

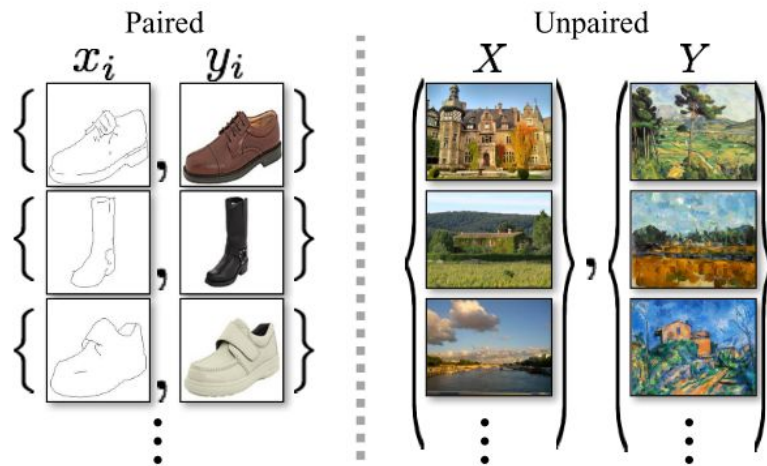
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Jun-Yan Zhu* Taesung Park* Phillip Isola Alexei A. Efros

Presenter: Hai Nguyen

Introduction

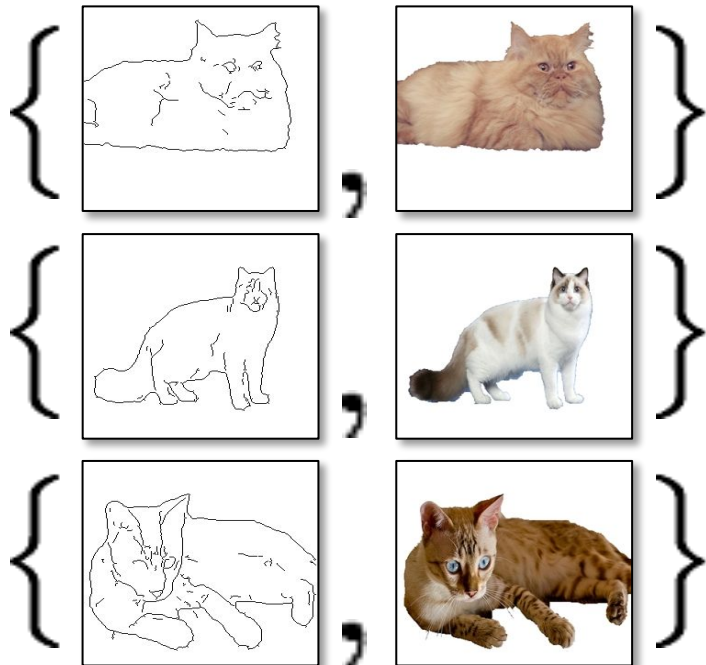
- Focus on image-to-image translation when paired training data is not available
- Assumption: There is some underlying relationship between 2 domains e.g. two different renderings of the same underlying
- Applications in style transfer, object transfiguration, season transfer, photo enhancement



Paired

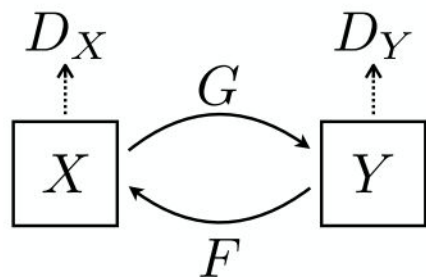
x_i

y_i



- Expensive to collect pairs.
- Impossible in many scenarios.

Formulation

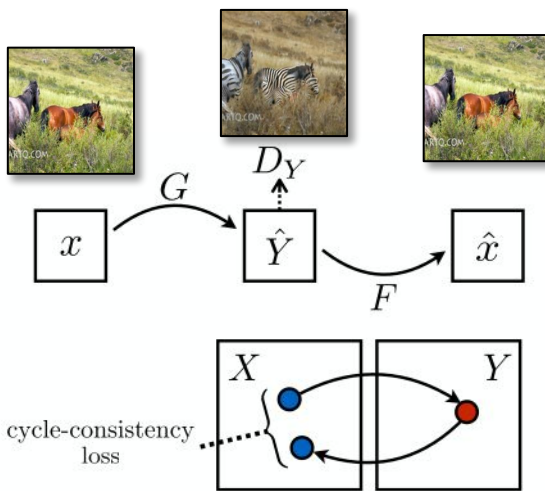


$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y)$$

$$\mathcal{L}_{\text{GAN}}(F, D_X, Y, X)$$

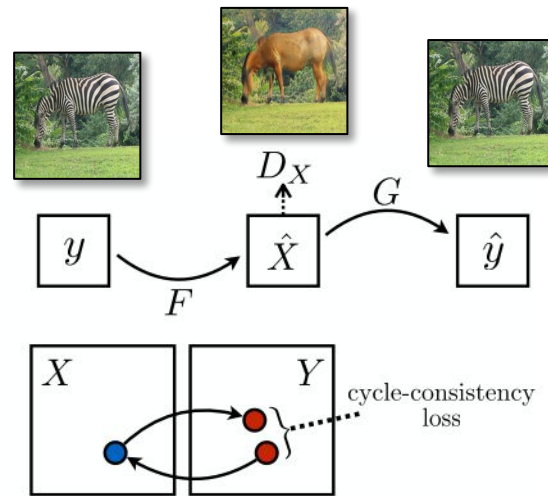
$$\begin{aligned} \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) &= \mathbb{E}_{y \sim p_{\text{data}}(y)} [\log D_Y(y)] \\ &+ \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log(1 - D_Y(G(x)))] \end{aligned}$$

Full objective



Forward cycle-consistency loss

$$\|F(G(x)) - x\|_1$$



Forward cycle-consistency loss

$$\|G(F(y)) - y\|_1$$

$$\begin{aligned} \mathcal{L}(G, F, D_X, D_Y) &= \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) \\ &+ \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ &+ \lambda \mathcal{L}_{\text{cyc}}(G, F), \end{aligned}$$



GAN does not force output to correspond to input

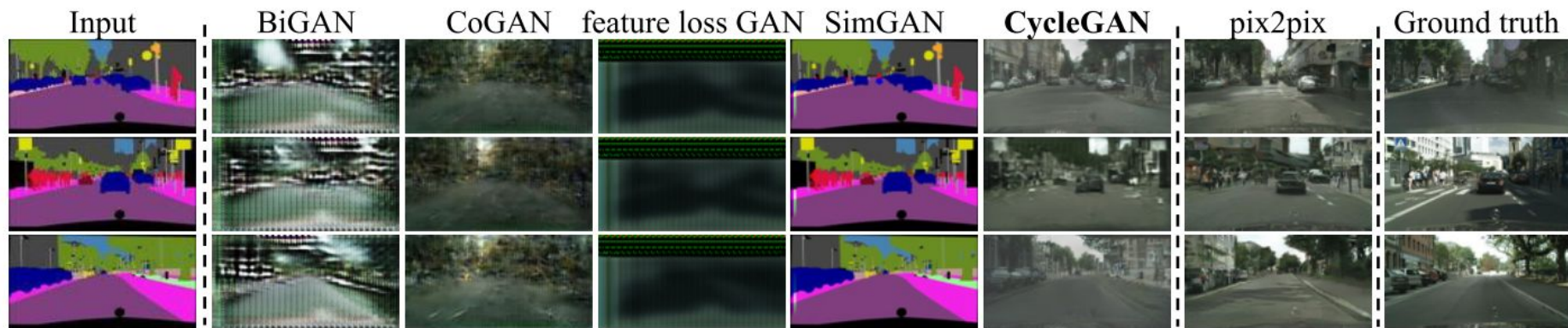
The generator can output the same input with different inputs as long as the discriminator cannot distinguish fake from real



Mode collapse

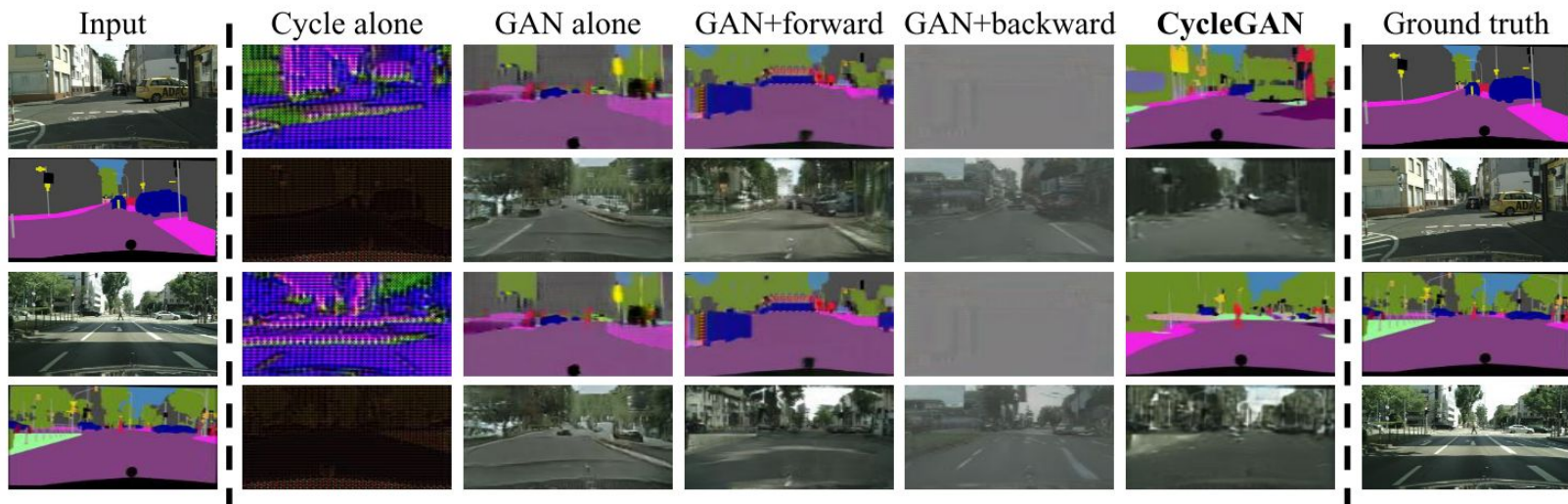


Results



- CoGAN: G_X and G_Y have weights tied on the first few layers
- SimGAN: GAN + a different regularization term
- Feature Loss GAN: Like SimGAN but the regularization is computed over deep image features using a VGG-16
- BiGAN: GAN in the latent space
- Pix2Pix: Train on paired data

Ablation Study



Loss	Per-pixel acc.	Per-class acc.	Class IOU
Cycle alone	0.22	0.07	0.02
GAN alone	0.51	0.11	0.08
GAN + forward cycle	0.55	0.18	0.12
GAN + backward cycle	0.39	0.14	0.06
CycleGAN (ours)	0.52	0.17	0.11

Style Transfer



Photograph
@ Alexei Efros



Monet



Van Gogh



Cezanne



Ukiyo-e

Input



Monet



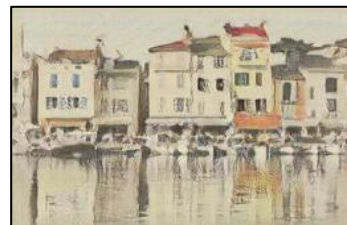
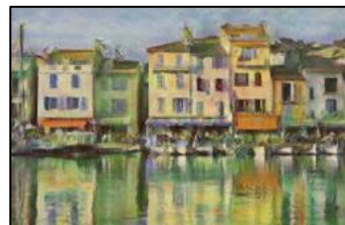
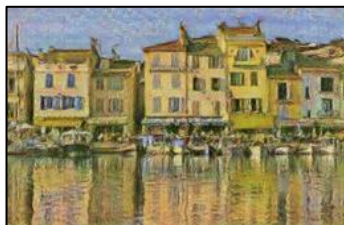
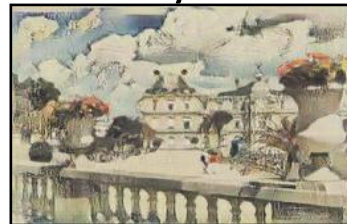
Van
Gogh



Cezanne



Ukiyo-e



Failure Cases

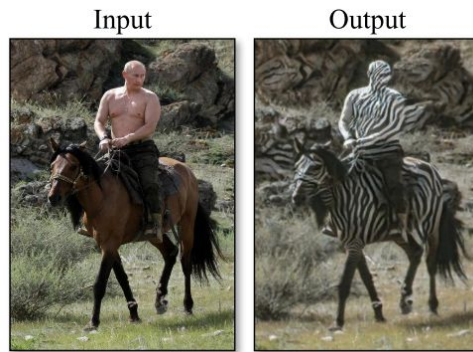
- Geometric changes, possibly due to the tailored generator that favors appearance changes
- With unseen data during training i.e. human riding a horse



dog → cat



cat → dog



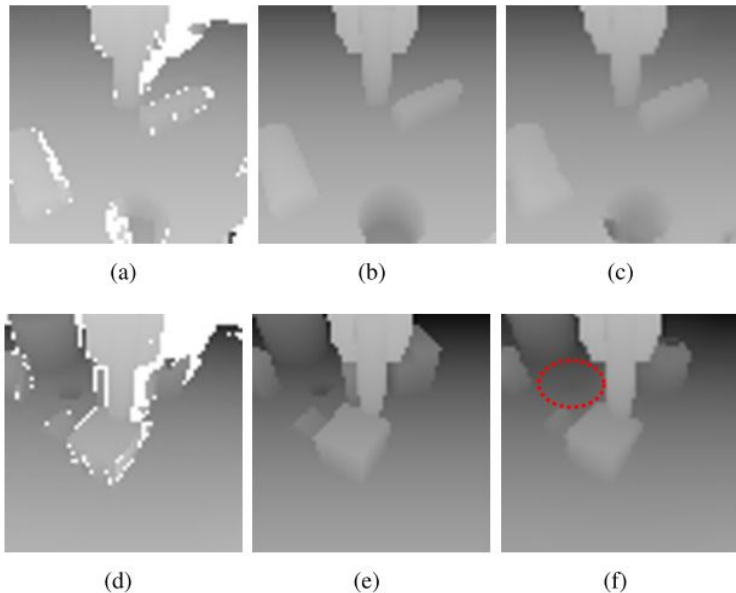
horse → zebra

RL Papers That Use Cycle GAN

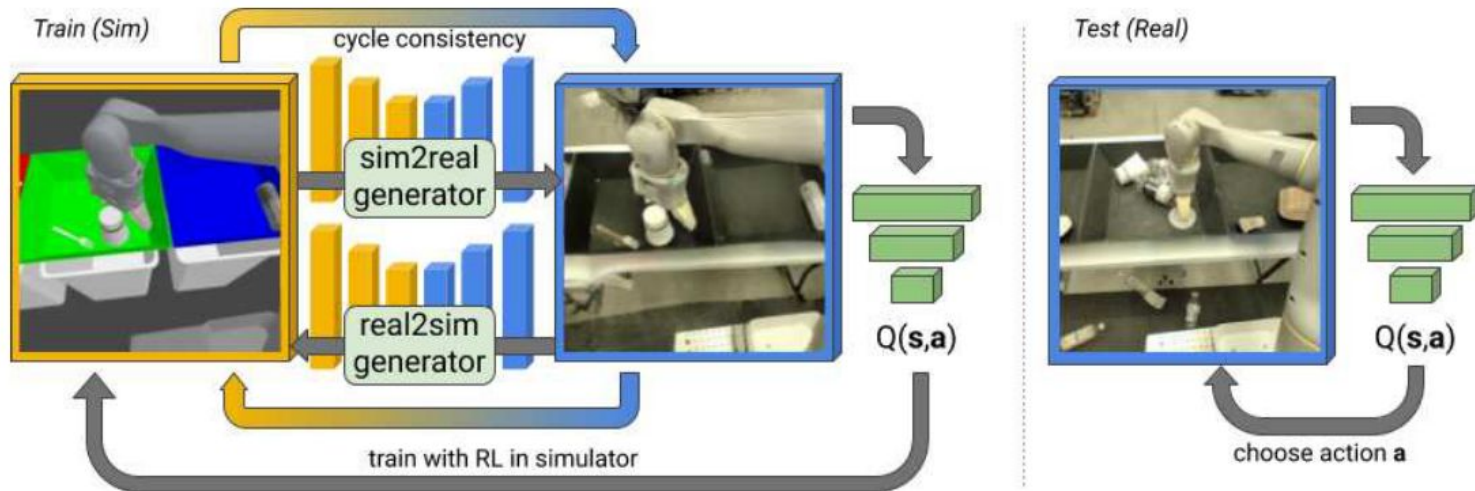
- Learning visual servo policies via planner cloning - Ulrich & Rob
- RL-CycleGAN: Reinforcement Learning Aware Simulation-To-Real - Rao et. al., CVPR 2020
- AVID: Learning Multi-Stage Tasks via Pixel-Level Translation of Human Videos - Laura et.al., RSS 2020

Planner Cloning

- Using pix2pix GAN to convert sim \leftrightarrow real depth images
- Images a+d: real; b+e: simulated; c+f: prediction made by pix2pix
- A hole is missed in figure f
- Real depth images are pre-processed with pix2pix and the result is fed to the policy learned in simulation



RL-Cycle GAN



RL-CycleGAN:
Reinforcement
Learning Aware
Simulation-To-Real -
Rao et. al., CVPR
2020

Figure 1. RL-CycleGAN trains a CycleGAN which maps an image from the simulator (left) to a realistic image (middle), a jointly trained RL task ensures that these images are useful for that specific task. At test time, the RL model may be transferred to real robot (right).

- Naive image-2-image generative model is task-agnostic (Ulrich's case) - the model might filter out useful features for the task
- Jointly trained with RL loss to fix to make it task-aware
- Make sure Q-value for an image is unchanged over the sim2real image translation

RL-Cycle GAN

Sim2Real: $G: X \rightarrow Y$, Real2Sim: $F: Y \rightarrow X$

$$\mathcal{L}_{RL-scene}(G, F) = d(q_x, q'_x) + d(q_x, q''_x) + d(q'_x, q''_x) \\ + d(q_y, q'_y) + d(q_y, q''_y) + d(q'_y, q''_y)$$

$$\mathcal{L}_{RL}(Q) = \mathbb{E}_{(x,a,r,x')} d(Q(x,a), r + \gamma V(x'))$$

$$\mathcal{L}_{RL-CycleGAN}(G, F, D_X, D_Y, Q) \\ = \lambda_{GAN} \mathcal{L}_{GAN}(G, D_Y) \\ + \lambda_{GAN} \mathcal{L}_{GAN}(F, D_X) + \lambda_{cycle} \mathcal{L}_{cyc}(G, F) \\ + \lambda_{RL-scene} \mathcal{L}_{RL-scene}(G, F) + \lambda_{RL} \mathcal{L}_{RL}(Q)$$

$$(x, a) \sim \mathcal{D}_{sim}, (y, a) \sim \mathcal{D}_{real}$$

$$q_x = Q_{sim}(x, a)$$

$$q'_x = Q_{real}(G(x), a)$$

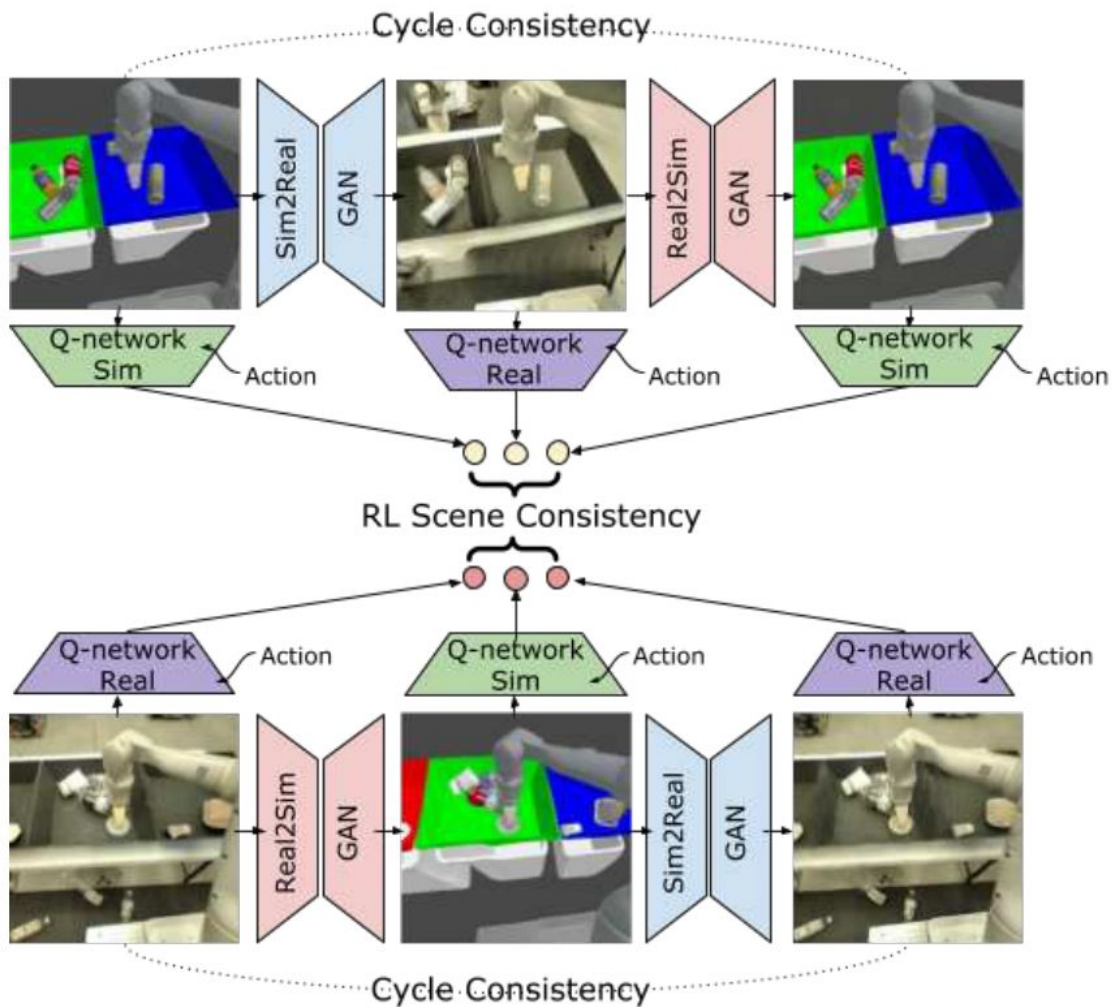
$$q''_x = Q_{sim}(F(G(x)), a)$$

$$q_y = Q_{real}(y, a)$$

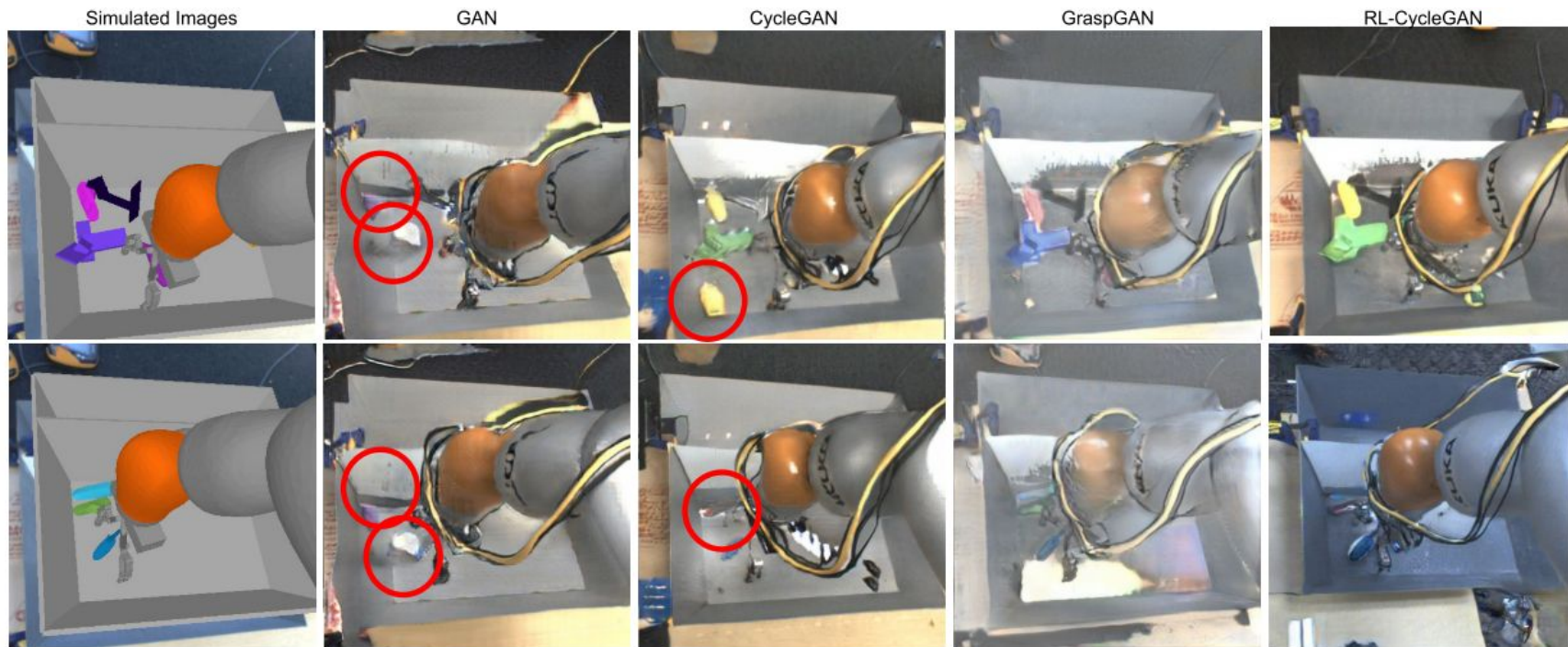
$$q'_y = Q_{sim}(F(y), a)$$

$$q''_y = Q_{real}(G(F(y)), a)$$

RL-Cycle GAN



RL-Cycle GAN



AVID

- Use CycleGAN to convert images of human demonstration to images of a robot performing the task
- Then use a model-based learning approach (MPC + CEM) to learn in a stage-wise fashion with human inputs to mark a stage is successful or not

