APPENDIX

A. Theoretical Detail

The upper bound of MI:

$$\begin{aligned} \mathrm{I(X;Z)} &= \mathrm{H(X)} - \mathrm{H(X|Z)} = \mathrm{H(Z)} - \mathrm{H(Z|X)} \\ &= \mathbb{E}_{p(x,z)} \log \frac{p(z|x)}{p(z)} \\ &= \mathbb{E}_{p(x,z)} \log \frac{p(z|x)q(z)}{q(z)p(z)} \\ &= \mathbb{E}_{p(x,z)} \log \frac{p(z|x)}{q(z)} - \mathbb{E}_{p(z)} \mathrm{KL}[p(z)||q(z)] \\ &\leq \mathbb{E}_{p(x,z)} \log \frac{p(z|x)}{q(z)} \end{aligned} \tag{7}$$

B. Environment Detail

In the simulation, the agent controls a 7-dof Sawyer arm in the MuJoCo environment. Two tasks are considered: (i) pushing single and multiple objects on a table and (ii) object pickup. Regarding the action space, simulated actions for the pickup task consist of motions along the YZ plane along with gripper control. Whilst for object push, the action space consists of motions in the XYZ plane.

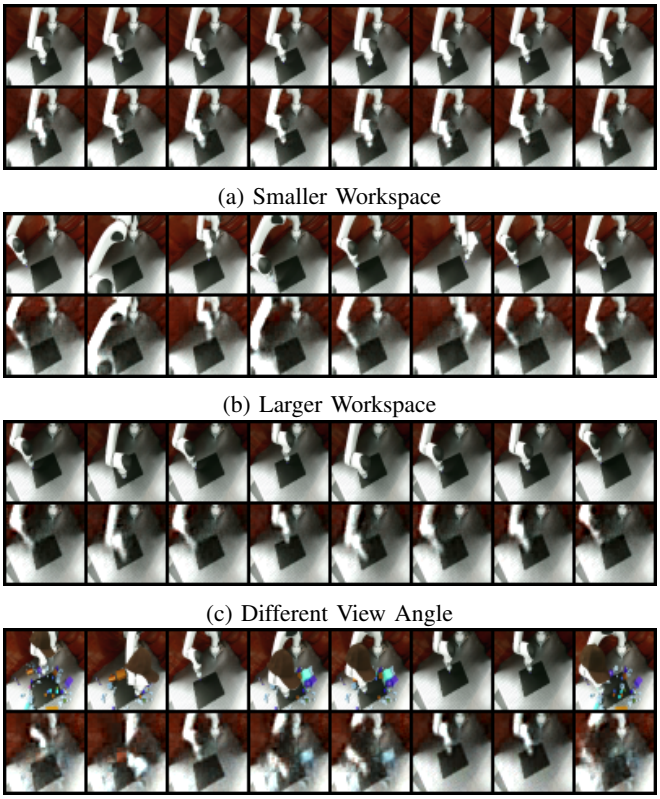
In real-world experiments, we used Elfin 6-dof arm to learning in a reach environment as shown in Fig 1. The action space is XY. Different with the simulated environment, we programmatically wait 2.5 seconds to guarantee the action completion and allow the camera to obtain a stable image after publishing an action to robot.

C. Implementation Details

Hyperparameter and environmental settings are shown in Tab. I. We use the negative ELBO statistics in last testing phase to create auto-tuning methodology without any additional computation cost across experiments. We use the samples from the replay buffer to test the VAE. Auto-tuning in different manipulation environments. Different colors represent different numbers of gradient updates. Performance metrics and completion times for the whole learning process are showing.

D. Additional Examples and Results

Examples of varying diversities from different observation's due to the different workspaces are shown in Fig. 7.



(a) Smaller Workspace

(b) Larger Workspace

(c) Different View Angle

(d) Combination

Fig. 7: Visualization of replay buffer samples (first row) and reconstructed images (second row) of the real robot learning environment.

Hyperparameters	Simulate Navigation	Simulated Push	Simulated Multi-Object Push	Real Reach
Algorithm	SAC	SAC	SAC	SAC
Q network hidden sizes	400,300	400,300	400,300	400,300
Policy network hidden sizes	400,300	400,300	400,300	400,300
Q network and policy activation	ReLu	ReLu	ReLu	ReLu
Exploration Noise	None	None	None	None
RL Batch Size	1024	1024	1024	1024
VAEs Batch Size	64	64	64	64
β for β-VAE	20	20	20	100
Latent Dimension Size	4	4	4	2
VAEs Training Schedule	Always train with 1000 steps			
VAEs Testing Epochs	10 Epochs	10 Epochs	10 Epochs	10 Epochs
Sample Latent Goals From	Ture Prior $q(z)$	Ture Prior $q(z)$	Ture Prior $q(z)$	Ture Prior $q(z)$
Resample Goals	Future and VAEs	Future and VAEs	Future and VAEs	Future and VAEs
Latest Decoder Activation	Sigmoid	Gaussian (Identity)	Gaussian (Identity)	Gaussian (Identity)
Object Shape	No	Puck	Two Cylinders	No
Action Workspace	48×48	$10 \times 10 \times 50(cm^3)$	$10 \times 10 \times 50(cm^3)$	$190 \times 100 \times 5(\mathrm{cm}^3)$
Goal Space	48×48	$10 \times 10 \times 50(cm^3)$	$10 \times 10 \times 50(cm^3)$	$190 \times 100 \times 5(\mathrm{cm}^3)$

TABLE I: RIG hyperparameters for the visual robotic control task. The rest of the hyperparameters are the same as the open-source core with traditional RIG-SAC.

Environment Detail	Simulated Robot Curriculum	Real Curriculum
Action Workspace	$10 \times 10 \times 50(cm^3)$	$190 \times 100 \times 5(cm^3)$

TABLE II: Environment detail of curriculum setup and multitask setup experiments.

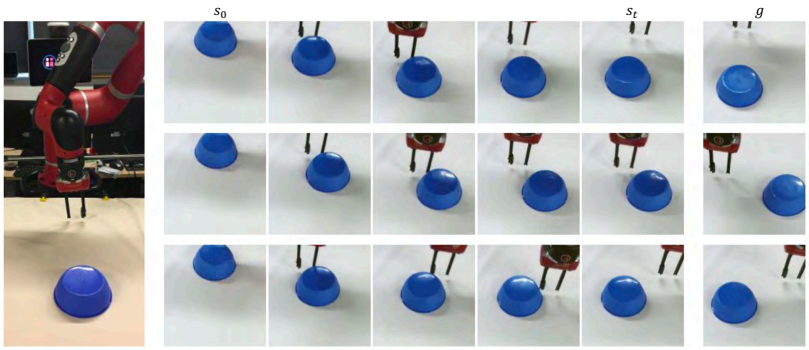


Fig. 8: We use the image from RIG paper directly. The user-specified goal g can be an image having a puck where in a desired position in the environment

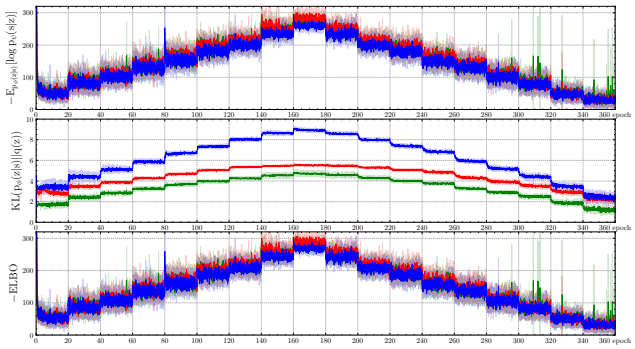


Fig. 9: Experimental results for the fashion MNIST dataset with changing diversity.