GraspGAN

Using **Simulation** and **Domain Adaptation** to Improve Efficiency of Deep Robotic Grasping

https://arxiv.org/pdf/1709.07857.pdf

Abstract

- Collecting labeled visual grasping datasets is “time-consuming and expensive”

- Models trained strictly on simulated data don’t generalize well

- Randomized **simulated environments** + **domain adaptation** can be used to train grasping systems to grasp novel objects from **monocular RGB images**

- Reduce the number of real-world samples needed to achieve a given level of performance by up to **50 times**

- **Unlabeled** real-world data + GraspGAN obtains same real-world grasping performance as using **939,777 labeled real-world samples**

Grasping Model

Grasp Prediction CNN

- $C(\mathbf{x}_i, \mathbf{v}_i)$
- $\mathbf{x}_i = \{\mathbf{x}_{i0,}\, \mathbf{x}_{ic}\}$ (*image before robot is visible, image at current time-step*)
- $\mathbf{v}_i$ (*relative change in EE current position + rotation*)
- Only top-down pinch grasps (3 position, 2 rotation)

“Manually designed servoing function”

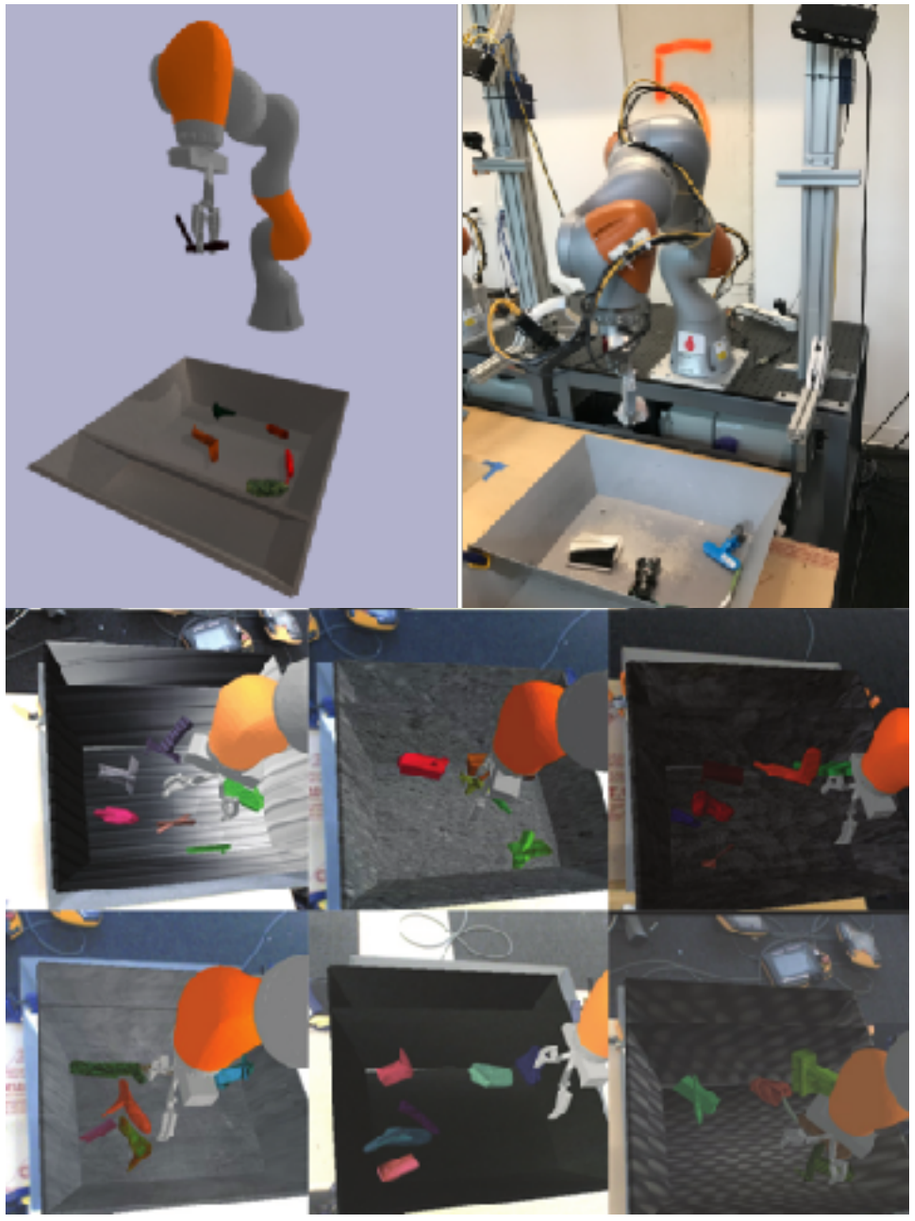
- Uses grasp probabilities from C to choose $\mathbf{v}_i$ that will continuously control the robot
- C trained w/ standard supervised learning independent of the servoing function

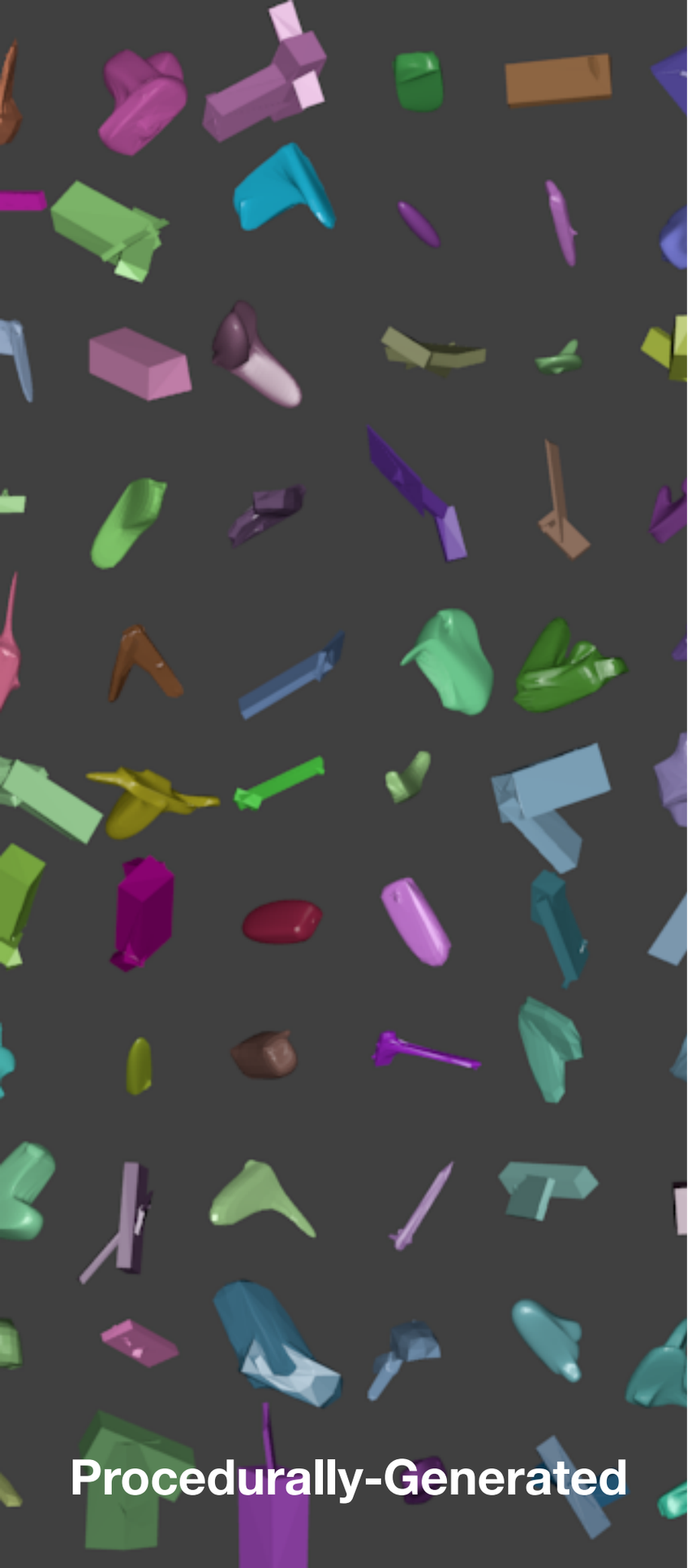
Setup + Data Collection

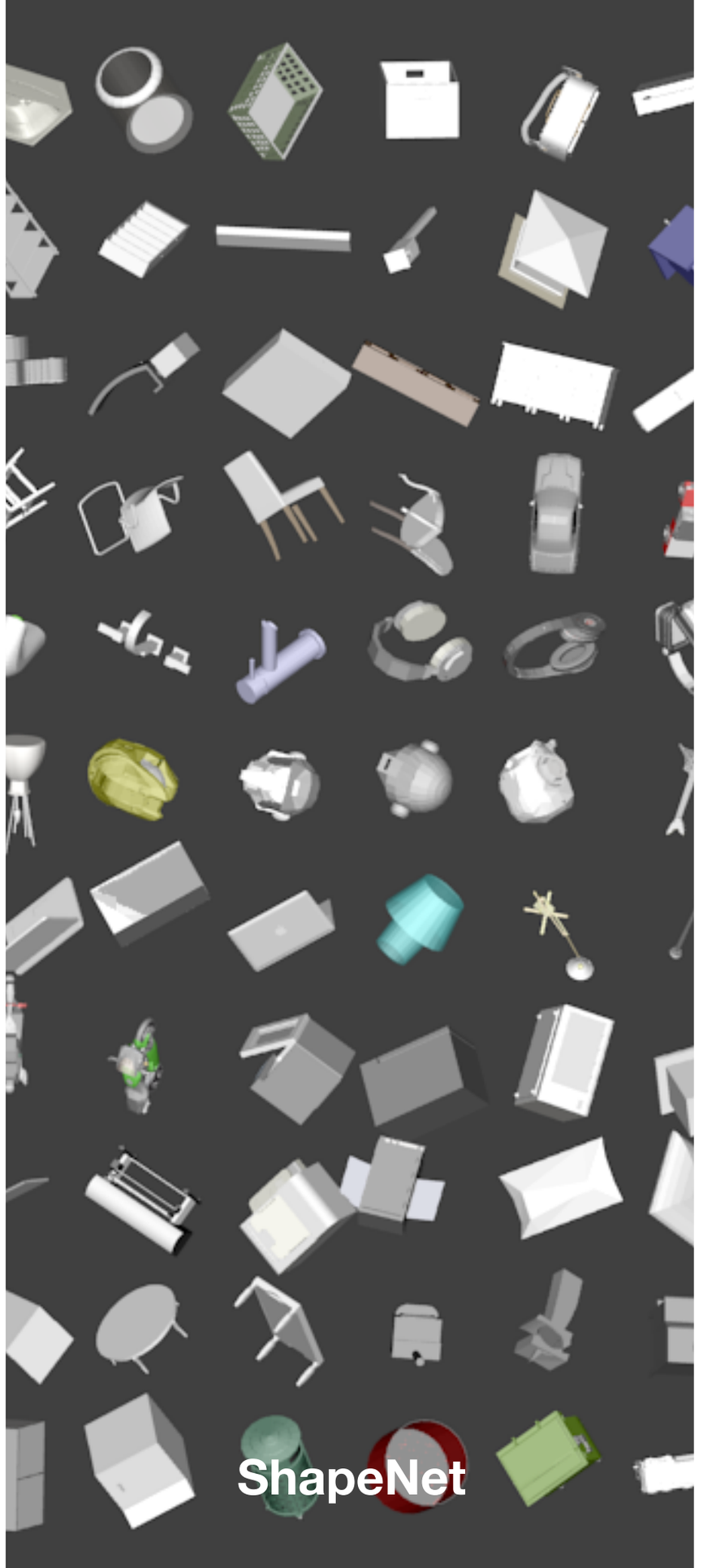
- Grasp Prediction CNN dataset $C(\mathbf{x}_i, \mathbf{v}_i)$ consisted of grasp episodes
  - *T* time-steps with *T* distinct training samples $(\mathbf{x}_i, \mathbf{v}_i)$ and grasp outcome $(\mathbf{y}_i)$
    - 640x512 image inputs randomly cropped to 472x472
- Real-world dataset collected from Levine *et al* [6] w/ 6 Kuka IIWA arms
  - Dataset includes 1 million grasp attempts pf 1100 different objects
  - $C(\mathbf{x}_i, \mathbf{v}_i)$ trained on this dataset achieved successful grasps 67.65% of the time
- 2-finger gripper w/ a monocular RGB camera mounted behind the arm

Simulated Environment

- Bullet physics engine + built-in renderer
- Simulated Kuka hardware, RGB camera mounted behind arm
- a) Procedurally-generated objects
  - Random shapes and colours
  - Used Blender 💖
- b) High-fidelity ShapeNet models
  - 51300 models in 55 categories
  - Resized to [12, 23]cm w/ mass between [10, 500]g
- Collected data w/ 1000-2000 simulated arms and the grasping model was being continually updated
- Simulated robots achieved 70-90% grasp success




Procedurally-Generated


ShapeNet


Real

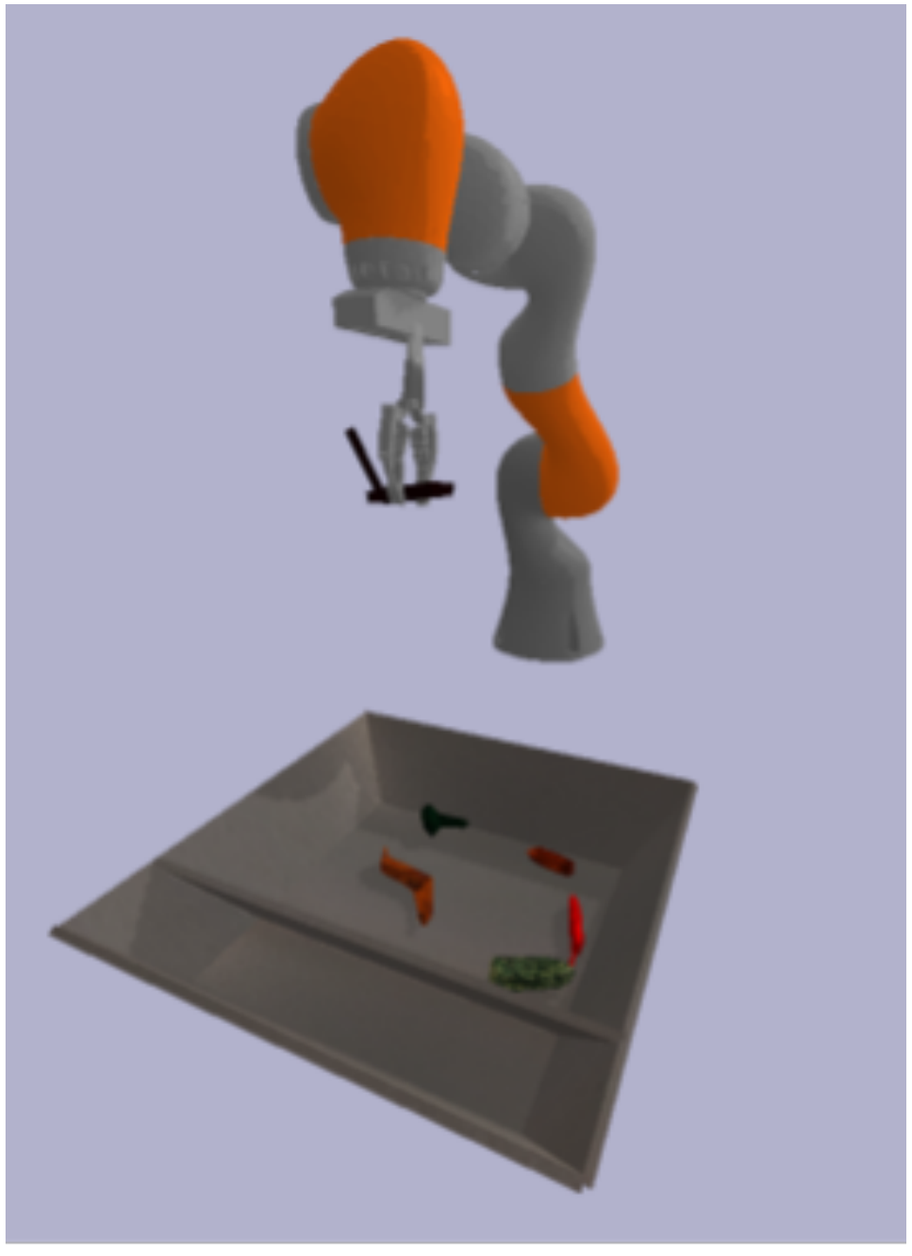
Scene Randomization

A. **No randomization**: Similar to real-world data collection - only varied camera pose, bin location, and used 6 different real-world images as backgrounds

B. **Visual Randomization**: Varied tray texture, object texture and colour, robot arm colour, lighting direction and brightness

C. **Dynamics Randomization**: Varied object mass, and object lateral/rolling/ spinning friction coefficients

D. **All**: visual + dynamics randomization

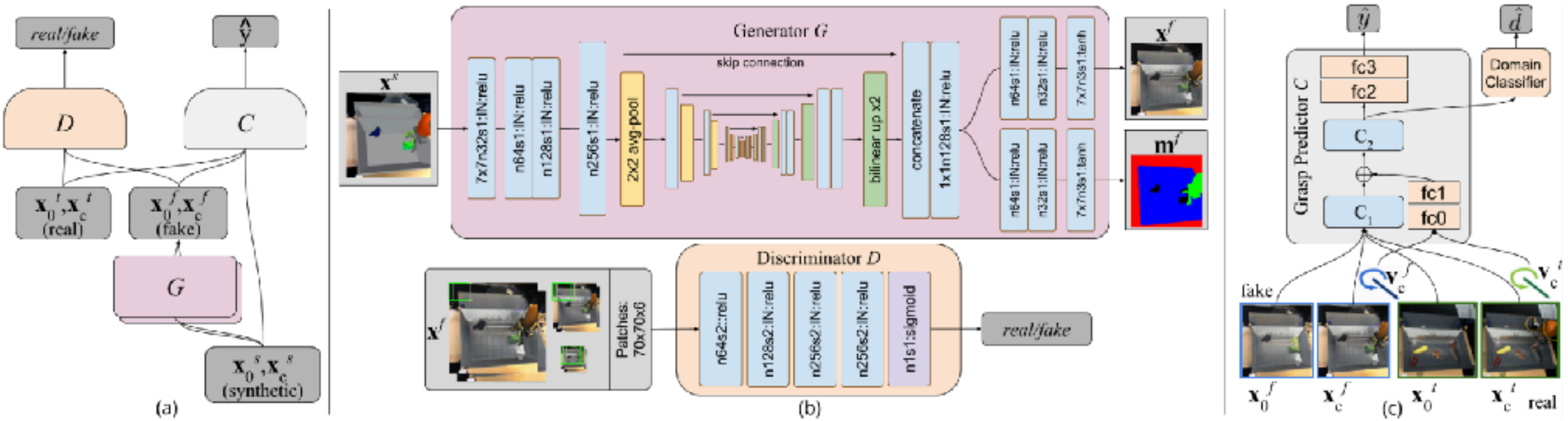


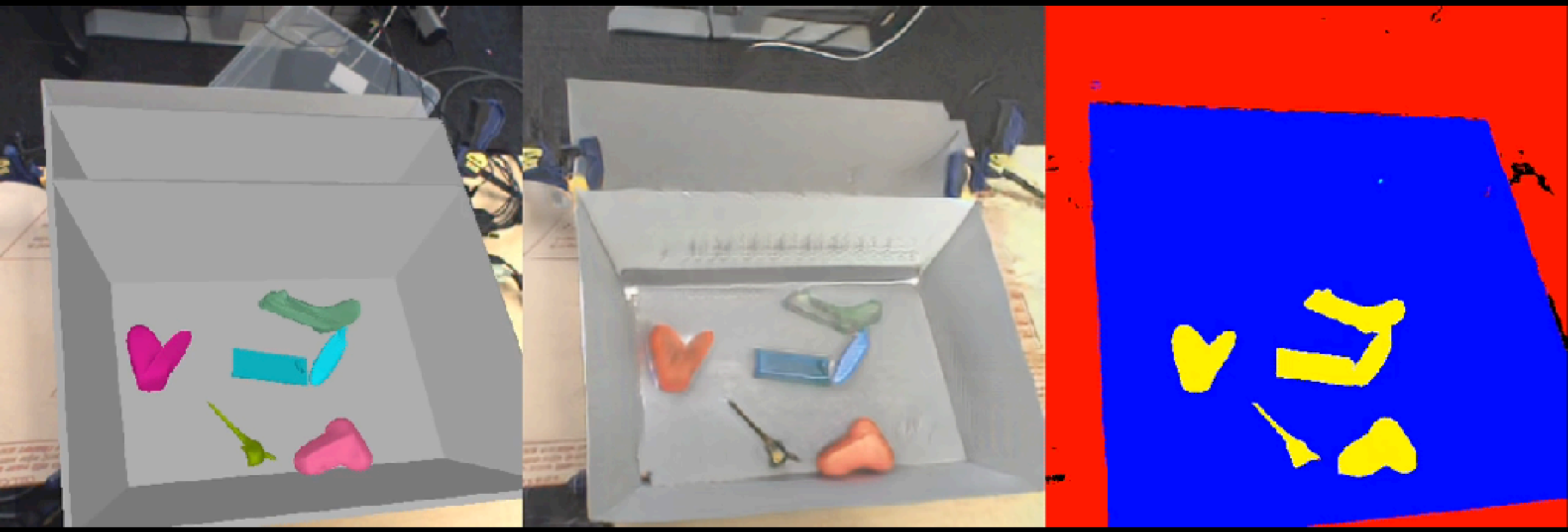
Domain Adaptation

- **Domain Adversarial Training (DANN)**
  - Feature-level domain adaptation used in grasp predictor, C
  - Used batch normalization at every layer of C
    - Extension: domain-specific batch normalization

- **Pixel-level Domain Adaptation (GraspGAN)**
  - G, a CNN, follows U-Net Architecture (2 instances of G), trained as LSGAN
    - Additional loss terms to help anchor generated image to simulated one on a semantic level
    - Generator has auxiliary task to predict semantic maps
  - D, a patch-based CNN, with 6 channel input (3RGB x2 images) to pick-up on relationships between images

GraspGAN Architecture





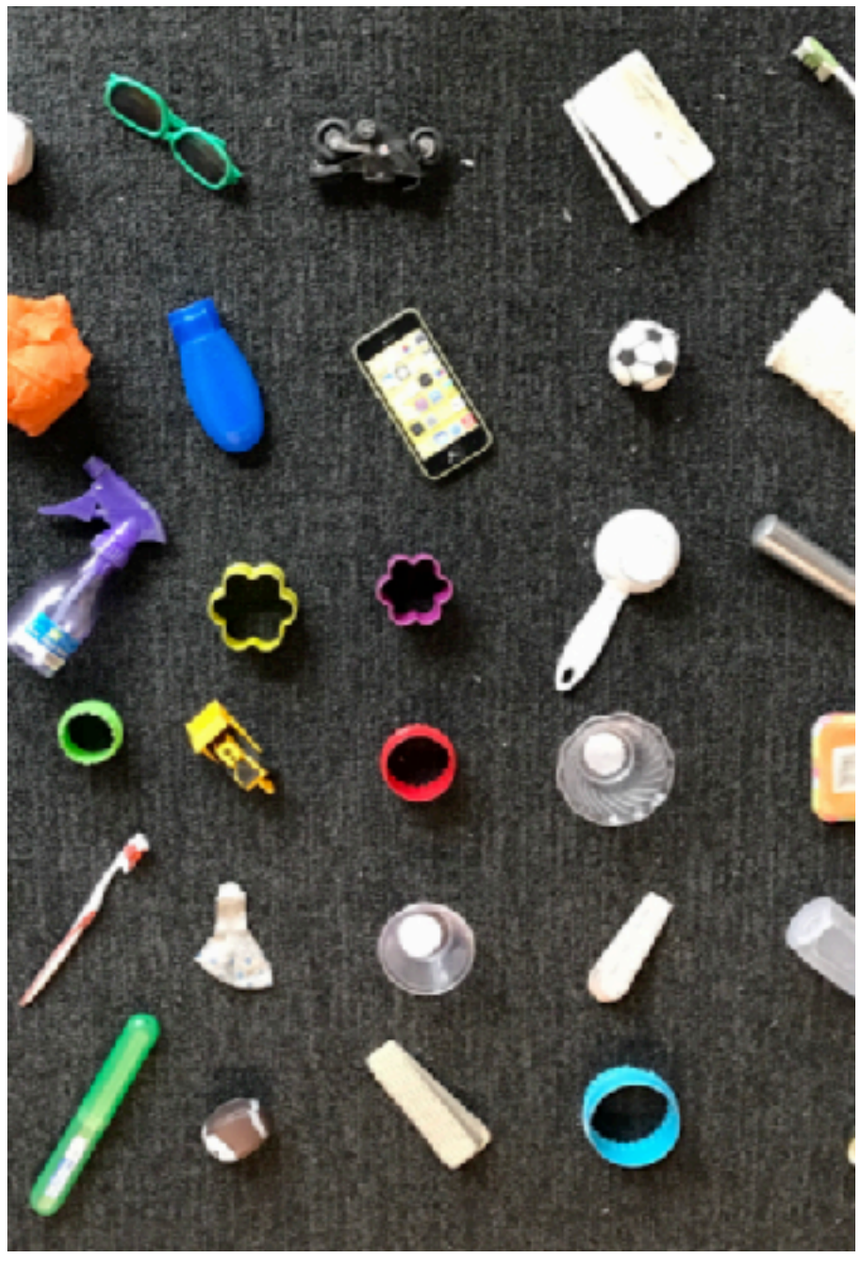
Simulation

Adapted Image

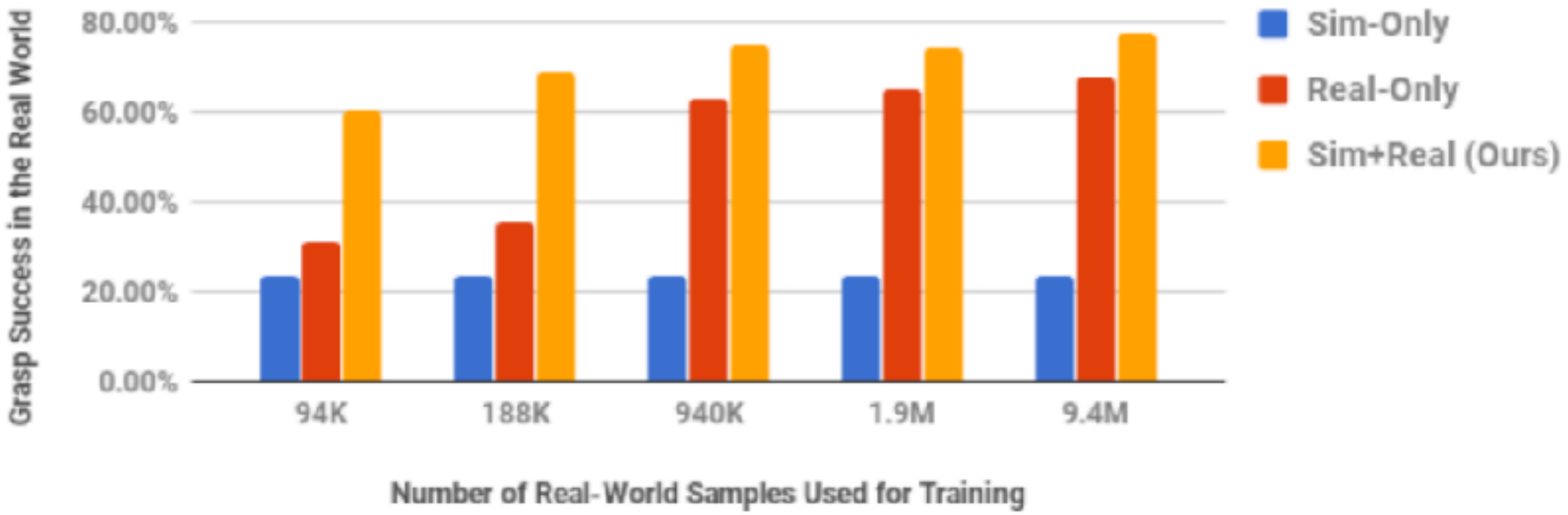
Predicted Semantic Map

Evaluation Setup

- Used training set of ~8 million simulated samples
- 6 Kuka IIWA robots to build a test dataset of,
  - 6 unique unseen objects in each bin of each robot
  - Each robot executed 102 grasps (=612 total)
  - Each robot picks up objects from one side of the bin and drops into the other, alternating every 3 grasps
- Optimal models, C, selected from a held-out 94k validation set



Evaluation of Real and Simulated Dataset



Grasp success gains from incorporating simulated data from procedurally-generated objects.

Evaluation of Model-Fidelity

TABLE I: **The effect of our choices for simulated objects and randomization in terms of grasp success.** We compared the performance of models trained jointly on grasps of procedural vs ShapeNet objects with 10% of the real data. Models were trained with DANN and DBN mixing.

Randomization	None	Visual	Dynamics	Both
Procedural	**71.93%**	**74.88%**	**73.95%**	**72.86%**
ShapeNet	69.61%	68.79%	68.62%	69.84%

Procedurally-generated models led to higher-accuracy vs high-fidelity models in all cases.

Evaluation of Simulated Data Generation and Domain Adaptation

TABLE III: Success of grasping 36 diverse and unseen physical objects of all our methods trained on different amounts of real-world samples and 8 million simulated samples with procedural objects. Method names are explained in the text.

Method	**All 9,402,875**	**20% 1,880,363**	**10% 939,777**	**2% 188,094**	**1% 93,841**
Real-Only	67.65%	64.93%	62.75%	35.46%	31.13%
Naïve Mix.	73.63%	69.61%	65.20%	58.38%	39.86%
Rand.	75.58%	70.16%	73.31%	63.61%	50.99%
DANN	76.26%	68.12%	71.93%	61.93%	59.27%
DANN-R.	72.60%	66.46%	**74.88%**	63.73%	43.81%
GraspGAN	**76.67%**	**74.07%**	70.70%	**68.51%**	**59.95%**

Procedurally-generated models led to higher-accuracy vs high-fidelity models in all cases.

Conclusion

- Including simulated data can greatly improve grasping accuracy
- Using simulated data to supplement real-world data can drastically reduce the necessary real-world dataset size
- Results suggest it’s not important to use high-fidelity 3D models
- Feature and pixel-level domain adaptation substantially improves performance