V^T}{\sqrt{M}}\Big)V$$

Where M is a normalizing factor. This cross-modal attention may provide some additional benefit over general mixing of the modalities via an MLP.

Another idea is to implement a future observation state prediction head to the end of our current model. Alongside the model's prediction of a_t via the existing action decoding head, this future prediction head would predict $\hat{o}_{t+1}$, the low-dimensional (proprioceptive) state vector at the immediate future time-step. This head will be trained via an L1 loss with the true proprioceptive observation vector at that step, o_{t+1}. This may help serve as a form of regularization or smoothing on the predicted action sequences of the Transformer, which may in turn produce smoother trajectories that have a higher success rate and avoid the "shaky" motion we observe qualitatively in our rollouts.

A key direction for future work is to identify and evaluate the Transformer-Encoder model on tasks with a wider variety of disparate data sources, which are still relatively short-horizon. These multi-modal, short-horizon tasks may offer the conditions in which our Transformer-Encoder outperforms an RNN-based approach.

Another direction to be explored is evaluating sim-to-real generalizability of these BC models. In particular, simulated data may offer only a limited type of trajectory with low variability in the actual path taken. Training on a mix of simulated and real-world trajectories with larger variability in path may help the model learn a more robust policy, and evaluating rollouts on real-world setups offers a better characterization of the model's true performance.

References

[1] Arthur Bucker et al. *Reshaping Robot Trajectories Using Natural Language Commands: A Study of Multi-Modal Data Alignment Using Transformers*. 2022. DOI: 10.48550/ARXIV.2203.13411. URL: https://arxiv.org/abs/2203.13411.

[2] Kashyap Chitta et al. *TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving*. 2022. DOI: 10.48550/ARXIV.2205.15997. URL: https://arxiv.org/abs/2205.15997.

[3] Sudeep Dasari and Abhinav Gupta. *Transformers for One-Shot Visual Imitation*. 2020. DOI: 10.48550/ARXIV.2011.05970. URL: https://arxiv.org/abs/2011.05970.

[4] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2020. DOI: 10.48550/ARXIV.2010.11929. URL: https://arxiv.org/abs/2010.11929.

[5] Heecheol Kim, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. *Transformer-based deep imitation learning for dual-arm robot manipulation*. 2021. DOI: 10.48550/ARXIV.2108.00385. URL: https://arxiv.org/abs/2108.00385.

[6] Ze Liu et al. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. 2021. DOI: 10.48550/ARXIV.2103.14030. URL: https://arxiv.org/abs/2103.14030.

[7] Ajay Mandlekar et al. *RoboTurk: A Crowdsourcing Platform for Robotic Skill Learning through Imitation*. 2018. DOI: 10.48550/ARXIV.1811.02790. URL: https://arxiv.org/abs/1811.02790.

[8] Ajay Mandlekar et al. *What Matters in Learning from Offline Human Demonstrations for Robot Manipulation*. 2021. DOI: 10.48550/ARXIV.2108.03298. URL: https://arxiv.org/abs/2108.03298.

[9] Adam Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. DOI: 10.48550/ARXIV.1912.01703. URL: https://arxiv.org/abs/1912.01703.

[10] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. *Multi-Modal Fusion Transformer for End-to-End Autonomous Driving*. 2021. DOI: 10.48550/ARXIV.2104.09224. URL: https://arxiv.org/abs/2104.09224.

[11] Ashish Vaswani et al. *Attention Is All You Need*. 2017. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.