Sim-to-Real via Sim-to-Sim: Data-efficient Robotic Grasping via Randomized-to-Canonical Adaptation Networks

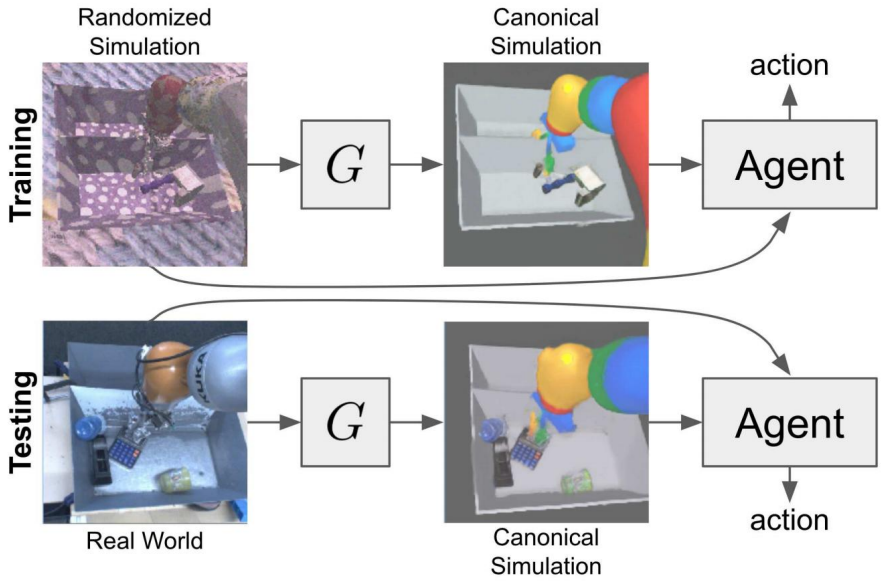
Authors: Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, Konstantinos Bousmalis (2019)

Presenter: Haoyue Cui

Nov 8th, 2022

Motivation and Main Problem

- Problem Solved: They present Randomized_x0002_to-Canonical Adaptation Networks (RCANs), a novel approach to crossing the visual reality gap that uses no real_x0002_world da



Motivation and Main Problem

Why important?

- Real world robotics data is costly. Real-robot data collection expensive and cumbersome.
- large amounts of labelled data can be produced by the power of simulation but the difficulty of transferring simulated experience into the real world comes (reality gap)
    - domain adapatation methods require a large amount of unlabelled real world data(easier than labelled data, but still costly)
    - domain randomization methods:
        - directly used on the input makes the task harder than necessary for modeling the arbitrary changes in the visual domain and decipher the dynamics of the task at the same time.
        - some popular RL algorithms(DDPG and A3C) can be destabilized by this transfer method.

RCAN:

- They learn to adapt from one heavily randomized scene to an **equivalent non-randomized, canonical version**.
- Then they train a robotic grasping algorithm in a **pre-defined canonical version** of our simulator
- They use RCAN model to convert the real_x0002_world images to the canonical domain where their grasping algorithm(QT-Opt, a recent reinforcement learning algorithm) was trained on.
- Advantages:
    - no need for any real-world data
    - gives an interpretable intermediate output
    - solves the stability issue as it is trained in a supervised manner and preprocesses the input

Relate Work - Robotic Grasping

- based on visual and geometric similarity: assumes same/similar objects
- data-driven methods (hand-labeled, grasp positions, self-supervision, predicting grasp outcomes) are important
- state-of-the-art grasping system
    - open-loop (choose grasping locations at fist and execute motion)
    - closed-loop(continuously run grasp prediction during motion)
- **Why vision-based robotic closed-loop grasping?**
    - Robotic grasping is exceptionally challenging since a grasping system must successfully pick up previously unseen objects(cannot just memorize) with internal understanding of geometry and physics.
    - This presents a particularly difficult challenge for simulation-to-real-world transfer.

Relate Work - Randomization

- Random textures, lighting, and camera position, etc
- Apply domain randomization on physical properties of the simulator to aid transferability

    - **Mild randomization** consists of varying tray tex_x0002_ture, object texture and color, robot arm color, lighting di_x0002_rection and brightness, and a background image consisting of 6 different images from the view of the real-world cam_x0002_era.

    - **Medium randomization** adds a diverse mix of back_x0002_ground images to the floor

    - **Heavy randomization** uses the same scheme used to train RCAN

Relate Work - Visual domain adaptation

- training samples from a source domain(simulation) to a target domain(real-world)
- prior methods
    - feature-level adaptation: domain-invariant features are learned between source and target domains
    - pixel_level adaptation: focuses on re-stylizing images from the source domain to make them look like images from the target domain
    - image-to-image translation deals with the easier task of learning such a re-stylization from matching pairs of examples from both domains.
- Their technique can be seen as an image-to-image translation model that transforms randomized renderings from their simulator to their equivalent nonrandomized, canonical ones.
- orthogonal to GraspGAN by Bousmalis et al

Method

- 3 domains: the randomized simulation domain, the canonical simulation domain, and the real-world domain.
- D is a dataset of N samples, where each sample is a tuple containing an RGB image x_s from the randomization(source) domain, an RGB image x_c from the canonical(target) domain(with scene configuration), a segmentation mask m_c , and a depth image d_c .

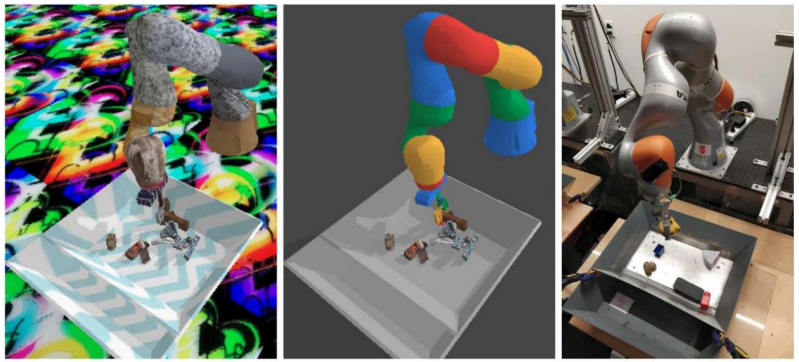
$$\mathbb{D} = \{(x_s, x_c, m_c, d_c)_j\}_{j=1}^{N}$$

- Both the segmentation mask and depth mask are only used as auxiliary tasks during the training of the generator.

RCAN Data Generation

- Simulated environments: Bullet physics engine, default renderer.

a Kuka IIWA robot, a tray, an over-the-shoulder RGB camera aimed at the tray, and a set of graspable objects (a combination of 1,000 procedurally generated objects and 51,300 realistic objects from 55 categories obtained from the **ShapeNet** repository)



(a) Randomized (b) Canonical (c) Real

- Pairs of observations:

scene in canonical version(b)
same scene with randomization applied(a)

- Canonical Environment

uniform colors to the background, tray and arm
leave textures for objects to preserve identity

Future work: opens up the potential for instance-specific grasping

fixed light source

RCAN Training Method

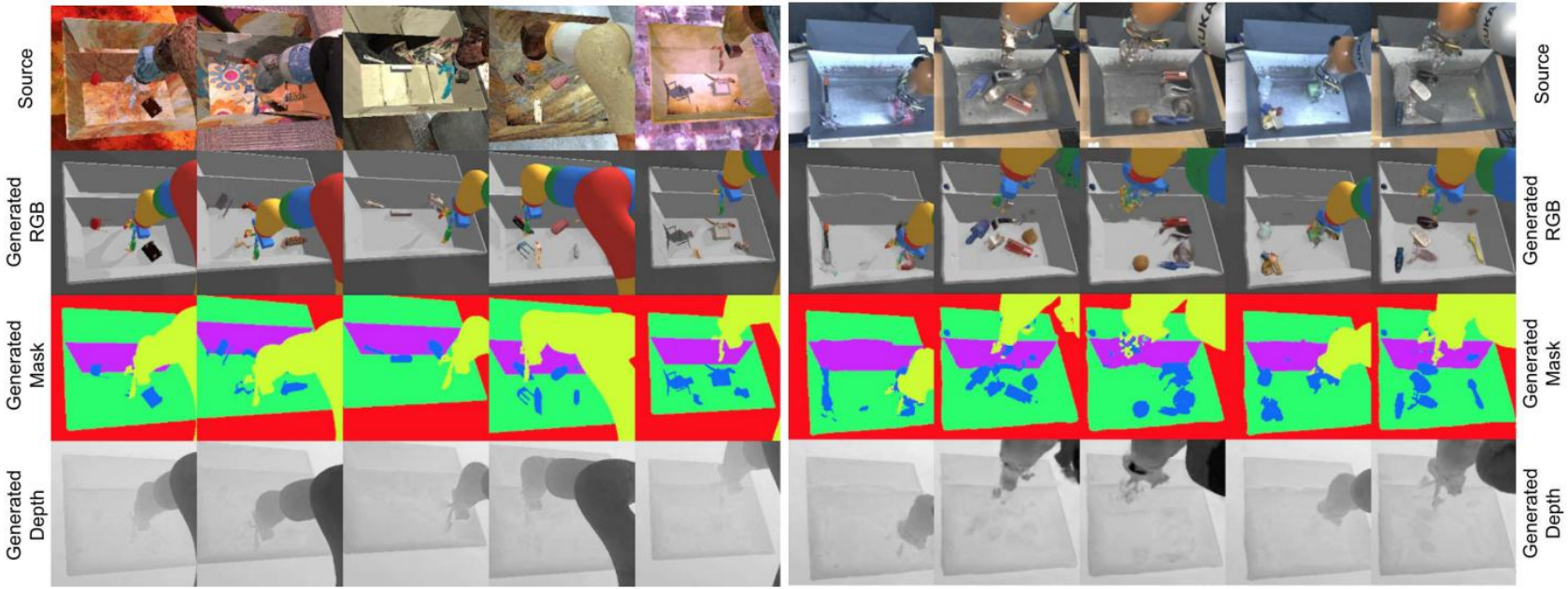
- Consists of image-conditioned generative adversarial network (**cGAN**)
- **The RCAN generator function G(x) → {x_a, m_a, d_a}**, maps an image x from any domain to an adapted image x_a, segmentation mask m_a, and depth image d_a, such that they appear to belong to the canonical domain.
- $G(x_s)$: randomized sim images $G(x_r)$: real world images
- Visual equality between the generated x_a and target x_c through a loss function l_{eqx} (mean pairwise squared error (MPSE))
- Semantic equality between m_c and m_a through a function l_{eqm} (L2)
- Depth equality between d_c and d_a through a function l_{eqd} (L2)

RCAN Training Method

- **L_{eq} loss:** $\mathcal{L}_{eq}(G) = \mathbb{E}_{(x_s,x_c,m_c,d_c)}[\lambda_x l_{eq_x}(G_x(x_s), x_c) + \quad (2)$
$+ \lambda_m l_{eq_m}(G_m(x_s), m_c) + \lambda_d l_{eq_d}(G_d(x_s), d_c)],$
- G_x, G_m, and G_d denote the image, mask, and depth element of the generator output respectively.
- λ_x, λ_m and λ_d represent the respective weightings.
- **L_{GAN} loss:** $\mathcal{L}_{GAN}(G, D) = \mathbb{E}_x[\log D(x)] + \mathbb{E}_x[\log(1 - D(G_x(x))],$
- D(x) is a discriminator that outputs the likelihood that a given image x is from the canonical domain.
- G_x denotes the image element of the generator output
- **Final objective:**

$$\hat{G} = \arg\min_G \max_D \mathcal{L}_{GAN}(G, D) + \mathcal{L}_{eq}(G)$$

RCAN Training Method



(a) Randomized-to-canonical samples.

(b) Real-to-canonical samples.

Q-function Targets via Optimization(QT-Opt)

- QT-Opt is a state-of-the_x0002_art method for vision base grasping, which made it an ideal choice as a **baseline** for a direct comparison.
- QT-Opt is an off-policy, continuous_x0002_action generalization of Q-learning, where the goal is to learn a parametrized Q-function
- Much like other works in RL, stability was improved by the introduction of two target networks.
- Different action selection: QT-Opt instead evaluates the argmax via a stochastic optimization algorithm over a; in this case, the cross-entropy method

QT-Opt training in simulation

- At the **begining** of each episode, ramdomly sampled divider position + 5 randomly selected objects
- At **each** timestep, freeze the scene and apply a new arbitrary randomization to capture the **randomized observation**. Reset to and capture an **canonical version observation** with same transformation to match semantics
- **Observations** consist of RGB images, depth, and segmentation masks
- **Categories**: labeling each pixel with graspable objects, tray, tray divider, robot arm, and background
- **Randomization**: mild randomization, medium randomization, heavy randomization(textures, lighting, arm and tray)

Real World Grasping with QT-Opt

Original QT-Opt vs. RCAN QT-Opt

- Original QT-Opt(Kalashnikov et al.) **st = (xt, gapt,t, gheight,t)**
- RCAN QT-Opt **st = ([G(xt) + xt], gapt,t, gheight,t)**

[G(xt) + xt] represents the concatenation of source image xt and the resulting generated xa with generator G.

- Original QT-Opt trained with 580,000 off-policy real-world grasps, and jointly finetune with an additional 28,000/5,000 on-policy grasps.
- RCAN QT-Opt trained on 28,000/5,000 real on-policy data and generated on-policy simulation data

Experimental Setup

- 102 grasps attempts on 5 to 6 unseen test objects
- Failure case: no object has been grasped after 20 times of grasp attempts.
- Hypotheses
    - Can they train an agent to grasp arbitrary unseen objects without having seen any real-world images?
    - How does QT-Opt perform with standard domain ran_x0002_domization, and can our method perform better than this?
    - Does the addition of real-world on-policy training of our method lead to higher grasping performance while still drastically reducing the amount of real-world data required?

Experimental Results

QT-Opt Data Source	Offline Real Grasps	Performance In Sim	Performance In Real	Online Real Grasps	Performance In Real
Real	580,000	-	87%	+5,000 +28,000	85% 96%
Canonical Sim	0	99%	21%	+5,000	30%
Mild Randomization	0	98%	37%	+5,000	85%
Medium Randomization	0	98%	35%	+5,000	77%
Heavy Randomization	0	98%	33%	+5,000 +28,000	85% 92%
RCAN	**0**	99%	**70%**	+5,000 +28,000	**91%** **94%**

Table 1: Average grasp success rate on test objects after 102 grasp attempts on each of the multiple Kuka IIWA robots. The first 4 columns of the table highlight the performance after training on a specified number of real world grasps. Zero grasps implies that all training was done in simulation. The last 2 columns highlight the results of on-policy joint finetuning on a small amount of real-world grasps.

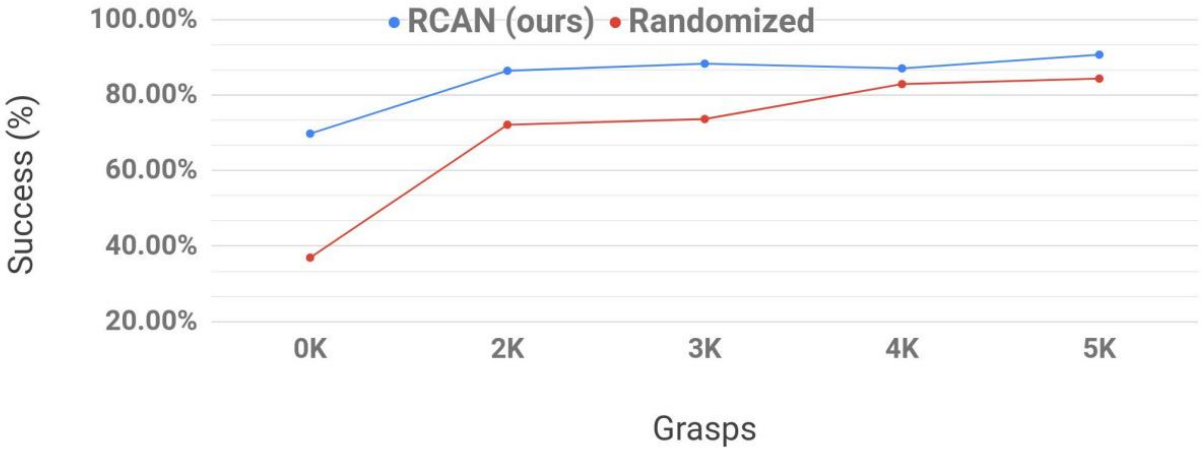
Experimental Results

QT-Opt Data Source	Offline Real Grasps	Performance In Sim	Performance In Real	Online Real Grasps	Performance In Real
Real	580,000	-	87%	+5,000 +28,000	85% 96%
Canonical Sim	0	99%	21%	+5,000	30%
Mild Randomization	0	98%	37%	+5,000	85%
Medium Randomization	0	98%	35%	+5,000	77%
Heavy Randomization	0	98%	33%	+5,000 +28,000	85% 92%
RCAN	**0**	99%	**70%**	+5,000 +28,000	**91%** **94%**

Table 1: Average grasp success rate on test objects after 102 grasp attempts on each of the multiple Kuka IIWA robots. The first 4 columns of the table highlight the performance after training on a specified number of real world grasps. Zero grasps implies that all training was done in simulation. The last 2 columns highlight the results of on-policy joint finetuning on a small amount of real-world grasps.

Experimental Results

- Set up to see how performance varies as they progress from 0 to 5,000 on-policy grasps for both RCAN and Mild Randomization at every 1,000 grasps



Failure cases

- Compared to original QT-Opts which got 96% success with 28,000 online grasps, RCAN lost the **regrasping ability,** the policy to detect when there is no object in the closed gripper, and decide to re-open it in an attempt to try and re-grasp

- **Guess**: the concatenation of source image to generated adapted image affected.
- **Hypothesis**: as the number of joint finetuning grasps increase, the network would eventually learn to solely rely on the source (real-world) image

Future work

- direct domain randomization on other fields
- other use of interpretable output for sim-to-real transfer.
- fusing ideas from other transfer methods that require some real-world data

Summary

- Problem: RCAN, a sim-to-real method that learns to translate randomized simulation images into a canonical representation, which in turn allows for real-world images to also be translated to this canonical representation.
- Why important? Real world data costly
- Key limitation of prior work?

Domain adaptation - still costly; Domain randomization - complicate and destablize

- What did they demonstrate by this insight?

Double the performance than direct domain randomization on start-of-the-art QT-Opt algorithm.

Increase the performance to 91% using 5000 grasps and RCAN than QT-Opt when trained with 580,000 grasps

Any Questions?
Thank you