Pay attention! - Robustifying a Deep Visuomotor Policy through Task-Focused Visual Attention

Pooya Abolghasemi [*], Amir Mazaheri [*], Mubarak Shah and Ladislau Bölöni
University of Central Florida, Orlando, FL 32816
pooya.abolghasemi, amirmazaheri@knights.ucf.edu, shah@crcv.ucf.edu, lboloni@cs.ucf.edu
https://pouyaab.github.io/pay-attention/

Abstract

Several recent studies have demonstrated the promise of deep visuomotor policies for robot manipulator control. Despite impressive progress, these systems are known to be vulnerable to physical disturbances, such as accidental or adversarial bumps that make them drop the manipulated object. They also tend to be distracted by visual disturbances such as objects moving in the robot's field of view, even if the disturbance does not physically prevent the execution of the task. In this paper, we propose an approach for augmenting a deep visuomotor policy trained through demonstrations with Task Focused visual Attention (TFA). The manipulation task is specified with a natural language text such as "move the red bowl to the left". This allows the visual attention component to concentrate on the current object that the robot needs to manipulate. We show that even in benign environments, the TFA allows the policy to consistently outperform a variant with no attention mechanism. More importantly, the new policy is significantly more robust: it regularly recovers from severe physical disturbances (such as bumps causing it to drop the object) from which the baseline policy, i.e. with no visual attention, almost never recovers. In addition, we show that the proposed policy performs correctly in the presence of a wide class of visual disturbances, exhibiting a behavior reminiscent of human selective visual attention experiments.

1. Introduction

Many recent researches show the possibility of end-to-end training of deep visuomotor policies that perform object manipulation tasks such as pick-and-place, push-to-location, stacking and pouring. These systems perform all the components of the task (vision processing, grasp and trajectory planning and robot control) using a neural network trained by variations of deep reinforcement learning and learning from demonstration (supervised learning).

[*]Equal Contribution.

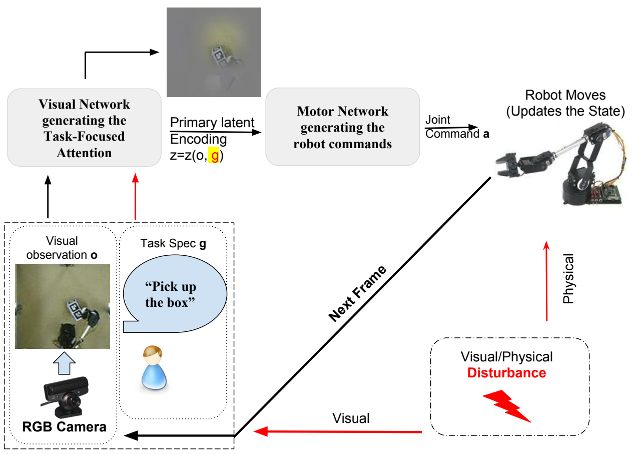
Visual Network generating the Task-Focused Attention
Primary latent Encoding z=z(o, g)
Motor Network generating the robot commands
Joint Command a
Robot Moves (Updates the State)
Visual observation o
RGB Camera
Task Spec g
"Pick up the box"
Next Frame
Physical
Visual/Physical Disturbance
Visual

Figure 1. The robot performs a given command. Our proposed network attends the image regions that matter the most, and is robust to physical and visual *disturbance.*

Deep visuomotor policies for manipulator control are neural network architectures that have as input an observation composed of an image or video frame and possibly other sensor data, $\mathbf{o}_t$, a task (or goal) specification, $\mathbf{g}$, and output robot commands, $\mathbf{a}_t = \pi(\mathbf{o}_t, \mathbf{g})$. The robot executes these commands, enacting a change in the external environment, which creates a new observation $\mathbf{o}_{t+1}$, and the cycle repeats. Architecturally, most currently proposed systems follow variations of the generic model of Figure 1, which posits the existence of a primary latent encoding, $\mathbf{z}$, the result of the visual processing of the input by a specialized visual network. This encoding, of dimensionality orders of magnitude smaller than the input, is then used by the motor network to generate the next state joint angles command, $\mathbf{a}$. While most demonstrations (supervised data) had been made in unstructured but relatively benign environments, our own experiments and personal communication with other researchers had shown that task independent visual networks for visuomotor policies are highly vulnerable to *physical* and *visual* disturbances. An example of physical disturbance is the robot arm being bumped such that it drops the manipulated object. The desired behavior would be for the robot to immediately notice this, change its trajectory, pick up the dropped object and continue with the manipulation task. Instead, with an otherwise reliably performing policy, we notice situations where the robot arm, having lost the object, continue to go empty-handed through the full trajectory of the manipulation, recovering either much later, or not at all. A visual disturbance may involve distracting mobile objects appearing in the robot's field of view. Clearly, if the visual disturbance prevents the execution of the task, for instance, by blocking the view of the manipulated object, it is acceptable for the robot to stop or even cancel the manipulation. There are, however, visual disturbances that should not prevent the execution of the task: for instance, hands waving in the visual field of the robot but not covering the manipulated object or the robot arm. We have found that in the case of an task independent visual network, even such visual disturbances cause the robot to behave erratically – possibly due to the robot interpreting the situation as a state never encountered before.

In engineered robot architectures such problems can be dealt by developing explicit models of the possible disturbances, which may allow the robot to reason around the situation. In deep learning systems, one possible brute-force solution is to gather more training data containing physical and visual disturbance events; however, data collection for robotic tasks is time consuming. Also, there are unlimited visual and physical disturbance scenarios for a single task. It is impossible for to record demonstrations to cover all possible scenarios of physical and visual disturbances.

Pay attention! Task dependent visual network: The principal idea of this paper is that performance benefits can be obtained if we make the vision system pay attention to relevant regions of each frame regarding the current task or user command. Humans are known to exhibit selective attention - when observing a scene with a particular task in mind, features of the scene relevant to the task are given particular attention, while other features are de-emphasized or even ignored. This had been illustrated in the famous experiments of Chabris and Simmons [1]. In this paper we propose *Task Focused (Visual) Attention* (TFA) as an auxiliary network to increase the robustness of the robot manipulator network to physical and visual disturbances, without the need of any additional training data. Thus, our objective is to create a system that implements a selective visual attention similar to what human perception is doing: we want the robot to focus on the objects of the scene that are relevant to the current manipulation task. We conjecture that using TFA, $\mathbf{z}$ will better represent the objects and colors that are the subject of the attention, allowing for more precision in grasping and manipulation (See Figure 2).

Our Contributions: The contributions of the paper are as follows: **1-** We describe a novel architecture for a visuomotor policy trained end-to-end from demonstrations, which features a task focused visual attention system. The visual attention system is guided by a natural language description of the task and focuses on the currently manipulated object. **2-** We show that, under benign conditions, the new policy outperforms a closely related baseline policy without the attention model over pick-up and push tasks using a variety of objects. **3-** We show that in the case of a severe physical disturbance, when an external intervention causes the robot to miss the grasp or drop the already grasped object, the new policy recovers in the majority of situations, while the baseline policy almost never recovers. **4-** We show that the task focused visual attention allows the policy to ignore a large class of visual disturbances, that interfere with the task for the baseline policy. We show experimentally that the system exhibits the "invisible gorilla" phenomenon [1] from the classic selective attention test. **5-** The teacher network for the task focused visual attention can be trained offline, does not require additional training data or pixel level annotation of objects.

2. Related Work

A deep visuomotor policy for robotic manipulation transforms an input video stream (possibly combined with other sensory input) into robot commands by the means of a single deep neural network. Such a system had been first demonstrated in [2] using guided policy search, a method that transforms policy search into supervised learning, with supervision provided by a trajectory-centric reinforcement learning method. In recent years, several alternative approaches have been proposed using variations of both deep reinforcement learning and deep learning from demonstration (as well as combinations of these).

Deep reinforcement learning is powerful paradigm which, in applications where exploration can be performed in a simulated environment allowing millions of trial runs, can train systems that perform at superhuman level [3], even when no human knowledge is used for bootstrapping [4]. Unfortunately, for training visuomotor policies controlling real robots, it is very difficult to perform reinforcement runs on these scales. Even the most extensive projects could only collect several orders of magnitude lower number of experiments: for example, in [5] 14 robotic manipulators were used over the period of two months to gather 800,000 grasp attempts. Even this number of experimental tries are unrealistic in many practical settings.

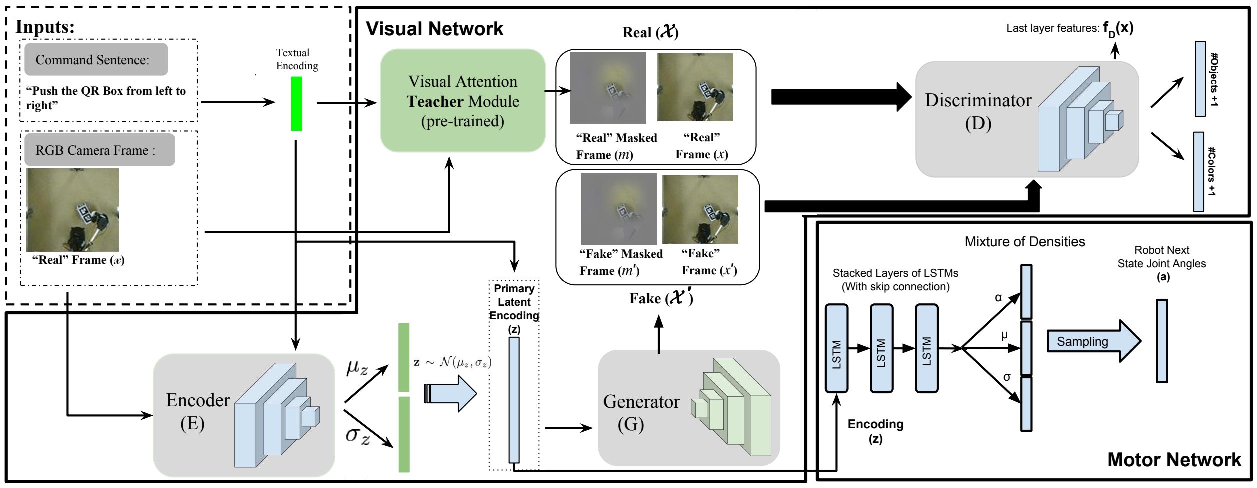

Figure 2. **The proposed visuomotor architecture**. Given an image captured from the scene and a command sentence provided by the user, the **Encoder (E)** produces the *Primary Latent Encoding (z)*. **z** is the input to the **Motor Network**, which decides the next state of the robot joint angles. Also, **z** is the input to a **Generator (G)**, which produces "Fake" frame and masked frame. A pre-trained Visual Attention **Teacher** Module masks the original frame by an spatial attention computed employing the textual input. The **Discriminator (D)** must discriminate between real/fake frames and masked frames, and also classify the object and color of the object being manipulated.

Thus, many efforts focus on reducing the number of experimental runs necessary to train an end-to-end visuomotor controller. One obvious direction is to learn a better encoding of the input data, which can improve the learning rate. In [6], a set of visual features were extracted from the image to be used as state representation for a reinforcement learning algorithm.

Another direction involves the use of learning from demonstration instead (or in combination with) of reinforcement learning. The demonstrations can be performed in real [7] or simulated [8, 9] environments. Meta-learning [10] and related approaches promise to drastically lower the amount of training data needed to learn a specific task from a class of related tasks (possibly, down to a single task specific demonstration). However, they still require a costly meta-learning phase.

An approach that is similar to ours in objective, but different in implementation, is described in [11]. Considering manipulation tasks, the authors implement two layers of attention. The first, a task independent visual attention semantically identifies labels and localizes objects in the scene. This labeling relies on training on an external labeled dataset, thus in this respect the approach is not "end-to-end". The second, a task-specific attention is learned by selecting from the segmented objects, by the task independent attention, those objects that contribute most to the correct prediction of demonstrated trajectories.

Another point concerns the way in which the task is specified to the robot. Specifying the task in the form of a human readable sentence is a natural choice [12], as creating such a command is very easy for a human user. In the general case, however, translating a command into a task is not yet feasible with an end-to-end learned controller. In this paper, we assume the existence of the command, but only as an additional input that helps the creation of the task-focused attention. Alternative ways of specifying the task are possible. A purely visual specification was proposed in [13], where the user identifies a pixel in the image and specifies where it should be moved. A technique of control based on visual images was also demonstrated in [14].

One component of our work has its roots in recent work on visual attention networks. These networks often appear as components of larger networks, solving problems like image captioning [15, 16], visual question answering [17, 18, 19] or visual expression localization [20]. Although the applications are different, the role of attention networks, i.e., focusing on information-rich parts of the visual input, remains the same. Our proposed attention mechanism is most similar to [17]. However, in our model we train the attention network with word selection objective. The objective is to select some regions on a video frame regarding a textual input, such that it be able to regenerate the words in the input sentence just based on the visual features of selected image regions.

3. Approach

As shown in Figure 2, our architecture contains a **Motor Network** and **Visual Network**.The Motor Network, often but not always, contains a recurrent neural network and is trained on a loss that favors the execution of the specified task, **g**. This training may take several forms. In the case of RL we need a source of rewards. If the task is specified by

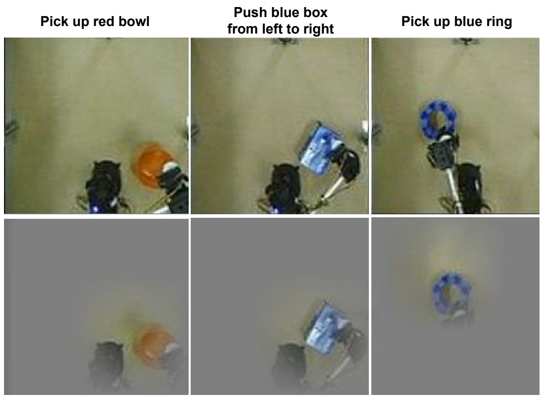

Figure 3. Examples of task focused visual attention. We provide the command sentence on top of each column. The first row shows frames from RGB camera and the second row is the same image masked by the attention, produced by **teacher network**. We denote the first/second row images by x/m in our equations.

demonstrations (our case), the training may be executed in a supervised fashion using a behavioral cloning loss.

The Visual Network contains an Encoder module that encodes the input frame into the *Primary Latent Variable,* **z**. To get a richer representation **z**, we incorporate two other modules. First, a teacher network which computes an attention map and masks the input frame. We train the teacher network separately (Section 3.1). Second, a GAN network that takes **z** as input and generates two reconstructed frames, the input frame and the masked input frame.

3.1. A Teacher Network for TFA

We consider robot manipulation commands expressed in natural language such as, "Push the *red plate* to the left", "Push the *blue box* to the left", and "Pick up the *red ring*".

The goal of the TFA is to identify the parts of the visual input, where objects relevant to the task appear, that is, to focus the attention on the red plate, blue box and blue ring respectively (see Figure 3).

A TFA system could be trained as a supervised learning model, if we can create a sufficient amount of training data. However, this would require us to label with attention blobs on an unrealistically large number of input video frames. Our approach is to generate our own labels by implementing a teacher network that provides training data for the controller. Our approach fits in the established technique of student-teacher network training models [21, 22, 23], with the qualification that the attention teacher only teaches one particular aspect of the final controller. In the remainder of this section, we describe the implementation of a teacher network which computes the TFA as shown in Figure 4.

The proposed approach allows us to train the TFA without pixel level annotations. The principal idea is that the attention should be on those regions that *allow us to reconstruct the input text based on those regions only*. The overall architecture is described in Figure 4.

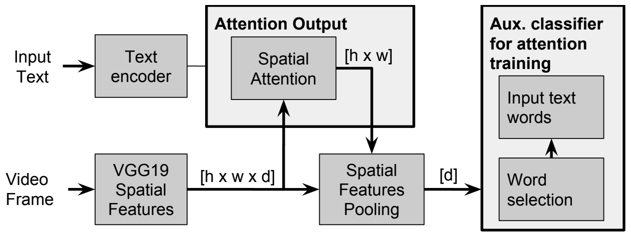

Figure 4. Proposed visual attention network. The network uses the pre-trained VGG19 [24] network's last convolution layer output as the visual spatial features. The attention module combines the spatial and textual features, and assigns one probability to each spatial region. To train the attention network, first we pool the visual features by the attention probabilities (weighted average), and second, we use an auxiliary classifier to reconstruct the input text's words based on the pooled visual features.

We divid the visual field (video frame), x, into k regions. The visual attention we aim to obtain is a vector of probabilities, $p_{TFA} \in (0,1)^k$, with a probability for each of the k regions. The higher the probability, the more attention is paid to the specific region. In general, our goal is to focus the attention on a small number of regions.

The first step is to encode the text and image inputs.

Text input: Let $\{v_1, v_2, \ldots, v_n\}$ be the textual input with n words, with one-hot indicators $v_i \in \{0,1\}^{|V|}$, where V is the dictionary of the words in our dataset. Thus, a word-to-vector encoding is employed:

$$w_i = v_i \times W_\omega, \tag{1}$$

where $W_\omega \in \mathcal{R}^{|V| \times d_v}$, and d_v is the length of encoded word vectors. To encode a whole sentence, we feed the series of word vectors to an LSTM. To obtain the text encoding, we extract the last hidden state of the LSTM, $u \in \mathcal{R}^{d_h}$, where d_h is cell size of the LSTM. We observed better performance by concatenating the LSTM output with a binary vector indicating objects' shape and color.

Visual input: We use the last Convolution layer of a pre-trained VGG19 [24] network to obtain k spatial visual feature vectors. The resulting spatial visual features have the form $\phi_f \in \mathcal{R}^{k \times d_\phi}$, where k is the number of spatial regions and d_ϕ is the length of feature vector for each region.

We combine the textual and visual encodings through a technique similar to [17]. We learn a mapping on both visual and text data and combine them through an element-wise summation:

$$\psi = \tanh(\phi_f \times W_f \oplus u \times W_u), \tag{2}$$

where $W_u \in \mathcal{R}^{d_h \times d_\psi}$ and $W_f \in \mathcal{R}^{d_\phi \times d_\psi}$ are mapping matrices, $\oplus$ is element-wise summation. $\psi \in \mathcal{R}^{k \times d_\psi}$ is the combination matrix of textual and visual inputs. Note that, u is a vector, while ϕ_f is a matrix. We augment the u vector by repeating it for k times. To compute the final attention probabilities, the model must assign higher scores to a few spatial regions.

$$p_{TFA} = \text{softmax}(\psi \times W_p), \tag{3}$$

where $W_p \in \mathcal{R}^{d_\psi \times 1}$ is trainable weights vector, which is used to assign a score to each region. The final $p_{TFA} \in (0,1)^k$ is the vector containing attention scores of all k regions'. We use a softmax non-linearity to force the network to attend to a few number of regions.

Our method does not need any spatial pixel level annotation to compute the attention. The attention in our formulation is a latent variable dependent on the input text and frame (See Figure 4). The main idea which allows us to train the attention network, is that from a pooled spatial features weighted by the latent variable attention, p_{TFA}, we should be able to reconstruct the input text (user command sentence) words $\mathcal{V} \in \{0,1\}^{|V|}$. Here, we define the weighted pooled features $u \in \mathcal{R}^{d_\phi}$:

$$u = \sum_{i \in k} p_{TFA_i} {\phi_f}_i. \tag{4}$$

Basically, given a video frame and sentence, we force the network to select a few regions of the input frame, and reconstruct the input text just based on the selected regions. As a result, the only way that the network can reconstruct the original input text, is by selecting the relevant regions of the frame:

$$\hat{\mathcal{V}} = \sigma(\tau(u)), \tag{5}$$

where $\tau(.)$ is a multi-layer perceptron. $\hat{\mathcal{V}} \in (0,1)^{|V|}$ contains the predicted set of words. We optimize the entropy loss function $\mathcal{L}_{att} = -\mathcal{V} log(\hat{\mathcal{V}})$.

In Figure 3, we show RGB frames and the masked frame using computed attention p_{TFA}. To mask the RGB frames, we reshape and re-size p_{TFA} (using bi-linear interpolation) to the same size of the input frame (x); followed by smoothing the mask by applying a Gaussian filter on it. We denote the masked RGB input frame with the attention p_{TFA} by m.

3.2. The visual and motor networks

Our architecture follows the generic architecture for the visuomotor policy in Figure 1. It consists of a **Visual Network** sub-module that extracts a primary latent encoding, $\mathbf{z}$, and a **Motor Network** that transforms $\mathbf{z}$ into actions, which in our case are joint angle commands (next state of the robot arms). However, our architecture makes several specific decisions with the aim to take advantage of the available the text description of the current task and the TFA.

3.2.1 Visual Network

The objective of the Visual Network is to create a compact primary latent encoding that captures the important aspects of the current task. An ongoing problem is that the encoding needs to work within a certain limited dimensionality budget. Intuitively, general purpose visual features extracted from the image would waste space by encoding aspects of the image that are not relevant to the task. On the other hand, focusing only on the attention field may ignore parts of the image that are important for the task. For instance, in Figure 3- bottom right masked frame, the robot arm itself is not visible.

Our proposed architecture for the visual network, shown in Figure 2, incorporates several techniques that allows it to learn a representation that efficiently encodes the parts of the input that are relevant to the *current task*. The overall architecture follows the idea of a VAE-GAN [25]: it is composed of an encoder, a generator and a discriminator. The *Primary Latent Encoding ($\mathbf{z}$)* is extracted from the output of the visual encoder (E).

The visual network receives a raw frame $\mathbf{x}$ and a one-hot representation of the user command (input sentence), denoted by $I_c \in \{0,1\}^{|V|}$. In fact, I_c is indicates which words of the dictionary are appearing in the textual input command. We assume that $\mathbf{z} \sim \mathcal{N}(\mu_z, \sigma_z)$, and:

$$[\mu_\mathbf{z} | \sigma_\mathbf{z}] = E(x, I_c), \tag{6}$$

where μ_z, $\sigma_z \in \mathcal{R}^{d_z}$, and d_z is the length of the Primary Latent Encoding (z). In fact, E is a multi-layer convolutional neural network with a $2d_z$ dimensional vector which splits into μ_z and σ_z.

The generator, (G), takes the Primary Latent Encoding $\mathbf{z}$ as input, and produces two images, a reconstruction frame, and a reconstructed frame masked with attention ("Fake Frame" and "Fake masked frame" in Figure 2). Notice that a novel aspect of our proposed architecture is that the generator does not only create a reconstruction of the input, $\mathbf{x}'$, but also an approximation of the faked masked frame, m'.

Unlike traditional GAN discriminators, the discriminator D employed in our architecture performs a more complicated classification [26]. Masked and unmasked frames(m/m', x/x') are both inputs to the discriminator, and it classifies the objects (s) and color (c) of the object of interest, as well as whether the input was fake or real. The discriminator has two outputs of lengths of $|s|+1$ and $|c|+1$. $|s|$ and $|c|$ are respectively the number of colors and objects in the vocabulary $|V|$ and the "$+1$" is for the "fake" class. We make the set of s and c tags by parsing all the input sentences (user's textual commands) in the training.

3.2.2 Motor Network

The motor network in our architecture (see Figure 2) contains both recurrent and stochastic components. It takes as input the primary latent encoding, $\mathbf{z}$, which is processed through a 3-layer LSTM network with skip connections [27]. Note that the memory cells of LSTMs get updated through the time by doing the task (frame by frame). The output of the final LSTM layer is fed into a mixture density network (MDN) [28]. MDN provides a set of Gaussian kernels parameters namely μ_i, σ_i and the mixing probabilities $\alpha_i(x)$, all $\in \mathcal{R}^{|J|}$, and $1 \leq i \leq N_G$. Here, $|J|$ is the number of robot joints (specific to the robot) and N_G is the number of Gaussian components. The $|J|$-dimensional vector describing the next joint angles is sampled from this mixture of Gaussians. We provide the detailed architectures of D, G, E, and motor sub-networks in the Supplementary Material.

3.3. Loss Function and Training

In this section, we describe the discriminator loss function $\mathcal{L}_D$, and the generator loss function $\mathcal{L}_G$. All the parameters in the Discriminator have been optimized to minimize $\mathcal{L}_D$, and parameters of the visual Encoder, Generator, and Motor network are optimized by the loss value $\mathcal{L}_G$ in a GAN training manner. In following, to prevent repetition of equations, we use the unifying tuples $\mathcal{X}' = (x', m')$ and $\mathcal{X} = (x, m)$ as fake and real data respectively. To clarify, $(x', m') = G(\mathbf{z} \sim E(x, I_c))$, while x is the real frame from RGB camera, and m is the masked real frame by the teacher network (Section 3.1).

3.3.1 Discriminator Loss

If the discriminator D is receiving real data $\mathcal{X}$, it needs to classify the object and color contained in the user's textual command input:

$$\begin{aligned}\mathcal{L}_{real} = &- \mathbb{E}_{\mathcal{X},s\sim p_{data}}[\log(P_D(s|\mathcal{X}))] \\ &- \mathbb{E}_{\mathcal{X},c\sim p_{data}}[\log(P_D(c|\mathcal{X}))],\end{aligned} \tag{7}$$

where P_D is the class probabilities produced by the discriminator for both colors and objects. Similarly, if D receives $\mathcal{X}'$, it should classify them as fake:

$$\begin{aligned}\mathcal{L}_{fake} = &- \mathbb{E}_{\mathcal{X}'\sim G}[\log(P_D(|s|+1|\mathcal{X}'))] \\ &- \mathbb{E}_{\mathcal{X}'\sim G}[\log(P_D(|c|+1|\mathcal{X}'))].\end{aligned} \tag{8}$$

Finally, if D receives raw and masked faked frames, generated by G with the latent representation $\mathbf{z} \sim \mathcal{N}(0,1)$:

$$\begin{aligned}\mathcal{L}_{noise} = &- \mathbb{E}_{z\sim noise}[\log(P_D(|s|+1|G(z)))] \\ &- \mathbb{E}_{z\sim noise}[\log(P_D(|c|+1|G(z)))].\end{aligned} \tag{9}$$

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

3.3.2 Generator Loss

The Generator (G) must reconstruct a real looking frame and masked frame by attention that contains the object of interest. In fact, G tries not only to look real, but also presents the correct object in both of its outputs. Hence, it has to fool the discriminator which tries to distinguish between fake frames and different objects and colors:

$$\begin{aligned}\mathcal{L}_{GD} = &- \mathbb{E}_{\mathcal{X}',s\sim p_G}[\log p_D(s|\mathcal{X}')] \\ &- \mathbb{E}_{\mathcal{X}',c\sim p_G}[\log p_D(c|\mathcal{X}')].\end{aligned} \tag{10}$$

The training of GANs is notoriously unstable. A possible technique to improve stability is *feature matching* [29]– forcing G to generate images that match the statistics of the real data. Here, we use features extracted by the last convolution layer of D for this purpose and we call it $f_D(x)$. The generator must produce outputs that have similar f_D representation to real data. We define the loss term $\mathcal{L}_{fea}$ as a distance between the real inputs x/m and generated ones x'/m' features [26]:

$$\mathcal{L}_{fea} = ||f_D(x) - f_D(x')||^2 + ||f_D(m) - f_D(m')||^2. \tag{11}$$

To regularize the Primary Latent Encoding ($\mathbf{z}$), we minimize the KL-divergence between $\mathbf{z}$ and $\mathcal{N}(0,1)$:

$$\mathcal{L}_{prior} = D_{\mathrm{KL}}(E(x, I_c) \,||\, \mathcal{N}(0,1)). \tag{12}$$

Additionally, a reconstruction error of "fake" Frame/Masked generated by G is defined by:

$$\mathcal{L}_{rec} = ||x' - x||^2 + ||m' - m||^2. \tag{13}$$

Motor Network Loss: The motor loss is calculated according to the MDN negative log-likelihood loss formula over the supervised data based on the demonstrations (behavioral cloning loss):

$$\mathcal{L}_{motor} = -log\left(\sum_{i=1}^{N_G} \alpha_i(x) \cdot P_{\sim\mathcal{N}(\mu_i,\sigma_i)}(J)\right). \tag{14}$$

Finally, we write the Generator loss as $\mathcal{L}_G = \mathcal{L}_{DG} + \mathcal{L}_{rec} + \mathcal{L}_{prior} + \mathcal{L}_{motor}$.

4. Experiments

We collected demonstrations for the tasks of picking up and pushing objects using an inexpensive Lynxmotion-AL5D robot. We controlled the robot using a PlayStation controller. For each task and object combination we collected 150 demonstrations. The training data consists of joint-angle commands plus the visual input recorded in 10 fps rate by a PlayStation Eye camera mounted over the work area. The training data thus collected was used to train both the Visual and the Motor Networks. Note that this robot does not have proprioception – any collision or manipulation error needs to be detected solely from the visual input.

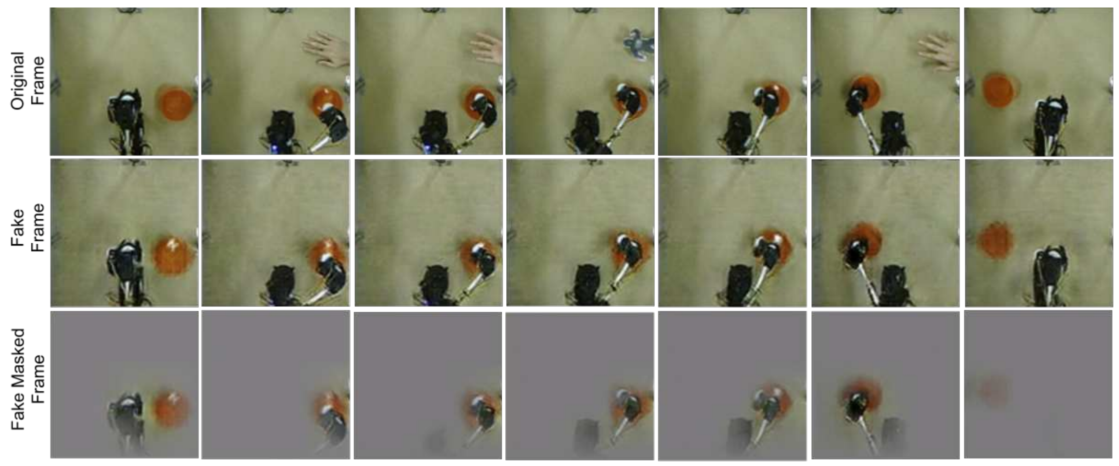

Figure 5. An execution of the pushing task with the sentence "Push the red bowl from right to left". Top row: original input image, middle row: fake frame generated by the **Generator(G)**, bottom row: fake masked image with TFA generated by **G**. You can compare the fake masked frames presented in this figure with attention maps generated by the **teacher network** in Figure 3. Notice that visual disturbances such as the hand and the gorilla do not appear in the reconstructed image.

Textual Command Sentences	**Pick up ...**							**Push ... from left to right**					
	Red Bowl	White Towel	Blue Ring	Black Dumbbell	White Plate	Red Bubbles	Mean pick up	Red Bowl	White Plate	Blue Box	B/W QR-box	Mean Push	Mean
Method	**Benign Condition**												
Just Encoder (%)	20	20	0	40	0	10	15.0	40	10	0	0	12.5	14.0
Traditional VAE (%)	60	60	20	20	50	30	40.0	50	**60**	**30**	30	42.5	41.0
(w/o TFA) (%)	70	50	30	40	60	10	43.3	80	**60**	10	20	42.5	43.0
with TFA (%)	**80**	**80**	**60**	**50**	**80**	**40**	**65.0**	**100**	**60**	**30**	**60**	**62.5**	**64.0**
	With Disturbance												
(w/o TFA) (%)	10	10	0	0	0	0	3.3	0	30	0	0	7.5	5.0
with TFA (%)	**70**	**80**	**60**	**60**	**40**	**40**	**58.3**	**90**	**50**	**30**	**50**	**55.0**	**57.0**

Table 1. The upper half of the table shows the rate of successfully performing the desired manipulation with different sentence commands. The model with **TFA** has superior results to a model without it [7]. We also train a version of our model without the Discriminator, named *Traditional VAE*. The model trained without D cannot effectively perform the manipulations since the adversarial loss helps to learn rich Primary Latent Variable ($\mathbf{z}$). Also, in *Just Encoder experiment*, we just use the Encoder as the visual network. The lower half of the table shows the rate of successfully performing the desired command, while being disturbed by an external agent. The model with **TFA** is by far better than a model without it [7] in all cases.

4.1. Performance under benign conditions

The first set of experiments studies the performance of the visuomotor controller under benign conditions, that is, under situations when the robot is given a textual command, I_c in Sec. 3.2.1, and it is left alone to perform the task in an undisturbed environment. To compare our approach against a baseline, we have reimplemented and trained the network described in [7], which can be used in the same experimental setup, but it does not feature a task focused visual attention. Note that the success rates are not directly comparable with [7], due to the more complex objects used here and the different camera position and environment of our robot. We trained the [7] model on our own dataset, tuned its hyper-parameters and also tried to get the best possible results by adding all the loss terms explained in Sec. 3.2.

Table 1 compares the performance of the four approaches for all the tasks, averaged over 10 tries each. We note that the proposed architecture using "TFA" outperforms the "w/o TFA" on all tasks. As an ablation study, we remove the discriminator and train the system as a traditional VAE (compared to VAE-GAN). Also, in another experiment we trained the E just by using the motor network loss without any GAN. We confirm the contribution of the adversarial loss and the GAN network to produce a rich primary latent variable $\mathbf{z}$. We observe that not having the adversarial loss will reduce the sharpness of the reconstructed images and fade out the details. Note that the model without adversarial loss fails to manipulate objects that require precise positioning like the black dumbbell or the blue ring, however, it can push the white plate much better as the plate is a big symmetric object. Please refer to the supplementary materials to compare the reconstructed images with and without the adversarial loss.

4.2. Recovery after disturbance

In the second series of experiments, we investigate the controller's ability to recover from a physical and visual disturbance. We are comparing the baseline model and our model which uses TFA. Physically disturbing means to disturbed the robot either by (a) pushing the object just when the robot was about to pick it up or (b) forcefully taking away the object from the robot after a successful grasp. For the push tasks, we bring in one or two hands into the scene (Figure 5). We make different visual disturbances by bringing in the hand in random positions, waving it, sometimes covering whole top part of the scene. In some cases we even put other random objects like a paper gorilla.

Under the described situations we count as success, if the robot notices the disturbance and recovers by successfully redoing the task. We remind the audience of the paper that due to the limitations of the Lynxmotion-AL5D robot, the *only* way the robot can detect the disturbance is through its visual system.

Table 1 shows the experimental results for scenarios with physical/visual disturbance. We notice that the results here are drastically better than the baseline. In the absence of TFA, the recovery rate is close to zero. In most cases, after loosing the object, the robot tried to execute the manipulation without noticing that it does not grasp the object. With the help of TFA, however, the robot almost always notices the disturbance, turns back and tries to redo the grasp. This phenomena is illustrated in our supplementary material video. Averaged over all the objects, the recovery rate is only 5% for the baseline policy in pickup and push tasks, while it is **57%** for the policy with the TFA (see Tables 1). Note that physical disturbance doesn't necessarily drop the robot's success rate since disturbing the robot occurs only when it is about to successfully perform the task, therefore the robot's success rate with and without the physical disturbance are not comparable. In other words, robot starts doing the task, a human judge decides if the robot is doing well and if it is, the human judge starts to disturbing the robot. We discard any tries that the robot is likely to fail even without disturbance.

The disappearing gorilla: The proposed architecture allows us to ignore many of the possible visual disturbances. Experiments comparing the architecture to one without TFA confirm that this is indeed the case. Another way to study whether the policy ignores the visual disturbance is to reconnect the generator during test time as well, and study the reconstituted video frames (which are a good representation of the information content of primary latent encoding). Figure 5 shows the input video frames (first row), the reconstructed video frames (second row) and the generated masked frames (third row). While the robot was executing the task of pushing the red bowl to the left, we added some disturbances such as waving a hand or inserting a cutout gorilla figure in the visual field of robot. Notice that in the reconstructed frames, the hand and the gorilla disappear, while the subject matter is reconstructed accurately. As these disturbing visual objects are ignored by the encoding, the task execution proceeds without disturbance. While we must be careful about making claims on the biological plausibility of the details of our architecture, we note that the overall effect implements a behavior similar to the selective attention experiments[1] of Chabris and Simmons [1], purely as a side effect of an architecture implemented for a completely different goal.

5. Conclusion

In this paper, we proposed a method for augmenting a deep visuomotor policy learned from demonstration with a task focused visual attention model. The attention is guided by a natural language description of the task – it effectively tells the policy to "Pay Attention!" to the task and object at hand. Our experiments show that under benign situations, the resulting policy consistently outperforms a related baseline policy. More importantly, paying attention has significant robustness benefits. In severe adversarial situations, where a bump or human intervention forces the robot to miss the grasp or drop the object, we demonstrated through experiments that the proposed policy recovers quickly in the majority of cases, while the baseline policy almost never recovers. In the case of visual disturbances such as moving foreign objects in the visual field of the robot, the new policy is able to ignore these disturbances which in the baseline policy often trigger erratic behavior.

Future work includes attention systems that can simultaneously focus on multiple objects, shift from object to object according to the requirements of the task, and work in severe clutter.

Acknowledgments: This work had been supported in part by the National Science Foundation under grant numbers IIS-1409823 and IIS-1741431. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[1]https://youtu.be/vJG698U2Mvo

References

[1] C. Chabris and D. Simons, *The invisible gorilla: And other ways our intuitions deceive us*. Harmony, 2010. 2, 8

[2] 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. 2

[3] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, p. 484, 2016. 2

[4] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017. 2

[5] 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," *International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018. 2

[6] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, "Deep spatial autoencoders for visuomotor learning," *arXiv preprint arXiv:1509.06113*, 2015. 3

[7] 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 Conference on Robotics and Automation (ICRA-2018)*, 2018, pp. 3758 – 3765. 3, 7

[8] S. James, A. J. Davison, and E. Johns, "Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task," *arXiv preprint arXiv:1707.02267*, 2017. 3

[9] 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. 3

[10] Y. Duan, M. Andrychowicz, B. Stadie, O. J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba, "One-shot imitation learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 1087–1098. 3

[11] C. Devin, P. Abbeel, T. Darrell, and S. Levine, "Deep object-centric representations for generalizable robot learning," *arXiv preprint arXiv:1708.04225*, 2017. 3

[12] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proc. of the Nat'l Conf. on Artificial Intelligence (AAAI-2011)*, San Francisco, CA, August 2011, pp. 1507–1514. 3

[13] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *IEEE Int'l Conf. on Robotics and Automation (ICRA-2017)*, 2017, pp. 2786–2793. 3

[14] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," in *Advances in neural information processing systems*, 2015, pp. 2746–2754. 3

[15] 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. 3

[16] 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. 3

[17] 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-2016)*, 2016, pp. 21–29. 3, 4

[18] A. Mazaheri, D. Zhang, and M. Shah, "Video fill in the blank using LR/RL LSTMs with spatial-temporal attentions," in *Proc of IEEE Int'l Conf. on Computer Vision (ICCV-2017)*, Oct 2017. 3

[19] D. Yu, J. Fu, T. Mei, and Y. Rui, "Multi-level attention networks for visual question answering," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR-2017)*, 2017, pp. 4187–4195. 3

[20] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg, "MAttNet: Modular attention network for referring expression comprehension," in *Proc. of IEEE Conf. on on Computer Vision and Pattern Recognition (CVPR-2018)*, 2018. 3

[21] S. Lawrence, C. L. Giles, and A. C. Tsoi, "Lessons in neural network training: Overfitting may be harder than expected," in *Proc. of the Fourteenth Nat'l Conf. on Artificial Intelligence (AAAI-97)*, 1997, pp. 540–545. 4

[22] C. Bucilu, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. of the 12th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 2006, pp. 535–541. 4

[23] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015. 4

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 4

[25] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *arXiv preprint arXiv:1512.09300*, 2015. 5

[26] 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. 5, 6

[27] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013. 6

[28] C. M. Bishop, "Mixture density networks," Aston University, Tech. Rep., 1994. 6

[29] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "CVAE-GAN: fine-grained image generation through asymmetric training," *arXiv preprint arXiv:1703.10155*, 2017. 6