Accept Synthetic Objects as Real: End-to-End Training of Attentive Deep Visuomotor Policies for Manipulation in Clutter

Pooya Abolghasemi[1], Ladislau Bölöni[1]

Abstract— **Recent research demonstrated that it is feasible to end-to-end train multi-task deep visuomotor policies for robotic manipulation using variations of learning from demonstration (LfD) and reinforcement learning (RL). In this paper, we extend the capabilities of end-to-end LfD architectures to object manipulation in clutter. We start by introducing a data augmentation procedure called Accept Synthetic Objects as Real (ASOR). Using ASOR we develop two network architectures: implicit attention ASOR-IA and explicit attention ASOR-EA. Both architectures use the same training data (demonstrations in uncluttered environments) as previous approaches. Experimental results show that ASOR-IA and ASOR-EA succeed in a significant fraction of trials in cluttered environments where previous approaches never succeed. In addition, we find that both ASOR-IA and ASOR-EA outperform previous approaches even in uncluttered environments, with ASOR-EA performing better even in clutter compared to the previous best baseline in an uncluttered environment.**

I. INTRODUCTION

Recent research demonstrated the feasibility of end-to-end learning of deep visuomotor policies for robot manipulation. Such a policy can be denoted as $\pi(O, T; \phi) \rightarrow a$ where O is an observation involving the vision input of the robot and possibly other sensors such as robot proprioception, T is a specification of the current task and a is a command sent to the robot. The policy is implemented as a neural network with parameters ϕ. Some approaches propose a learning from demonstration (LfD) model, where the training data consists of demonstrations in the form $D_i = \{T_i, O_{0i}, a_{0i}, \ldots O_{ni}, a_{ni}\}$. An alternative is the use of reinforcement learning (RL), which requires a task-specific reward function $r(O, T)$. The objective of learning is to find an appropriate parameterization ϕ of the policy that maximizes some metric of manipulation success. Robot end-to-end learning usually does not take place in a "big data" regime, as demonstrations are challenging to collect, reinforcement trials are expensive and dense reward functions difficult to engineer. A series of recent papers had investigated various settings of the problem, ranging from pure LfD over uninitialized networks to pure RL, as well as combinations of LfD and RL. The sparsity of training data puts a special focus on incorporating an appropriate inductive bias into the network structure and learning process.

In this paper, we consider the case of a robotic arm that was assigned the task to manipulate (push, pick up) a certain object. The task specification has the form $T =$

$\{a_m, (f_s, f_c)\}$ where a_m is the manipulation action while the object to be manipulated is identified by two features: the shape f_s and color f_c. Examples include "Pick up the white towel" or "Push the red bowl".

One of the challenges which had not yet been consistently solved by deep visuomotor policies is the purposeful manipulation of objects in the presence of random clutter. For the purpose of this paper, we define clutter as objects in the scene that do not need to be manipulated. If a scene contains a bowl and a towel, with the task to pick up the bowl, then the towel is clutter. If the robot needs to push aside the towel to pick up the bowl, then the towel becomes the target object for the push subtask.

In the following, we first discuss why operation in clutter is a particular problem for end-to-end learned policies and then outline our proposed solution. In a typical deep visuomotor policy there is an internal representation bottleneck we will call the *primary latent encoding* $\mathbf{z}$ separating the network into a vision component $f_v(\cdot) \rightarrow \mathbf{z}$ and a motor component $f_m(O, T) \rightarrow a$. It is tempting to use an off-the-shelf pretrained network as a vision component such as VGG-19 or ResNet and to keep the size of the encoding small. A low dimensional $\mathbf{z}$ allows us to keep the number of demonstrations and/or reinforcement trials low. For instance, it was found that if $||\mathbf{z}|| \approx 64$, the motor network can be trained from scratch with less than a hundred demonstrations per task, a number which can be further reduced with techniques such as meta-learning.

Clutter, however, presents a problem to a small, pretrained latent encoding. As the off-the-shelf encoding does not depend on the task specification T, the encoding $\mathbf{z} = f_v(O)$ will need to represent the entire scene, leaving it to the motor network to sort out what is important. If $||\mathbf{z}||$ is small, there is simply no space for an accurate representation for many objects in the scene. The quality of representation for the target object will suffer, impacting the success rate of the policy.

There are several ways this problem can be solved. We can increase the size of $||\mathbf{z}||$ - but this requires a corresponding, usually higher-than-linear increase in the training data. For instance, we would need to provide a large number of demonstrations done in various clutter scenarios. Another possibility is to use a higher dimensional encoding $\mathbf{z}$, but enforce on it a pre-defined object-oriented representation [1]. A benefit of this approach is that it is more explainable. However, to some degree, it backs off from the end-to-end learned paradigm towards more engineered approaches.

The approach we take in this paper retains the fully end-

[1]Pooya Abolghasemi and Ladislau Bölöni are with the Dept. of Computer Science, University of Central Florida, Orlando, United States pooya.abolghasemi@knights.ucf.edu, lboloni@cs.ucf.edu

to-end trained model. As a first step, we require the visual component of the network to create a primary latent encoding that depends on the current manipulation task $\mathbf{z} = f_v(O, T)$ - this can be seen as a way to reserve a larger part of the representation to the target object. Nevertheless, this does not entirely eliminate the need for a variety of demonstrations in conditions of clutter. Our approach is based on the observation that such extra demonstrations convey new visual data, but very little new motion information. Human demonstrators already have a mental mechanism for ignoring clutter – thus they will usually deliver the same robot trajectory whether clutter is present or not. This leads us to the idea that it should be possible to train a visuomotor policy that performs under clutter conditions without requiring *any* demonstration in clutter.

The contributions of this paper are as follows. (1) We introduce a data augmentation technique and associated training regime called Accept Synthetic Objects as Real which allows for the generation of training data suitable for training manipulation in the clutter from demonstrations collected in scenarios without clutter. (2) We introduce a network architecture ASOR-Implicit Attention that trains a visual representation that implicitly encodes an attention feature on the target object in the primary latent encoding and its associated motor component. (3) We introduce a network architecture ASOR-Explicit Attention which trains a visual representation that explicitly focuses on the target object in the form of spatial attention, as well as its associated motor component. Both approaches are trained with demonstrations generated by ASOR. (4) We evaluate the two approaches on a series of manipulation experiments and compare them with previous approaches trained on the same data. The results show that ASOR-IA and ASOR-EA succeed in a significant fraction of trials in cluttered environments where previous approaches never succeed. In addition, we find both ASOR-IA and ASOR-EA outperform previous approaches even in uncluttered environments, ASOR-EA performs better than ASOR-IA, and ASOR-EA performs better even in clutter compared to the previous approaches in an uncluttered environment.

II. RELATED WORK

The canonical approach for robot manipulation involves decomposing the task into several well-defined components such as state estimation, planning, and low-level control, which often have further subcomponents such as a model of environment's dynamics in the planner. Recent advances in deep learning led to solutions where some of these components are implemented as neural networks. Examples include object pose estimation [2], robot pose estimation [3]. In some cases, more than one component is implemented with neural networks, such as the inference of grasp and suction affordances and recognizing novel objects [4].

Taking this trend to the extreme are end-to-end learned deep visuomotor policies where all or almost all the canonical pipeline is implemented as a *single* neural network, transforming in one step a visual input into the control

signal [5], [6]. The challenge, of course, is that we need to learn a large, opaque neural network, losing the advantages of decomposition and module testing.

The two major learning models of end-to-end policies are reinforcement learning RL and learning from demonstration LfD. For RL, the task is specified through a reward function and requires the robot to interact with the environment in order to collect training data. An approach to avoid the requirement of reward engineering by periodic querying of the user is shown in [7]. Although some projects collected a large number of reinforcement trials [8], for most projects the goal is to reduce the number of physical trials required. This can be done by training in simulation with testing done in the same simulated environment [9], [6] or the learned behavior will be transferred to a real robot [10]. Another alternative is to use LfD to bootstrap RL and overcome exploration challenges [6], [10].

LfD has the advantage that it does not require an engineered reward function, the tasks being specified by demonstrations performed by the user. In a very primitive setting, using only a behavioral cloning/imitation loss, LfD would be limited to reproducing the trajectories of the user, with fast divergence whenever a new situation is encountered. In practice, there are many techniques that can learn well-generalizing policies from demonstrations by combining the imitation loss with other losses, regularization techniques and other ways to introduce inductive bias. Just like in the case of RL, it is desirable to reduce the number of demonstrations required to learn a new task. One possible approach for this is meta-learning: by utilizing demonstrations from a variety of tasks, under some conditions it is possible to learn a new task from a single demonstration [11].

One of the features of our model is the use of attention in the representation of the visual input. Attention is a technique which both humans and AI use to process a large amount of data in real-time. Attention can caption images [12], [13], answer questions based on videos [14], [15], help robots navigate through crowds [16]. In [17] the authors use a set of demonstrations to learn the attention among region proposals from a pre-trained network on the MSCOCO dataset [18] to create an object-centric representation to be used by an RL-learned robot manipulation policy. This can be contrasted to our approach where the attention is obtained in a single step, it does not rely on external training data, and its focus object is specified from a natural language description of the task.

III. ACCEPT SYNTHETIC OBJECTS AS REAL

Our objective is to teach a robot arm to manipulate objects of different types under conditions of clutter. We will perform this by training a visuomotor policy that takes as input a video stream of the scene and generates commands to the robot. To be able to make a one-to-one comparison to previous approaches, we will reuse one of the existing datasets for which both training data and code for previous approaches is publicly available [19]. This scenario includes training data for picking up 8 different object types and pushing to place 5 different object types. The objects are

distinguished by shape and color, and have different degrees of rigidity including a rigid plastic bowl, a foam dumbbell, a piece of bubble wrap and a cotton towel.
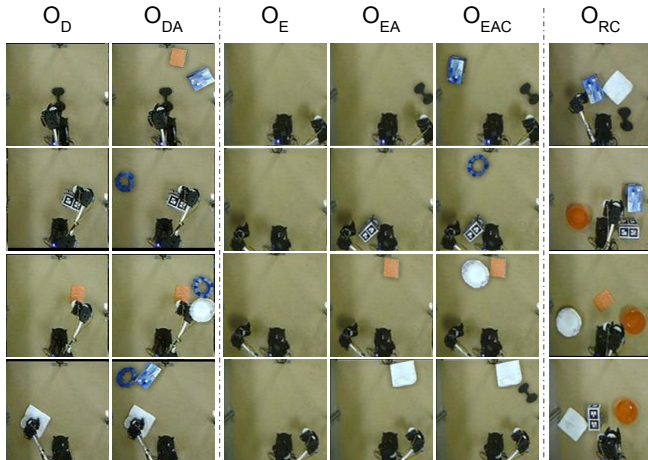

O_D	O_{DA}	O_E	O_{EA}	O_{EAC}	O_{RC}

Fig. 1. Faking clutter and accepting it as real. O_D: Demonstrations captured in an uncluttered environment [19]. O_{DA}: O_D images augmented by synthetic clutter. O_E: Empty images contain only the robot. O_{EA}: O_E images augmented by a synthetic target object. O_{EAC}: O_{EA} images augmented by synthetic clutter. O_{RC}: Real clutter(random objects in random positions).

The training data contains manipulation demonstrations done without the presence of clutter, with the only objects visible in the scene being the object to be manipulated and the robot arm itself (Figure 1 Demonstration Observation column or O_D).

As we discussed in the introduction, collecting training data in the presence of clutter is not an ideal way to use human effort, as humans will simply ignore the clutter. A better approach would augment the training data by adding clutter objects after the fact, using a visual manipulation of the images. This is shown in Figure 1 Augmented Demonstration Observation or O_{DA}.

Unfortunately, this approach, by itself, does not work. While copy-pasted synthetic objects may look natural to the human eye, they have low-level artifacts that allow the vision network to recognize them. As in these images, the target object is always real, and clutter objects are synthetic, the network learns to ignore clutter based on its synthetic nature. Such policies are not able to ignore clutter when tested with real objects. Attempts to improve our image processing capabilities to the degree that the artificially created clutter would be indistinguishable from the real one would be fighting against the training of the vision system which is motivated to distinguish them.

As our object is not to generate more convincing images but to train the robot, we propose an approach that intervenes in the end-to-end training loss of the robot. Instead of improving the synthetic objects to be indistinguishable from real objects, we train the vision system to Accept Synthetic Objects as Real. This technique requires both specific type of generated training images as well as training regime and losses relying on them. Columns 3-5 in Figure 1 show the new type of training images generated. We started by

recording a small set of images with only the robot arm in random positions and no other object (Empty Observations or O_E). Then we created images with one synthetic target object (Augmented Empty Observations or O_{EA}) and with a synthetic target object and several synthetic clutter objects (Clutter Augmented Observations or O_{EAC}). Note that there is no visual difference between the target object and the clutter objects: all of them are synthetic, the only feature of the target object is that it is referred to in the task specification. We can generate an arbitrary amount of training data through this technique. However, this training data is useful only for training the vision component, as it does not contain purposeful robot arm motion. We have another type of input images which we named Real Clutter Observations or O_{RC} which contains random real objects in random positions to give the model more generalization power.

With this set of images we use a training regime where the vision system does not distinguish between synthetic and real objects: it can accept the synthetic target object as real one in columns O_{EA} and O_{EAC}, and it can remove the clutter objects from O_{EAC} to create a generated scene as in O_{EA}.

In the following two sections we describe two different network architectures for visuomotor policies that take advantage of the ASOR model. We start by recognizing that any model where the primary latent encoding $\mathbf{z}$ prioritizes the target object specified by the task specification can be seen as attention based. This can be an *explicit attention* where the network actually creates a heatmap style 2D overlay for the visual field, a model used in ASOR-EA discussed in Section V. Alternatively, we can use ASOR in a network where such an explicit model does not exist – an *implicit attention* model such as the ASOR-IE discussed in Section IV.

IV. ASOR WITH IMPLICIT ATTENTION

The ASOR-IA network architecture, shown in Figure 2 uses a VAE-GAN based visual network evolved from the one introduced in [20] featuring an encoder E, generator G, and discriminator D.

Next, we discuss the discriminator's loss function $\mathcal{L}_D$, and the generator's loss function $\mathcal{L}_G$. All the parameters in the Discriminator have been optimized to minimize $\mathcal{L}_D$, and parameters of the Encoder, Generator and the motor network are optimized by the loss value $\mathcal{L}_G$.

The discriminator classifies the input frames based on target object's shape(s) and color(c), it also will classify fake frames in a separate class [21]. The discriminator's outputs are of lengths $|s| + 1$ and $|c| + 1$. $|s|$ and $|c|$ are respectively the number of unique shapes and colors and the "+1" represents the "fake" class. If the discriminator D receives a real frame from $\mathcal{O}$, it needs to classify the frame correctly in the corresponding object's shape and object's color class provided in the task specification ('Push the red bowl/pick-up the white towel'):

$$\mathcal{L}_{real} = - \mathbb{E}_{\mathcal{O},s\sim p_{data}}[\log{(P_D(s|\mathcal{O}))}]$$
$$- \mathbb{E}_{\mathcal{O},c\sim p_{data}}[\log{(P_D(c|\mathcal{O}))}] \quad (1)$$
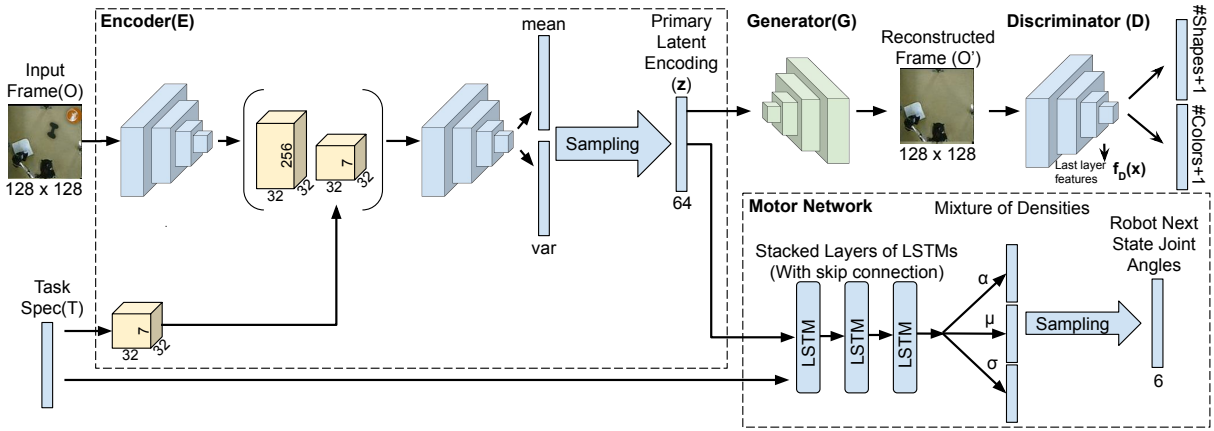
Fig. 2. The ASOR-IA network architecture. In each time-step the **Encoder(E)** receives the task specification and an image(O) as input. Features extracted from the task specification will be concatenated to feature maps extracted from the image somewhere in the middle of the CNN layers. The **Generator(G)** is trained to reconstruct the image given the primary latent encoding **z**. The reconstructed image should only contain the target object and ignore all the unnecessary information. **z** and the task specification is then passed to the motor controller to generate the next robot joint angles.

where $\mathcal{O} \in \{\mathcal{O}_D, \mathcal{O}_{EA}\}$ and P_D is the class probabilities produced by the discriminator for object's shape and color. Similarly, if D receives fake frames generated by the generator, it should classify them as fake:

$$\mathcal{L}_{fake} = - \mathbb{E}_{\mathcal{O}' \sim G}[\log\left(P_D(|s| + 1 \big| \mathcal{O}')\right)] \\ - \mathbb{E}_{\mathcal{O}' \sim G}[\log\left(P_D(|c| + 1 \big| \mathcal{O}')\right)] \quad (2)$$

where $\mathcal{O}'$ is the generator's reconstruction of any of the 5 input types. Finally, if D receives fake frames, generated by G with the latent representation $\mathbf{z} \sim \mathcal{N}(0, 1)$:

$$\mathcal{L}_{noise} = - \mathbb{E}_{z \sim noise}[\log\left(P_D(|s| + 1 \big| G(z))\right)] \\ - \mathbb{E}_{z \sim noise}[\log\left(P_D(|c| + 1 \big| G(z))\right)]. \quad (3)$$

The overall loss of the discriminator is thus $\mathcal{L}_D = \mathcal{L}_{real} + \mathcal{L}_{fake} + \mathcal{L}_{noise}$.

The Generator(G) must reconstruct a real looking frame(O') that only contains the object described in the input sentence. In fact, G tries not only to look real, but also presents the correct object in its output.

$$\mathcal{L}_{GD} = - \mathbb{E}_{\mathcal{O}', s \sim p_G}[\log P_D(s | \mathcal{O}')] \\ - \mathbb{E}_{\mathcal{O}', c \sim p_G}[\log P_D(c | \mathcal{O}')]. \quad (4)$$

The generator needs to learn to remove the irrelevant objects present in the scene. The discriminator as described in equation 1, only receives single object images. As a result, it should be able to easily classify multi object images as fake. Moreover, to enforce such behavior and to stabilize the training process we use feature matching [21] technique and the reconstruction error. Using these two terms in our loss function, we will force the network to reconstruct images of type O_{DA} and O_{EAC} as close as possible to the corresponding images of type O_D and O_{EA}.

$$\mathcal{L}_{rec} = ||O'_D - O_D||^2 + ||O'_{DA} - O_D||^2 \\ + ||O'_{EA} - O_{EA}||^2 + ||O'_{EAC} - O_{EA}||^2. \quad (5)$$

$$\mathcal{L}_{fea} = ||f_D(O'_D) - f_D(O_D)||^2 + ||f_D(O'_{DA}) - f_D(O_D)||^2 \\ + ||f_D(O'_{EA}) - f_D(O_{EA})||^2 + ||f_D(O'_{EAC}) - f_D(O_{EA})||^2 \quad (6)$$

where $f_D(x)$ is features extracted by the discriminator in the last layer for input x. To regularize the primary latent encoding (**z**), we minimize the KL-divergence between **z** and $\mathcal{N}(0, 1)$:

$$\mathcal{L}_{prior} = D_{\mathrm{KL}}(E(O, T) \ || \ \mathcal{N}(0, 1)). \quad (7)$$

Finally, we write the generator's loss as $\mathcal{L}_G = \mathcal{L}_{DG} + \mathcal{L}_{rec} + \mathcal{L}_{prior} + \mathcal{L}_{motor}$, where the $\mathcal{L}_{motor}$ is calculated based on the behaviour cloning loss similar to [20].

V. ASOR WITH EXPLICIT ATTENTION

The second architecture we discuss, ASOR-EA generates explicit attention maps for the input frames. The network, shown in Figure 3 has 5 primary components: encoder, generator, discriminator, attention and the motor network. First, the attention module extracts an attention mask to cover the parts of the input frame that are not relevant to the current task (the clutter). The masked frame M is the result of pixel-wise multiplication of the attention map A and the input frame O. Given the masked frame M, the encoder-generator pair will try to extract a set of features descriptive enough to create reconstructed versions of the original frame O' and of the masked frame M'. The reconstruction of the masked frame M' is necessary to ensure that the information regarding the attention map is preserved in the latent encoding to be used later by the motor controller. O' is fed to the discriminator network.

The attention network combines features from the task specification and the input frame to extract the attention map. Several convolutional layers augmented by batch normalization are used to obtain K spatial visual feature vectors. These convolution layers will provide spatial visual features with size $\phi_f \in \mathcal{R}^{K \times d_\phi}$, where d_ϕ is the number of features for region k_i. The task specification encodes the target object
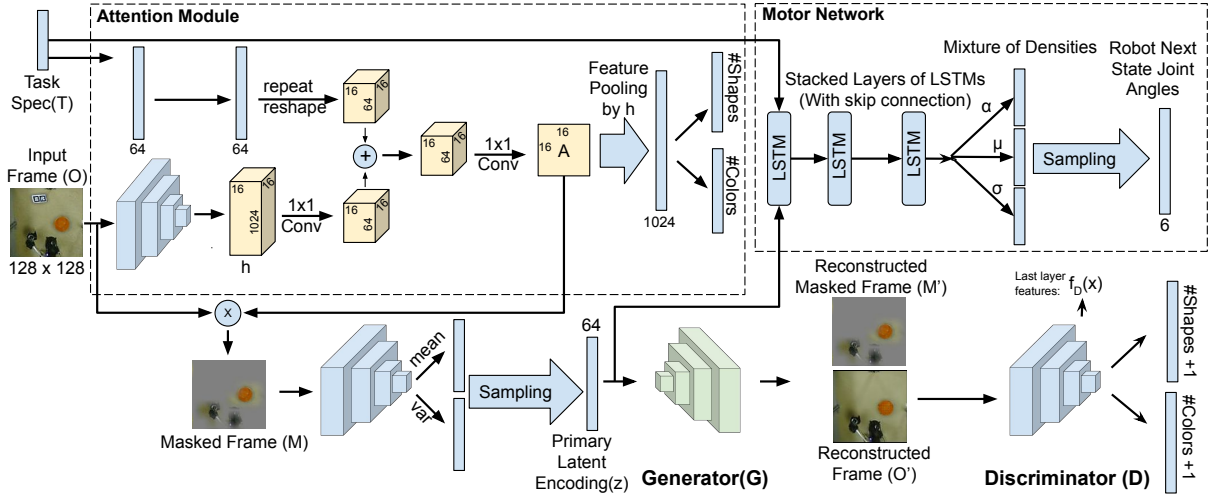
Fig. 3. The ASOR-EA network architecture. The task specification and the image capture are the inputs to the attention network. The attention network generates the attention map and should be able to classify the input image based on the features pooled from feature maps pooled from the last convolution layer of the attention map. The VAE-GAN will receive a masked image of the original input by the attention map and should be able to reconstruct the input image without the clutter objects.
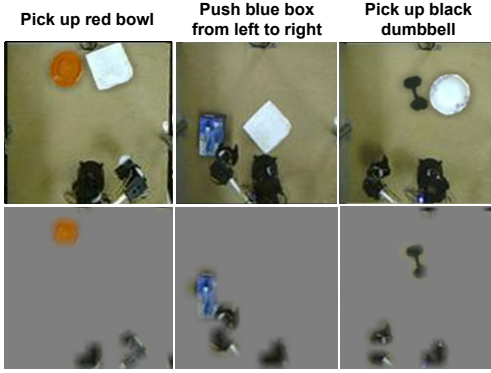

Fig. 4. Examples of the operation of the attention module in ASOR-EA. The first row is the input image O and the second row is the masked image M for the task specification on the top of the column.

as two concatenated one-hot vectors describing the shape and color of the object respectively. Several fully-connected layers are used to transform the task encoding to shape $u \in \mathcal{R}^{d_\psi}$. In order to combine the visual and task features repeated the u vector k times and combined it with the visual features using a technique similar to [15], [14]:

$$\psi = \tanh(\phi_f \times W_f \oplus u), \qquad (8)$$

where $W_f \in \mathcal{R}^{d_\phi \times d_\psi}$ is a mapping matrix, $\oplus$ is element-wise summation. The combined feature vector is of size $\mathcal{R}^{k \times d_\psi}$. To compute the final attention map we calculate

$$p = \text{sigmoid}(\psi \times W_p), \qquad (9)$$

$$p_{TFA} = \text{ReLU}(p - t), \qquad (10)$$

where $W_p \in \mathcal{R}^{d_\psi \times 1}$ represents a fully-connected layer, and t is a hyper-parameter. t can be a constant or it can be a statistical metric such as a running average over p. Here we set t to the constant value of 0.5. ReLU and the hyper-parameter t are needed to force the network to completely hide the information from the irrelevant regions by assigning

a score of exactly 0 to most regions. Without these, the attention map will keep small values in most regions and the encoder can easily reconstruct the original image. The final $p_{TFA} \in [0, 1]^k$ contains attention scores of all k regions.

Similar to the teacher network in [19] the network should be able to reconstruct the task specification from the pooled spatial features weighted by the attention map, p_{TFA}. Weighted pooled features are defined as $p_f = \sum_{i \in k} p_{TFA_i} \phi_{f_i}$ where $p_f \in \mathcal{R}^{d_\phi}$ and are passed to two multi-layer perceptrons (MLP) $\hat{s} = \tau_1(p_f)$, $\hat{c} = \tau_2(p_f)$. These MLP layers try to classify the pooled features based on target object's shape and color. In other words, the network is trying to focus the attention on regions which provide better information to reconstruct the task specification and an image which only contains the target object.

We use the L1 norm of the p_{TFA} and softmax cross entropy between the f_s and f_c from the task specification and $(\hat{s}, \hat{c})$ to calculate $\mathcal{L}_A$.

The masked frame M is generated using the attention map, $M = O \otimes p_{TFA}$ where $\otimes$ is element-wise multiplication. Examples of input frames and the masked frames using the computed attention p_{TFA} are shown in Figure 4.

The encoder E receives M as input, $[\mu_{\mathbf{z}}|\sigma_{\mathbf{z}}] = E(M)$ where μ_z, $\sigma_z \in \mathcal{R}^{d_z}$, and d_z is the length of the primary latent encoding $\mathbf{z}$. E is a multi-layer convolutional neural network with a $2d_z$ dimensional vector output split into μ_z and σ_z. We assume that $\mathbf{z} \sim \mathcal{N}(\mu_z, \sigma_z)$. The generator receives $\mathbf{z}$ and reconstructs the input frame O and the masked frame M. Note that in this case information regarding the target object and the text encoding is not passed to the encoder. The encoder should rely solely on the masked frame to extract the required information for G to reconstruct the original frame and the masked frame. In addition to the $\mathcal{L}_A$, the reconstruction loss between different types of input images will also force the attention to focus on parts of the image related to the task.
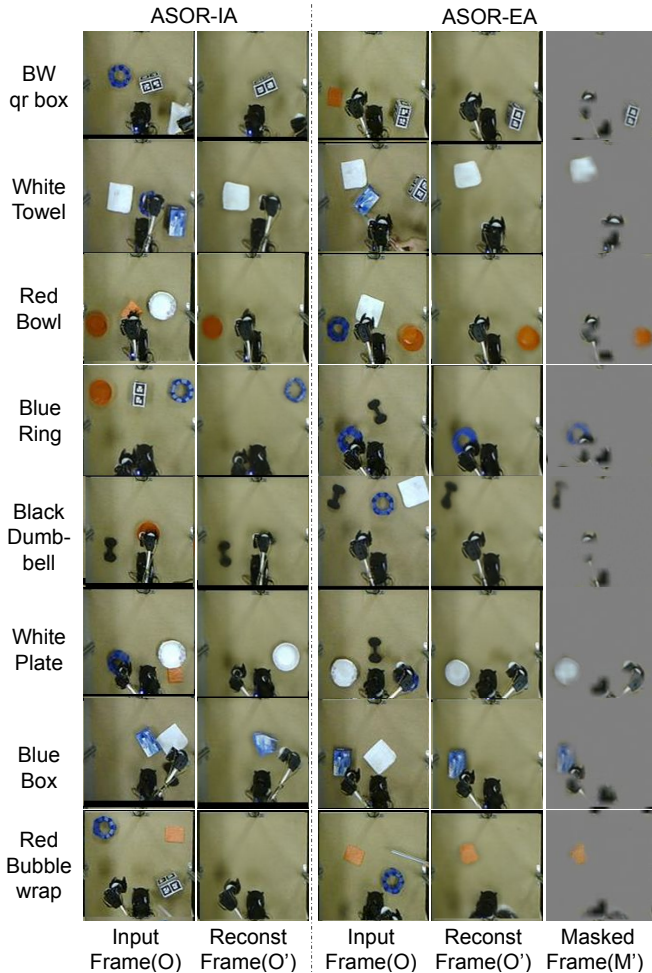
ASOR-IA | ASOR-EA

BW qr box
White Towel
Red Bowl
Blue Ring
Black Dumb-bell
White Plate
Blue Box
Red Bubble wrap

Input Frame(O) | Reconst Frame(O') | Input Frame(O) | Reconst Frame(O') | Masked Frame(M')

Fig. 5. Each row illustrates ASOR-IA and ASOR-EA reconstructions for a specific object. First column is the input frame(O) to ASOR-IA and the second column is it's output(O'). The third column is the input frame(O) to ASOR-EA and the next two columns are the generator's reconstructions O' and M'.

The loss functions of the discriminator, generator and motor network are similar to the one for ASOR-IA. The only difference being that $\mathcal{L}_A$ and the reconstruction loss for the masked frame M' is added to the generator's loss $\mathcal{L}_G$.

VI. EXPERIMENTS

Comparing robot control approaches is a general challenge due to difficult to reproduce environmental settings. It is to be hoped that in future years with the spread of inexpensive, replicable platforms such as REPLAB [22] and PyRobot [23], such one to one comparisons will be easier to make. For this paper, we will test the proposed approach over 10 tasks for which training data is publicly available [19]. The four algorithms we compare are the one from [20] which does not use an attention model, the algorithm from [19] which uses a language induced attention model called TFA, ASOR-IA as described in Section IV and ASOR-EA as described in Section V. All algorithms were trained with the same demonstrations.

The task set was composed of 6 tasks of picking up objects (a red bowl, a white towel, a blue ring, a black foam

TABLE I
SUCCESS PERCENTAGES: NO CLUTTER / WITH CLUTTER

	Task spec.	w/o att. [20]	w/ att. [19]	ASOR-IA	ASOR-EA
Pick up	Red Bowl	70 / 0	80 / 0	80 / 60	**90 / 80**
	White Towel	50 / 0	80 / 0	**100** / 70	**100 / 100**
	Blue Ring	30 / 0	60 / 0	60 / 40	**80 / 80**
	Black Dumbbell	40 / 0	50 / 0	40 / 50	**60 /70**
	White Plate	60 / 0	80 / 0	80 / 60	**100 / 90**
	Red Bubblewrap	10 / 0	**40 / 0**	30 / 20	40 / 20
	All pick up	43.3 / 0.0	65.0 / 0.0	65.0 / 50.0	**78.3 / 70.0**
Push to right	Red Bowl	80 / 0	**100 / 0**	80 / 80	**100 / 100**
	White Plate	60 / 0	60 / 0	**70** / 50	60 / 70
	Blue Box	10 / 0	30 / 0	60 / 50	**90 / 90**
	BW QR-box	20 / 0	60 / 0	60 / 40	**80 / 70**
	All push tasks	42.5 / 0	62.5 / 0	67.5 / 57.0	**82.5 / 82.5**
	Overall	43.0 / 0.0	64.0 / 0.0	66.0 / 53.0	**80.0 / 75.0**

dumbbell, a white plate and a red bubblewrap) and 4 tasks for pushing objects (a red bowl, a white plate, a blue box and a black-and-white QR patterned box). Figure 5 illustrates ASOR-IA and ASOR-EA reconstructions for each of these objects. For each algorithm, we repeated the experiments 10 times under two types of conditions: in an uncluttered scene with only the target object and the robot visible and in a cluttered scene where 2-3 other objects were also present. A trial was counted as a success if the manipulation was completed in 2 minutes.

Table I shows the experimental results as percentages of the successes for the no-clutter / clutter case, with the best results for each task setting highlighted in **bold**[1].

As expected, we find that for almost all trials, the results were worse in clutter. Neither of the previous approaches managed to complete any trial successfully in the presence of clutter. Note that the architecture proposed in [19] is trained with task-focused attention and is able to ignore physical and visual disturbances like a human hand and generally disturbances that are not present in the training set but clutter in our experiments contains objects involved in other demonstrations. ASOR-IA and ASOR-EA outperformed the previous approaches in the no-clutter setting, with ASOR-EA being the better of the two. More importantly, they managed to complete a significant portion of the tasks in clutter as well - in fact, ASOR-EA outperforms in clutter the previous best algorithm operating in an uncluttered setting[2].

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a data augmentation technique (Accept Synthetic Objects as Real) and two network models that take advantage of it to train end-to-end robot controllers which operate in the presence of clutter. Immediate extensions include more complex manipulation tasks with multiple objects. We believe, however, that the overall idea of co-training the network to accept simplified representations as real has applications in many other areas where robust, task-dependent representations need to be learned from limited training data.

[1]Checkout our YouTube video https://youtu.be/GchuLQhG3ug
[2]Code is available at https://github.com/pouyaAB/Accept_Synthetic_Objects_as_Real

REFERENCES

[1] Y. Jiang, S. Gu, K. Murphy, and C. Finn, "Language as an abstraction for hierarchical deep reinforcement learning," *arXiv preprint arXiv:1906.07343*, 2019.

[2] R. Rahmatizadeh, P. Abolghasemi, A. Behal, and L. Bölöni, "From virtual demonstration to real-world manipulation using LSTM and MDN," *Thirty-Second AAAI Conf. on Artificial Intelligence (AAAI-2018*, 2018.

[3] Y. Zuo, W. Qiu, L. Xie, F. Zhong, Y. Wang, and A. L. Yuille, "CRAVES: Controlling robotic arm with a vision-based economic system," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR-2019)*, 2019, pp. 4214–4223.

[4] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *Proc. of IEEE Int'l conf. on Robotics and Automation (ICRA-2018)*, 2018, pp. 1–8.

[5] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[6] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv preprint arXiv:1709.10087*, 2017.

[7] A. Singh, L. Yang, K. Hartikainen, C. Finn, and S. Levine, "End-to-end robotic reinforcement learning without reward engineering," *arXiv preprint arXiv:1904.07854*, 2019.

[8] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int'l Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[9] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. Levine, "Learning modular neural network policies for multi-task and multi-robot transfer," in *Proc. of IEEE Int'l Conf. on Robotics and Automation (ICRA-2017)*, 2017, pp. 2169–2176.

[10] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas, *et al.*, "Reinforcement and imitation learning for diverse visuomotor skills," *arXiv preprint arXiv:1802.09564*, 2018.

[11] T. Yu, C. Finn, A. Xie, S. Dasari, T. Zhang, P. Abbeel, and S. Levine, "One-shot imitation from observing humans via domain-adaptive meta-learning," *arXiv preprint arXiv:1802.01557*, 2018.

[12] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. of Int'l Conf. on Machine Learning (ICML-2015)*, 2015, pp. 2048–2057.

[13] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR-2016)*, 2016, pp. 4651–4659.

[14] A. Mazaheri, D. Zhang, and M. Shah, "Video fill in the blank using lr/rl lstms with spatial-temporal attentions," in *Proc. of the IEEE Int'l Conf. on Computer Vision (ICCV-2017)*, 2017, pp. 1407–1416.

[15] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR-2019)*, 2016, pp. 21–29.

[16] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *IEEE Int'l Conf. on Robotics and Automation(ICRA-2018)*. IEEE, 2018, pp. 1–7.

[17] C. Devin, P. Abbeel, T. Darrell, and S. Levine, "Deep object-centric representations for generalizable robot learning," *arXiv preprint arXiv:1708.04225*, 2017.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conf. on Computer Vision(ECCV-2014)*. Springer, 2014, pp. 740–755.

[19] P. Abolghasemi, A. Mazaheri, M. Shah, and L. Bölöni, "Pay attention! - Robustifying a deep visuomotor policy through task-focused visual attention," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR-2019)*, June 2019.

[20] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine, "Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration," in *Proc. of IEEE Int'l Conf. on Robotics and Automation(ICRA-2018)*, 2018, pp. 3758 – 3765.

[21] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[22] B. Yang, D. Jayaraman, J. Zhang, and S. Levine, "REPLAB: A reproducible low-cost arm benchmark for robotic learning," in *International Conference on Robotics and Automation (ICRA-2019)*, 2019, pp. 8691–8697.

[23] A. Murali, T. Chen, K. V. Alwala, D. Gandhi, L. Pinto, S. Gupta, and A. Gupta, "PyRobot: An open-source robotics framework for research and benchmarking," *arXiv preprint arXiv:1906.08236*, 2019.