Supplementary Material for 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

In the main manuscript, we propose a deep visuomotor policy that benefits from Task Focused visual Attention (TFA) and show that it outperforms policies using a task-independent visual network. More importantly, we show that the TFA has a very large impact when there are visual and physical disturbances in the environment.

In these supplementary materials we provide more details about the proposed architecture, and experimental settings. Moreover, we demonstrate some mid-level outputs of the proposed method as a comprehensive study. Additionally, we provide a video demo that includes some examples of our data collection process, experiments in benign condition, and experiments in presence of physical/visual disturbance.

Experimental Settings

In this subsection, we provide additional details about our collected dataset and the experimental settings. Figure 1 shows all the objects used in our experiments listed in Table 1 of the main manuscript.

The experimental protocol we followed was as follows. The robot always starts from a fixed starting position and the task is considered successful if the robot performs the required task and returns back to the starting position. During testing, the robot has to finish the task within *2 minutes*. A human judge observes the robot during the performance and decides if it has been successful or not.

For the experiments with physical/visual disturbance, the human judge decides if the robot is likely to succeed to perform the task and then provides the disturbance. We stop and do not consider experiments where the robot is clearly failing even without the disturbance (for example, if it doesn't get close to the object correctly or accidentally push the object to an out of access point before the disturbance starts). Thus, our measurements on the recovery from disturbance, only consider the cases when the need for recovery was caused by the disturbance.

*Authors contributed equally.

Figure 1. Objects used in all the picking up (top) and pushing (bottom) tasks experiments. (see Table 1 of the main manuscript)

Teacher Network Details

In Section 3.1 of the main manuscript, we describe the Teacher Network for visual attention. The masked frames produced by the teacher network are considered as "real" masked frames (denoted by m). Also, the teacher network is pre-trained separately (has its own loss function and optimizer), and helps the proposed network to produce the Primary Latent Variable z that is a rich representation of the visual world and is robust to disturbance. We consider the visual attention produced by the teacher network as ground

Hyper-parameters	Value
$\|V\|$	20
d_v	200
d_h	200
k	196
d_ϕ	512
d_ψ	200

Table 1. Hyper-parameter values in our implementation of teacher network, described in the Section 3.1 of the main manuscript.

truth attention.

In Table 1 we show the hyper-parameter values used in our implementation of the teacher network.

As explained in Section 3.1, we pre-train the teacher network by reconstructing the set of words used in the textual input sentence (refer to the $\mathcal{L}_{att}$ loss in the main manuscript). Figure 2 demonstrates a few examples of the trained teacher network and shows how good it can select the words appeared in the input sentence just based on the visual features of the attended spatial regions (Equation 5 of the main manuscript). We observe that it can predict all the object names and colors correctly. In some cases, since the input is only one frame, it is impossible to predict the verb words (push or pickup in our experiments). For example, in top-right example of Figure 2, since the robot arm has not reached the object, it is not easy to say if it going to be a push or pick-up task. On the other hand, in the top-left example, the robot arm has reached next to the object and it is clear that it is going to push it, and the teacher network also correctly predict the verb "push".

Detailed Architectures

In this section, as promised in the main manuscript, we show the detailed architectures of our Encoder (Figure 3), Generator (Figure 4), Discriminator (Figure 5) and Motor network (Figure 6).

Note that, except the visual Teacher Network which is trained separately, all other modules including Encoder, Generator, Discriminator, and Motor Network are trained end-to-end.

Attention and Disturbance

In the main manuscript, we show that our proposed approach with Task-Focused visual Attention (TFA) makes the robot policy robust to various types of visual and physical disturbances. Table 1 of the main manuscript shows that the average performance of the model without TFA is about 50% less than the proposed network. Here, by visualizing the generated images from the two named experiments, we qualitatively show the reasons behind robustness and effectiveness of visual attention during the disturbance.

Figure 7 shows the original and reconstructed frames from the "w/o TFA" experiment. The frames show the sequence of a task when two other objects, namely a human hand and an eyeglass box, enter the scene and disturb the internal representation (starting frame 8). We notice that the reconstructed frames become very blurry and inaccurate. For example, from frame 23 to 30, the object of interest is completely missing in the reconstruction. We conjecture that the disturbance forced the primary latent encoding z to move to an unseen state from which the generator cannot reconstruct meaningful images.

Figure 8 reproduces same disturbance scenario as in Figure 7 but using the model with TFA. We notice that the attention disregards the obstacles and disturbances and the quality of reconstructed frames do not drop drastically. Also, we see that the disturbing objects such as the hand are removed from reconstructions.

Attention Examples

Figure 9 illustrates several frames for a given textual command in each row. The first column of each row shows the original frames (denoted by x in the main manuscript), and the masked frames produced by the teacher network (denoted by m in the main manuscript). The second and third columns show the fake (reconstructed) frame and the fake masked frame generated by the generator (x' and m' in the main manuscript); however, the third column shows the results out of generator when it is trained merely based on a reconstruction loss, without any discriminator. The quality difference between the second and third columns reconstructions explains the performance difference between "traditional VAE" and other experiments with the "VAE-GAN" setting (see Table 1 of the main manuscript).

We notice that in some cases, the attention produced by the generator is even better than the teacher network attention. For example, compare the masked frames of the first and second columns of the second and fourth examples in Figure 9. We believe that this phenomena is due to the rich Primary Latent Variable, z that our network learns. In fact, the fake masked frame must be rich enough that the discriminator predict the correct object and color of the task and it provides some complementary information to the Encoder.

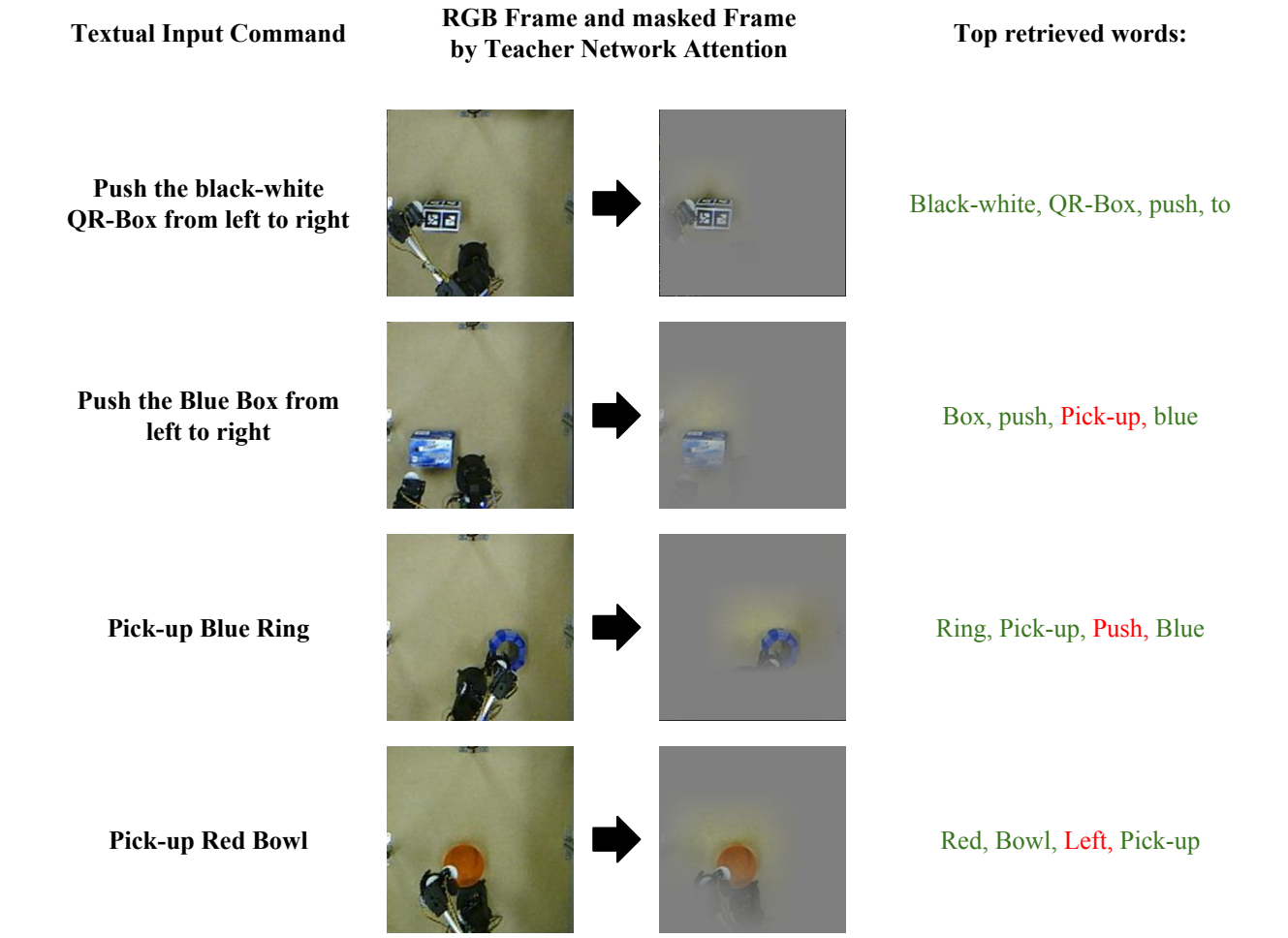

Figure 2. Here we show how well the trained teacher network can select correct words about a task being performed based on the generated visual attention. We show the textual input command, one frame, and the masked frame using the attention computed by the teacher network, and the top retrieved words by teacher network. Basically, we sort the scores $\hat{\mathcal{V}}$ in Equation 5 of the main manuscript, and show the top 4 words. The green/red color indicates the words which are/aren't in common with the textual input command.

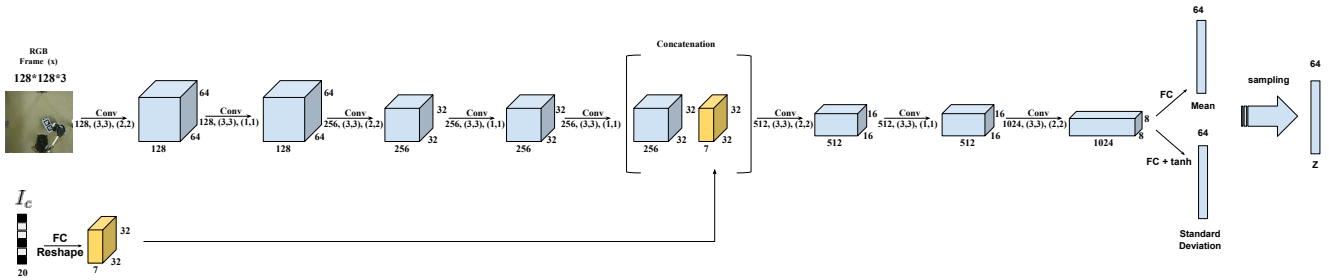

Figure 3. Encoder Architecture used in our framework. The textual command I_c and RGB frame are the inputs to the Encoder. The Primary Latent Variable z is the output of the Encoder. Here, we show all the layers used in the Encoder including Convolution and Fully-Connected (FC) layers. We also indicate the number of filters, kernel size, and the stride of each convolution. Note that, in our implementation, all the convolutions layers are followed by Batch-Normalization and a Leaky Relu.

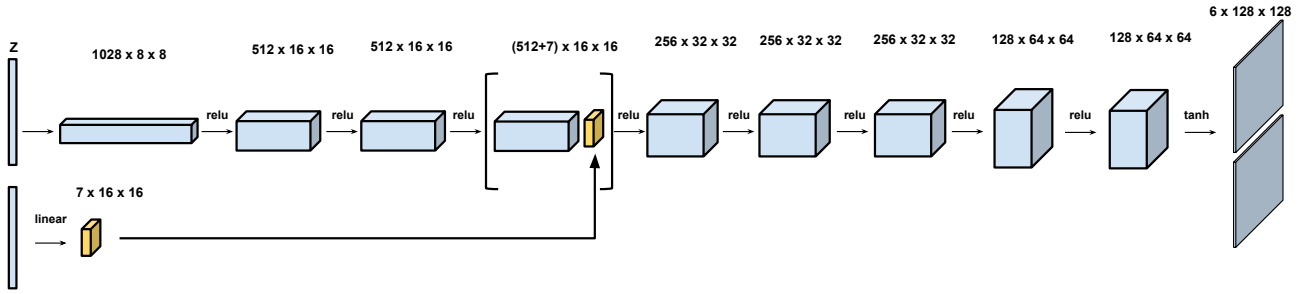

Figure 4. Generator Architecture used in our framework. The input to the generator is the Primary Latent Variable z and textual Encoding I_c, which is produced by the Encoder (Figure 3). And the outputs of generator are two images, a fake frame (x') and a fake masked frame (m') respectively. We use multiple layers of Deconvolutions to generate an image out of the input vector. We show the number of filters, kernel size and the stride of each Deconvolution. All the Deconvolutions are followed by Batch-Normalization and Relu.

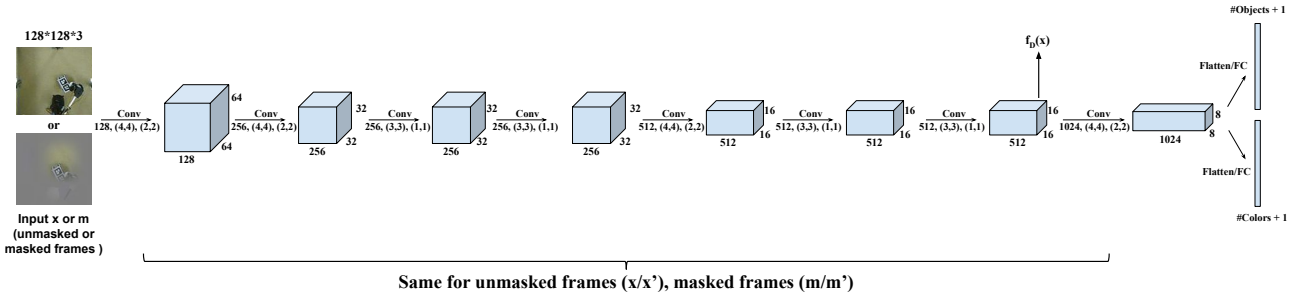

Figure 5. Discriminator Architecture used in our framework. Our proposed discriminator architecture not only distinguishes between the fake and real frames, but also classifies the object type and the color of the object being manipulated. Convolution layers are shared for both types of inputs (masked or unmasked) but the last Fully Connected (FC) layers are separate for each kind of input.

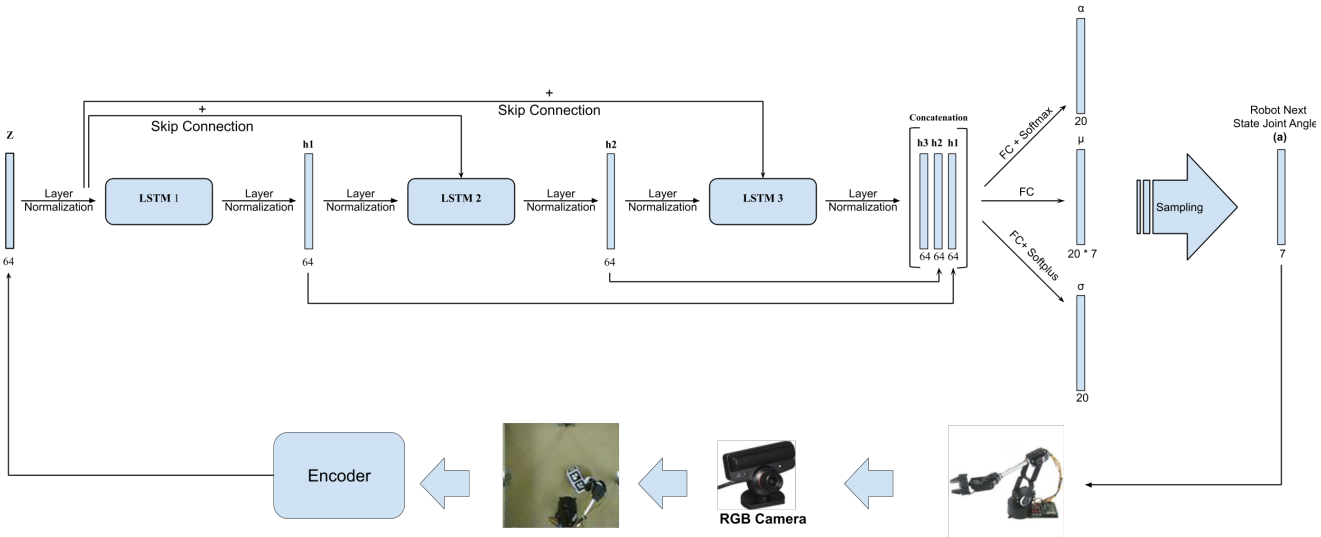

Figure 6. Motor Network Architecture used in our framework. Given the Primary Latent Variable z, the motor network predicts the next state of the robot, and produces 7 numbers (corresponding to 7 joint angles of the robot) to move the joints of the robot. We use 3 stacked layers of LSTMs with skip connections. Also, we use layer normalization in between LSTMs. We concatenate the outputs of all the LSTMs and generate the μ, σ, and the mix coefficients α for the Mixture Density Network (MDN) described in the main manuscript. We use 20 Gaussians for the MDN of our implementation. The states of all LSTMs get updated frame by frame. In fact, after each move of the robot, a new frame is captured by the RGB camera, fed to the Encoder, and then the next z vector is fed back to the Motor Network. Each frame corresponds to one time-step for LSTMs.

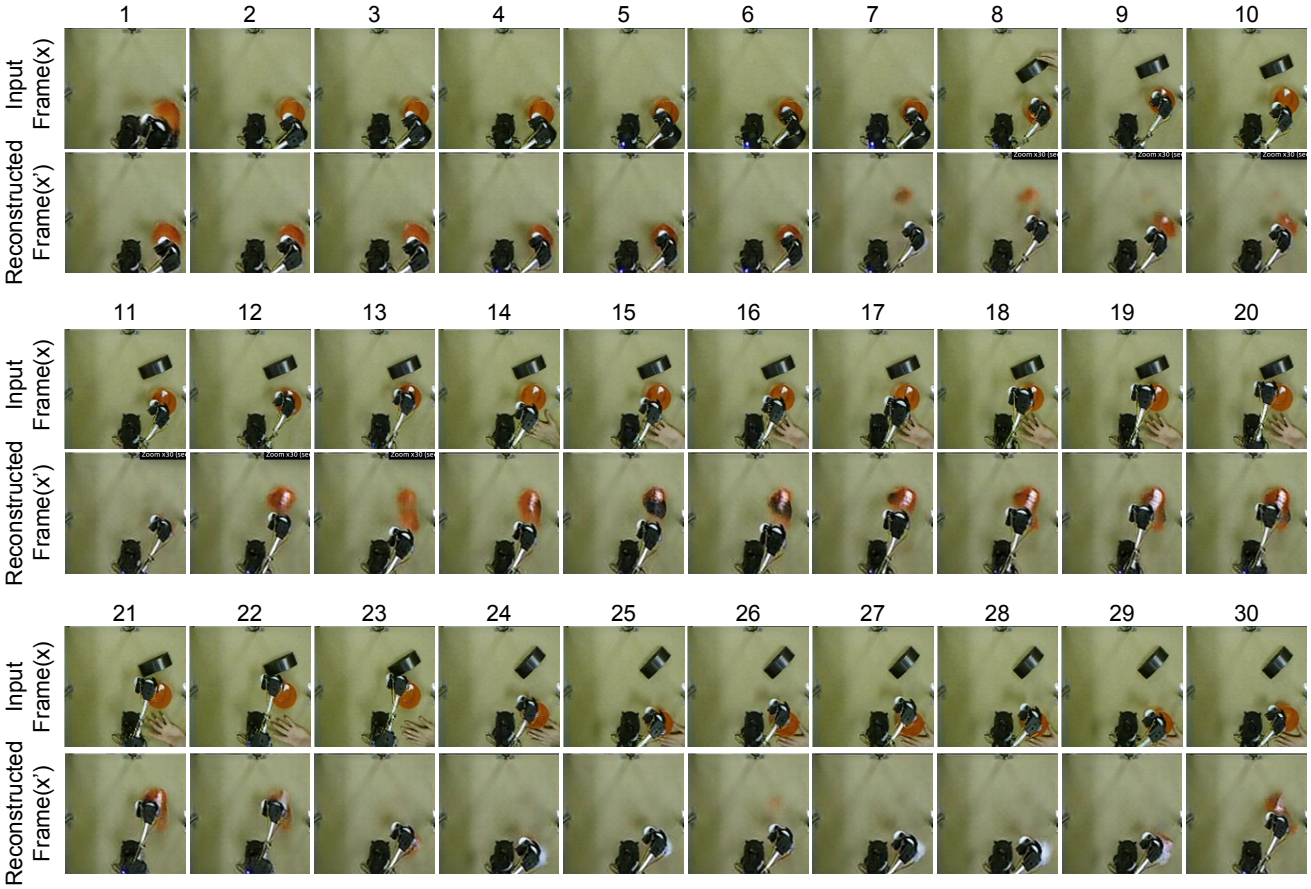

Figure 7. A sequence of frames from an experiment with visual disturbance. The textual command for this experiment is: "push the red bowl from left to right". Here, we show the frame reconstructions of the "w/o TFA" model. In many frames like 23-29, the model has failed to reconstruct the input frame properly, showing that the Primary Latent Variable z in this model is not robust to visual disturbance.

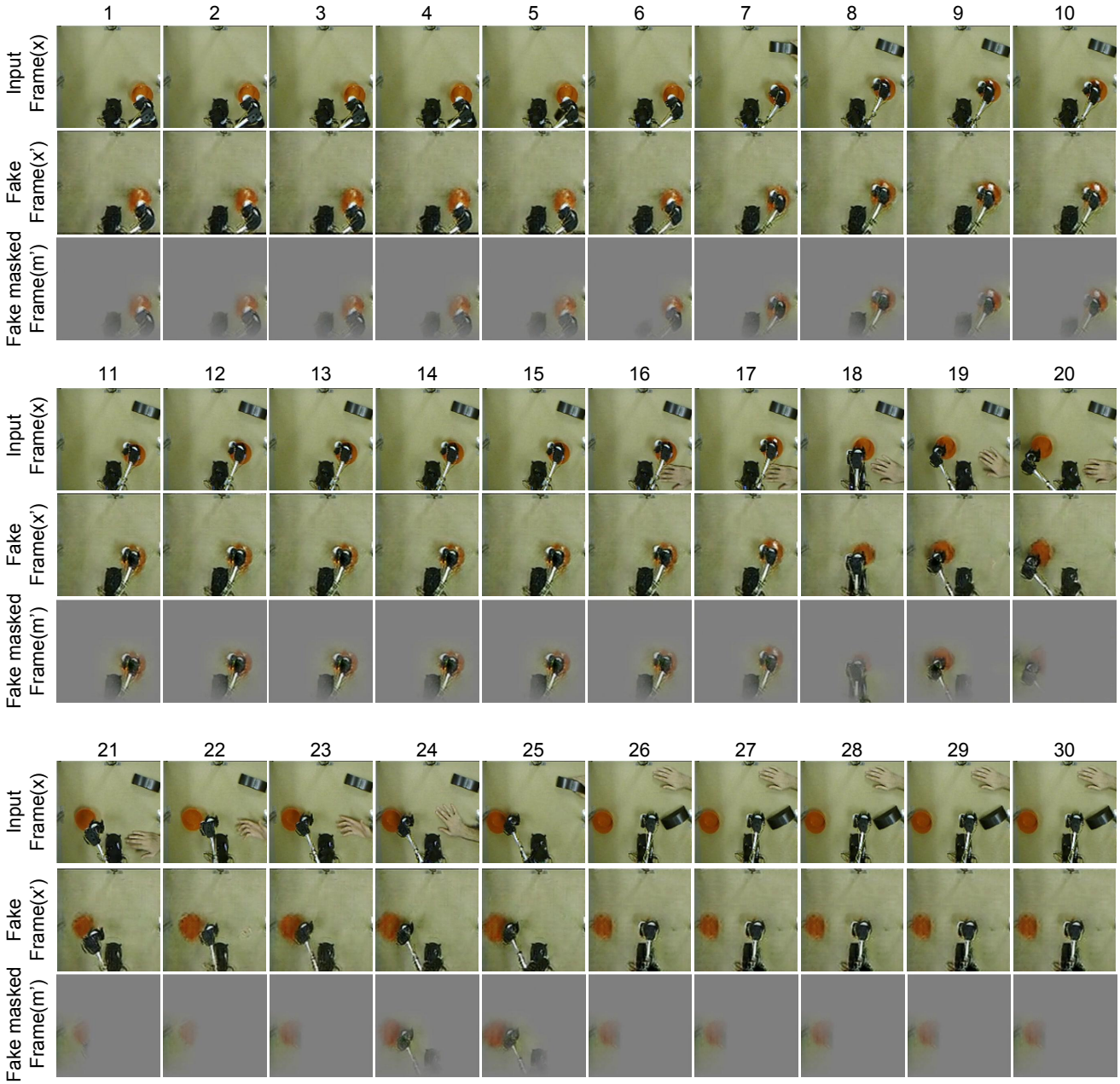

Figure 8. A sequence of frames of an example with a scenario similar to the one in Figure 7, using the TFA-augmented model. We notice that the attention stays on the correct object and the model has a better reconstruction of the object in both of masked and unmasked frames compared to Figure 7.

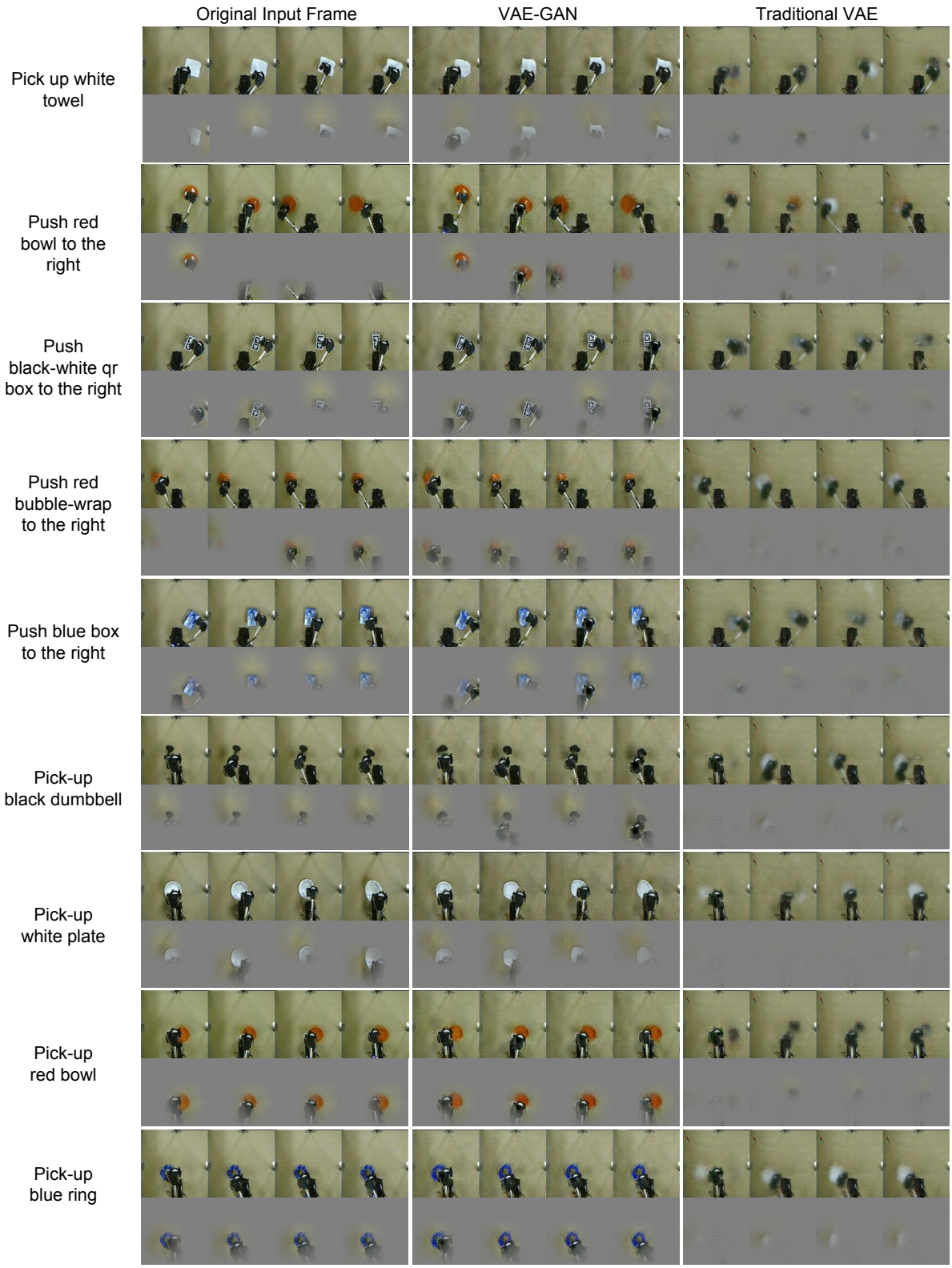

Figure 9. A comparison between the original frame, reconstructed frames by VAE-GAN and Traditional VAE. We also show the real masked frames by attention (using the teacher network), generated fake masked frame by VAE-GAN and Traditional GAN. By comparing the second and third columns of this figure, we can justify the performance drop of the "Traditional VAE" experiment in Table 1 of the main manuscript.