AuSRoS

Australian School of Robotic Systems

D2 - Bridging the Reality Gap

Prof Tat-Jun (TJ) Chin

make history.

THE UNIVERSITY of ADELAIDE

Lectorial D2 @ AuSRoS
Bridging the Reality Gap

Prof Tat-Jun (TJ) Chin, www.ai4space.group

Australian Institute for Machine Learning

AUSTRALIAN INSTITUTE FOR MACHINE LEARNING
adelaide.edu.au/aiml

SMARTSAT
COOPERATIVE RESEARCH CENTRE

Outline

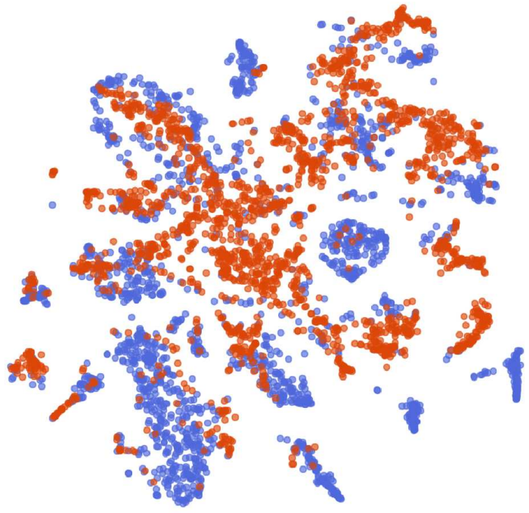
- Domain and sim2real adaptation
- Domain randomisation I
- Domain randomisation II
- Recap on generative adversarial networks (GAN)
- Domain translation
- Modularisation and abstraction
- Latent space alignment
- Adversarial training

Outline

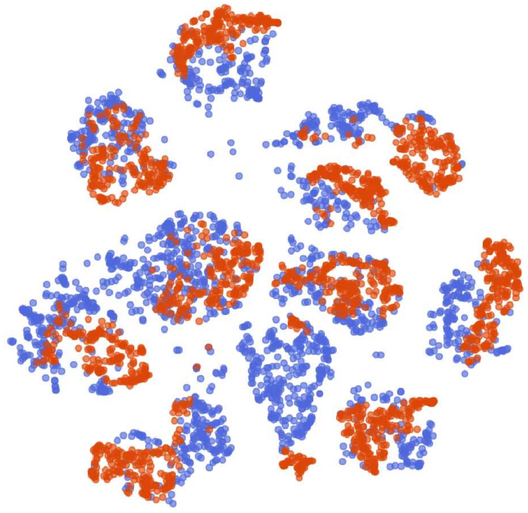
- Domain and sim2real adaptation
- Domain randomisation I
- Domain randomisation II
- Recap on generative adversarial networks (GAN)
- Domain translation
- Modularisation and abstraction
- Latent space alignment
- Adversarial training

Domain adaptation (DA)

- The goal of DA is to train the model f_θ on the source domain data $\mathcal{D} = \{(x_i^{\mathcal{S}}, y_i^{\mathcal{S}})\}_{i=1}^{N}$ and potentially target domain data $\mathcal{E} = \{x_j^{\mathcal{T}}\}_{j=1}^{M}$ in a way that allows f_θ to predict the true labels accurately for unseen before samples $x^{\mathcal{T}} \sim P_X^{\mathcal{T}}$ from the target domain.



without Adaptation



with Adaptation

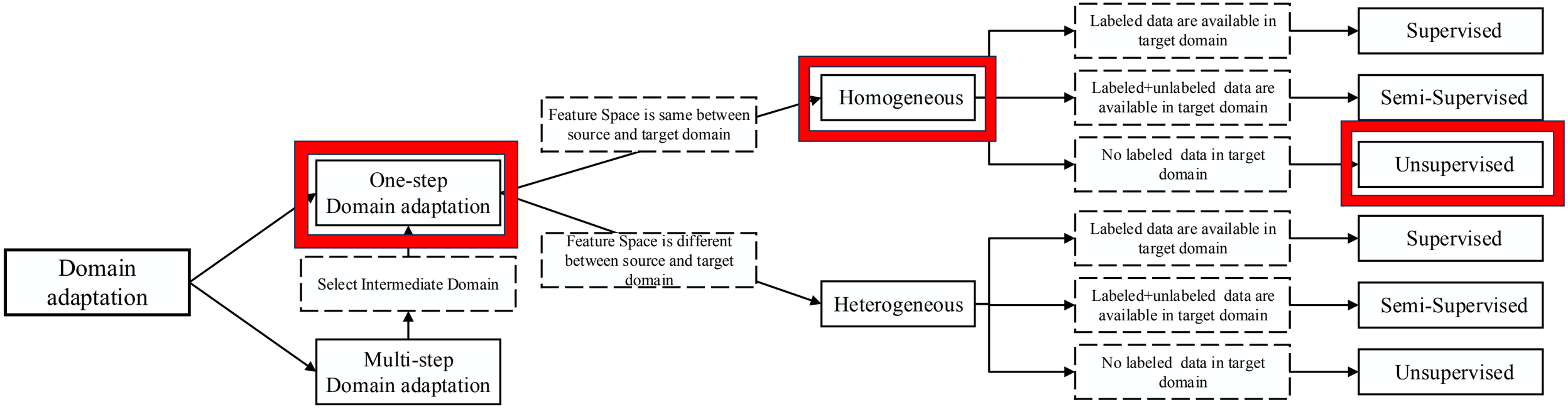
Relationship between DA and other learning frameworks

	K		$P_{XY}^{\mathcal{S}}$ vs $P_{XY}^{\mathcal{T}}$		$\mathcal{Y}_S$ vs $\mathcal{Y}_T$		Access to $P_X^{\mathcal{T}}$?
	$=1$	>1	$=$	$\neq$	$=$	$\neq$	
Supervised Learning	✓		✓		✓		
Multi-Task Learning		✓	✓		✓		
Transfer Learning	✓	✓		✓		✓	✓
Zero-Shot Learning	✓			✓		✓	
Domain Adaptation	✓	✓		✓	✓	✓	✓
Test-Time Training	✓	✓		✓	✓		✓†
Domain Generalization	✓	✓		✓	✓	✓	

K: number of source domains/tasks. $P_{XY}^{S/T}$: source/target joint distribution. $\mathcal{Y}_{S/T}$: source/target label space. $P_X^{\mathcal{T}}$: target marginal. †: Limited in quantities, like a single example or mini-batch.

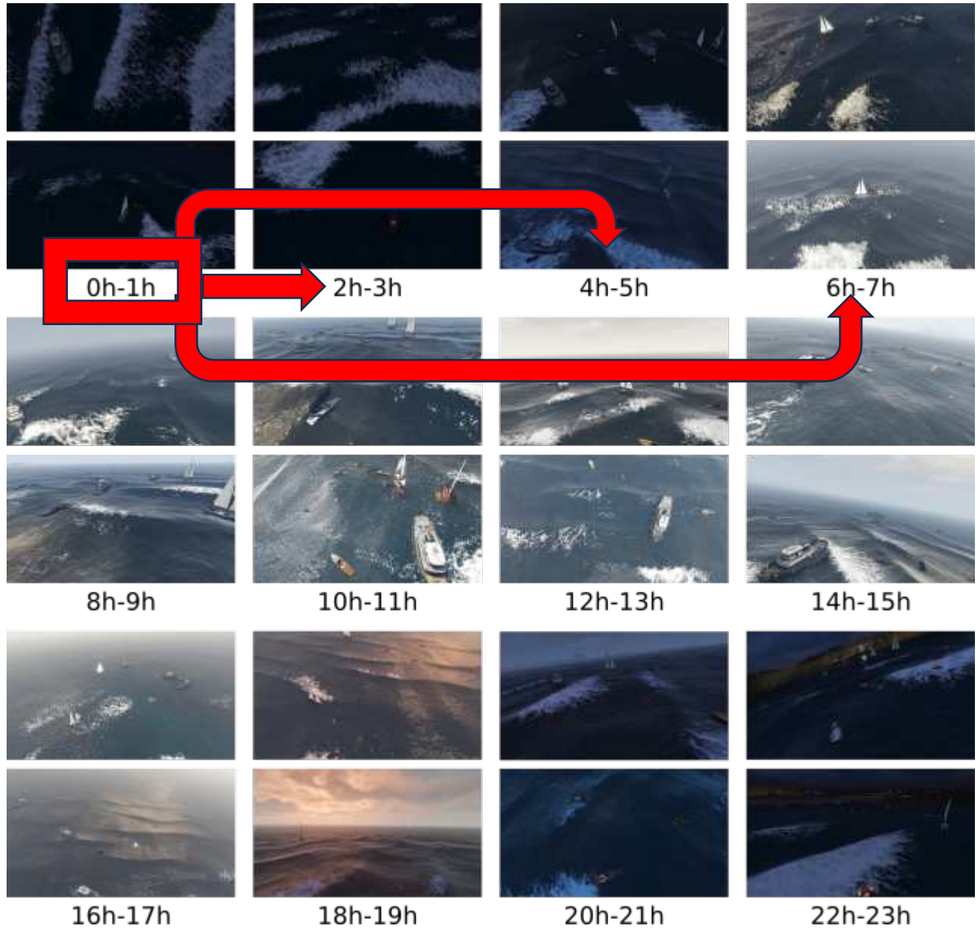
Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain Generalization: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 45, Issue: 4, 01 April 2023)

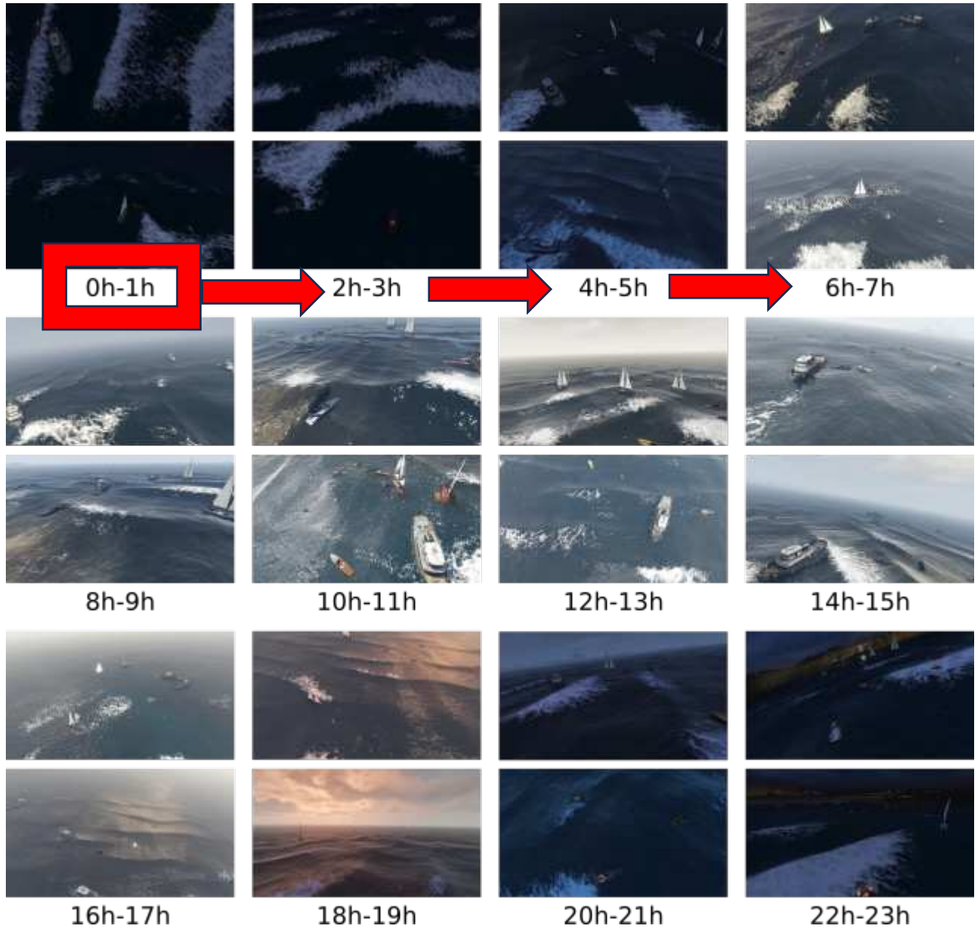
Taxonomy of DA settings



Mei Wang, Weihong Deng. Deep Visual Domain Adaptation: A Survey. Neurocomputing Volume 312, 27 October 2018.

One-step DA vs multi-step DA



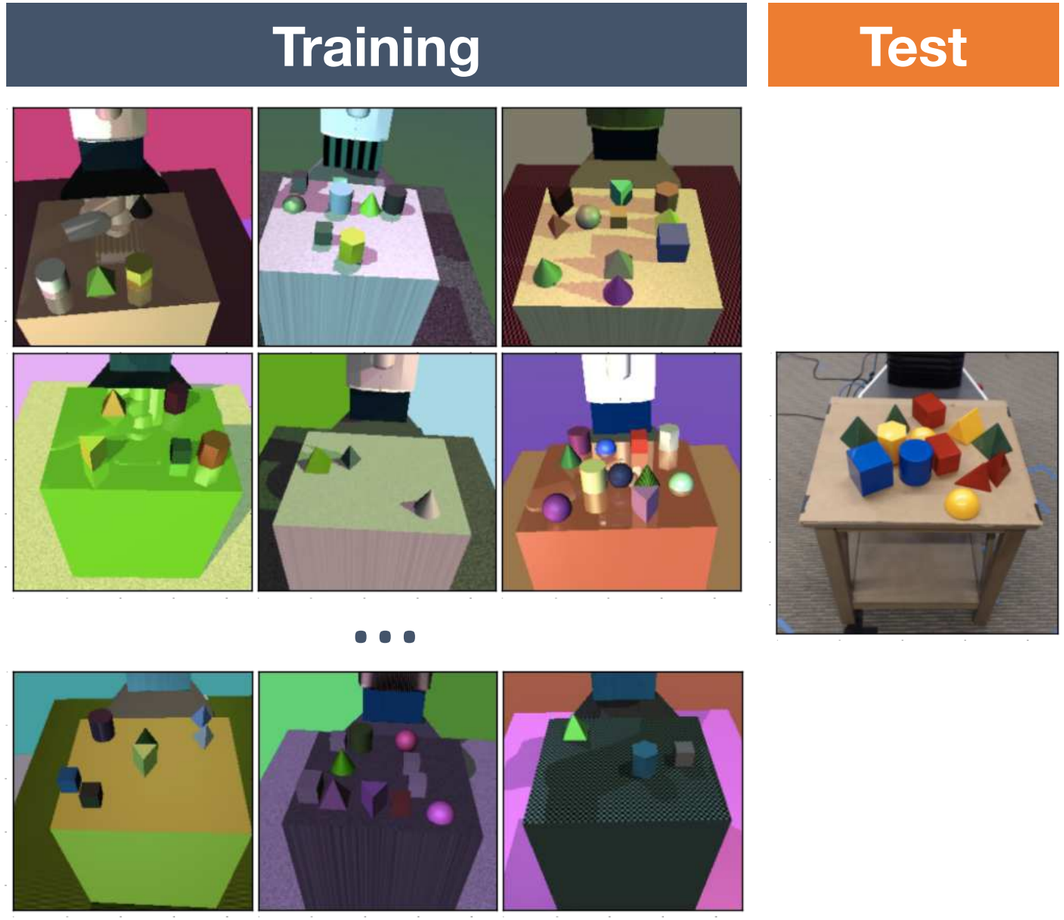


Outline

- Domain and sim2real adaptation
- Domain randomisation I
- Domain randomisation II
- Recap on generative adversarial networks (GAN)
- Domain translation
- Modularisation and abstraction
- Latent space alignment
- Adversarial training

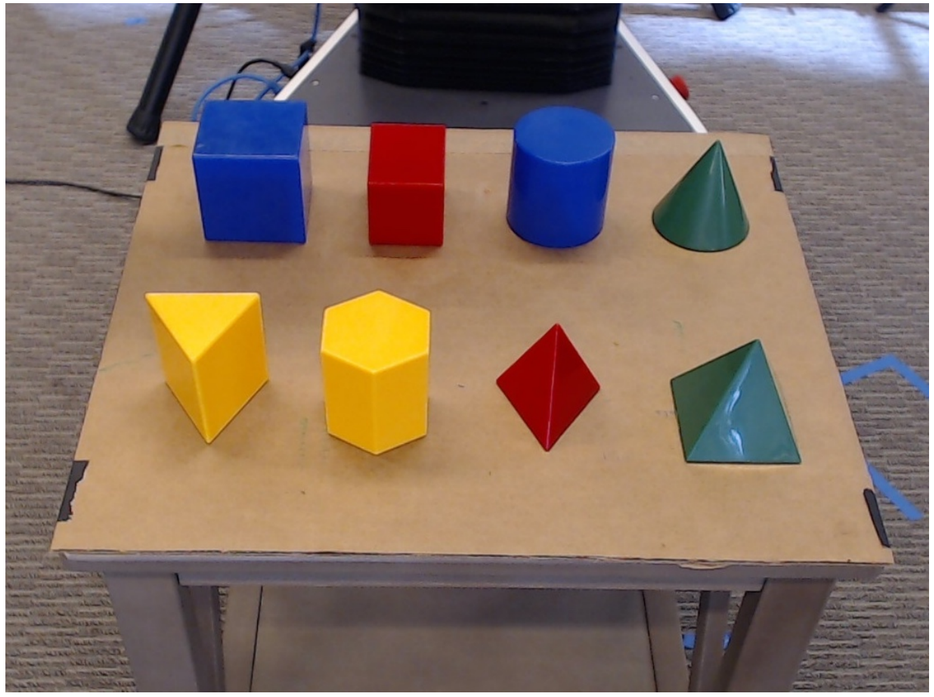
Domain randomisation I

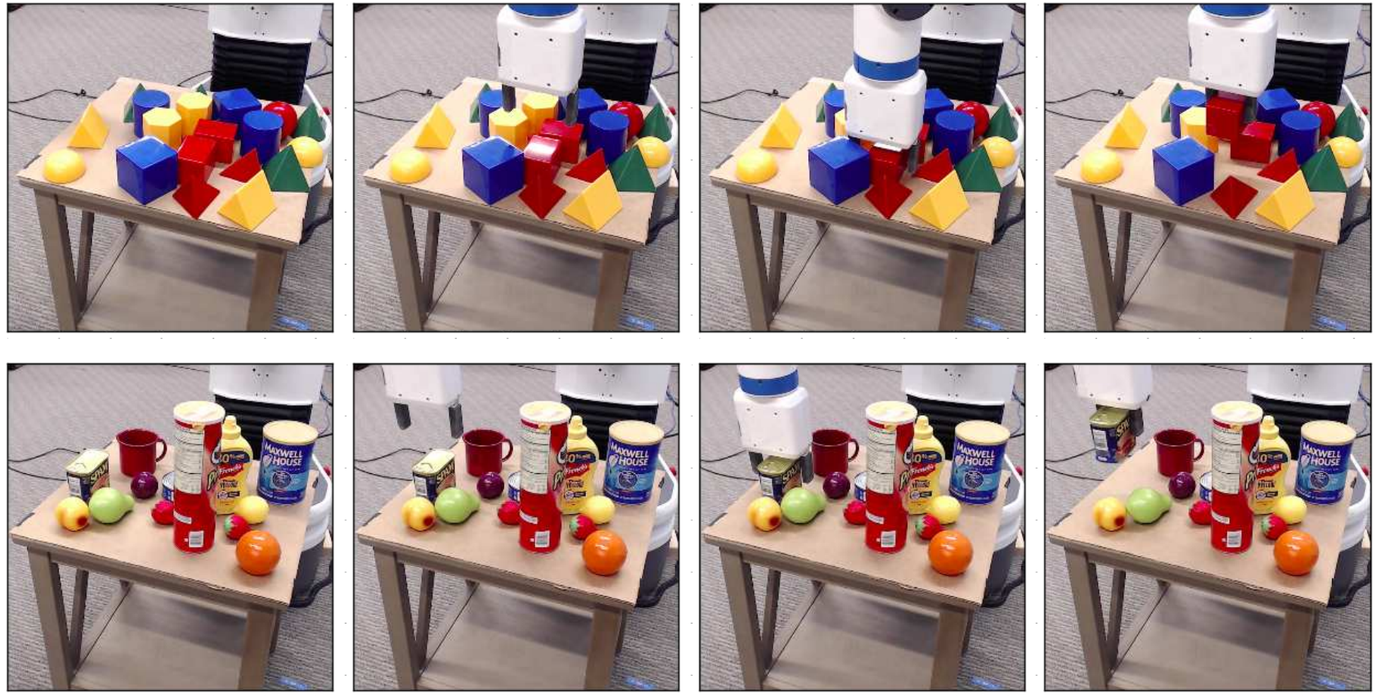
- Problem setting:
  - Sim2real DA to train an object 3D localiser for robotic object pickup.
- Key ideas:
  - Randomise the simulator extensively to expose the model to a wide range of conditions during training.
  - If the variability of the training conditions is extensive enough, the trained model can generalise to the real environment without needing additional finetuning.



. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," IROS 2017.

Problem setting





The object detector network estimates the 3D positions of the objects of interest relative to the camera, and then a motion planner plans a simple sequence of motions to grasp the object at that location.

. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," IROS 2017.

Model

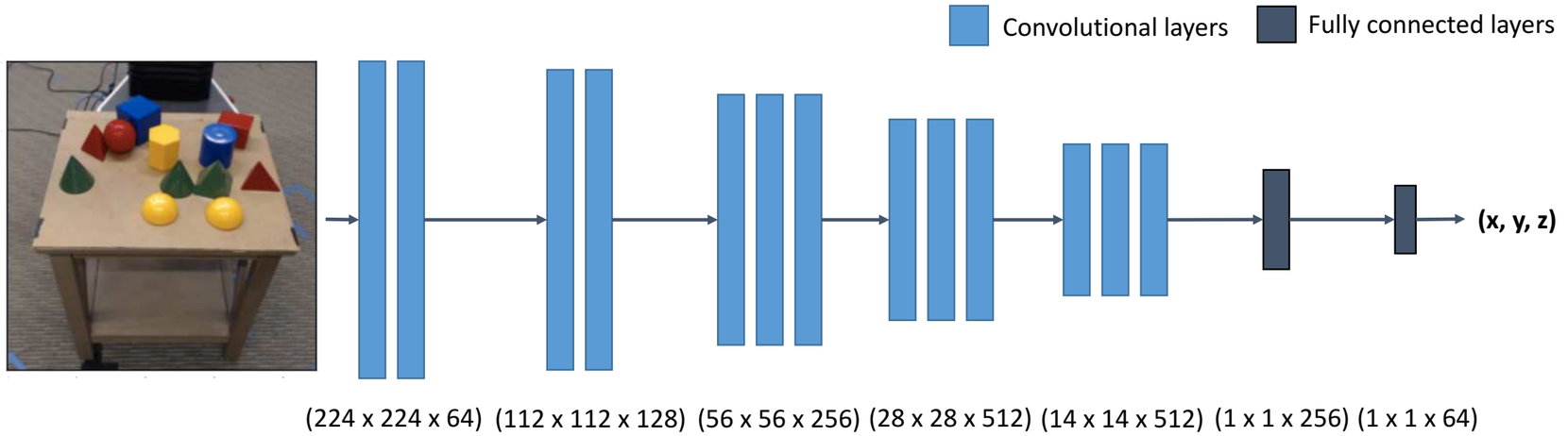


Fig. 2. The model architecture used in our experiments. Each vertical bar corresponds to a layer of the model. ReLU nonlinearities are used throughout, and max pooling occurs between each of the groupings of convolutional layers. The input is an image from an external webcam downsized to (224×224) and the output of the network predicts the (x, y, z) coordinates of object(s) of interest.

. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," IROS 2017.

Method

- Images were rendered using built-in renderer of MuJoCo Physics Engine. No focus on photorealism.
- Aspects randomised:
  - Number and shape of distractor objects on the table
  - Position and texture of all objects on the table
  - Textures of the table, floor, skybox, and robot
  - Position, orientation, and field of view of the camera
  - Number of lights in the scene
  - Position, orientation, and specular characteristics of the lights
  - Type and amount of random noise added to images
- Random textures are chosen among the following:
  - A random RGB value
  - A gradient between two random RGB values
  - A checker pattern between two random RGB values

. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," IROS 2017.

Sample randomised training data

. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," IROS 2017.

Results

Accuracy on real data

Detection error for various objects, cm			
Evaluation type	Object only	Distractors	Occlusions
Cone	1.3 ± 1.1^1	1.5 ± 1.0	1.4 ± 0.6
Cube	1.3 ± 0.6	1.8 ± 1.2	1.4 ± 0.6^1
Cylinder	1.1 ± 0.9^1	1.9 ± 2.8	1.9 ± 2.9
Hexagonal Prism	0.7 ± 0.5	0.6 ± 0.3^1	1.0 ± 1.0^1
Pyramid	0.9 ± 0.3^1	1.0 ± 0.5^1	1.1 ± 0.7^1
Rectangular Prism	1.3 ± 0.7	1.2 ± 0.4^1	0.9 ± 0.6
Tetrahedron	0.8 ± 0.4^1	1.0 ± 0.4^1	3.2 ± 5.8
Triangular Prism	0.9 ± 0.4^1	0.9 ± 0.4^1	1.9 ± 2.2

Ablation studies

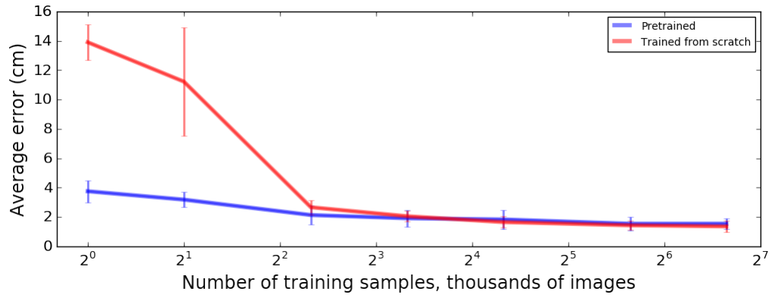


Fig. 4. Sensitivity of test error on real images to the number of simulatec training examples used. Each training example corresponds to a single labeled example of an object on the table with between 0 and 10 distractor objects. Lighting and all textures are randomized between iterations.

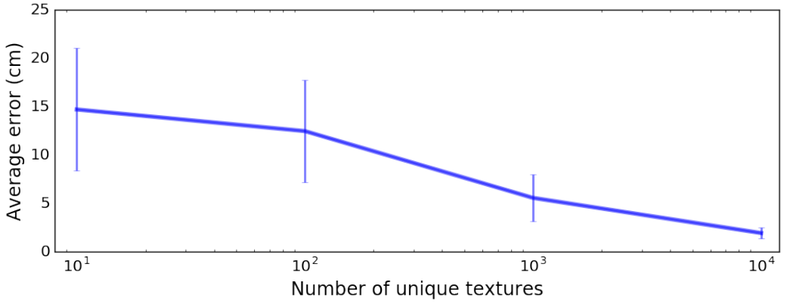


Fig. 5. Sensitivity to amount of texture randomization. In each case, the detector was trained using $10,000$ random object positions and combinations of distractors, but only the given number of unique texturizations and lighting conditions were used.

Average detection error on geometric shapes by method, cm[4]			
Evaluation type	*Real images*		
	Object only	Distractors	Occlusions
Full method	$\mathbf{1.3 \pm 0.6}$	$\mathbf{1.8 \pm 1.7}$	$\mathbf{2.4 \pm 3.0}$
No noise added	1.4 ± 0.7	1.9 ± 2.0	$\mathbf{2.4 \pm 2.8}$
No camera randomization	2.0 ± 2.1	2.4 ± 2.3	2.9 ± 3.5
No distractors in training	1.5 ± 0.6	7.2 ± 4.5	7.4 ± 5.3

. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," IROS 2017.

What acts as the bridge between sim and real?

The dataset is the bridge between sim and real.

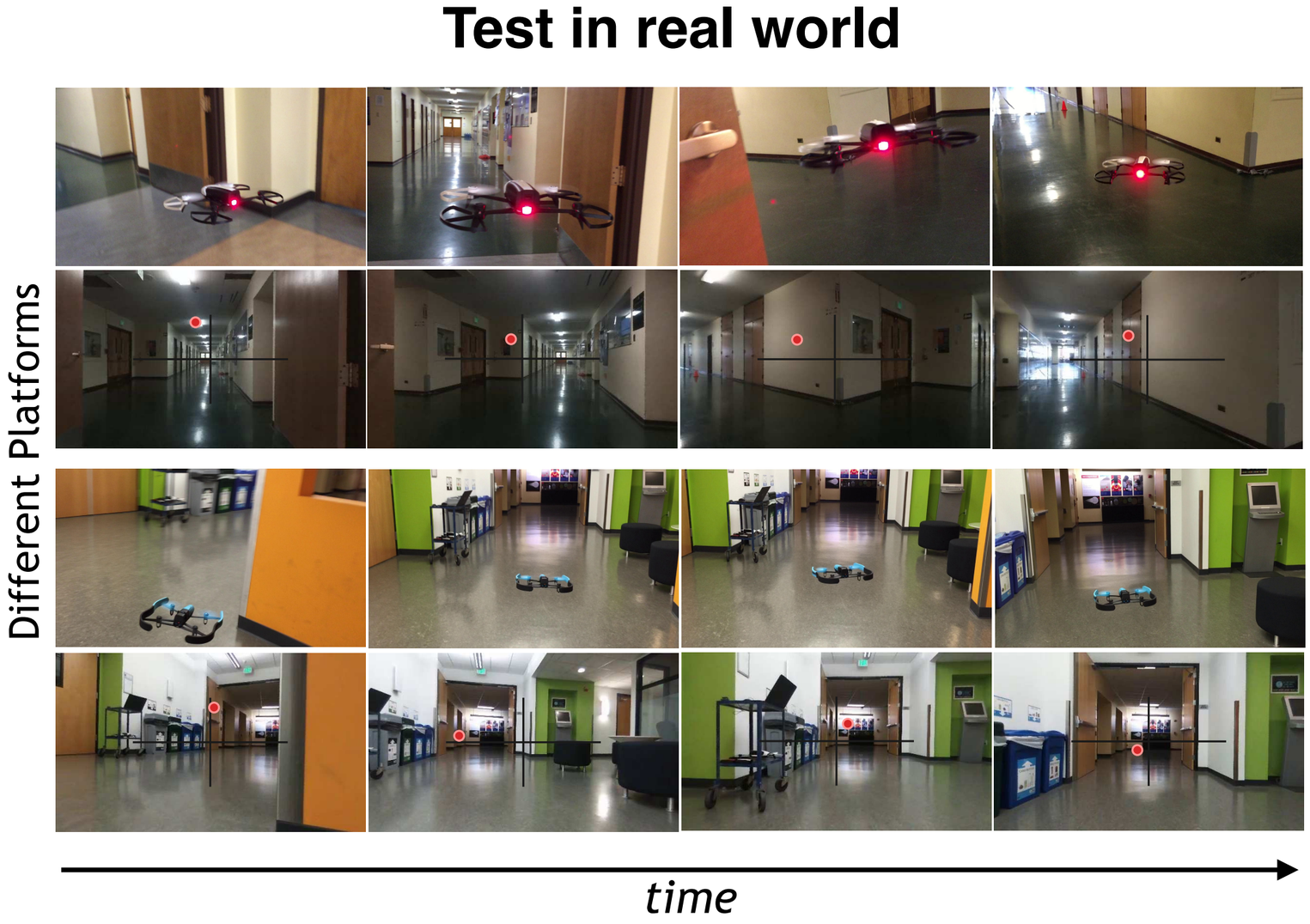
. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," IROS 2017.

Outline

- Domain and sim2real adaptation
- Domain randomisation I
- Domain randomisation II
- Recap on generative adversarial networks (GAN)
- Domain translation
- Modularisation and abstraction
- Latent space alignment
- Adversarial training

Domain randomisation II

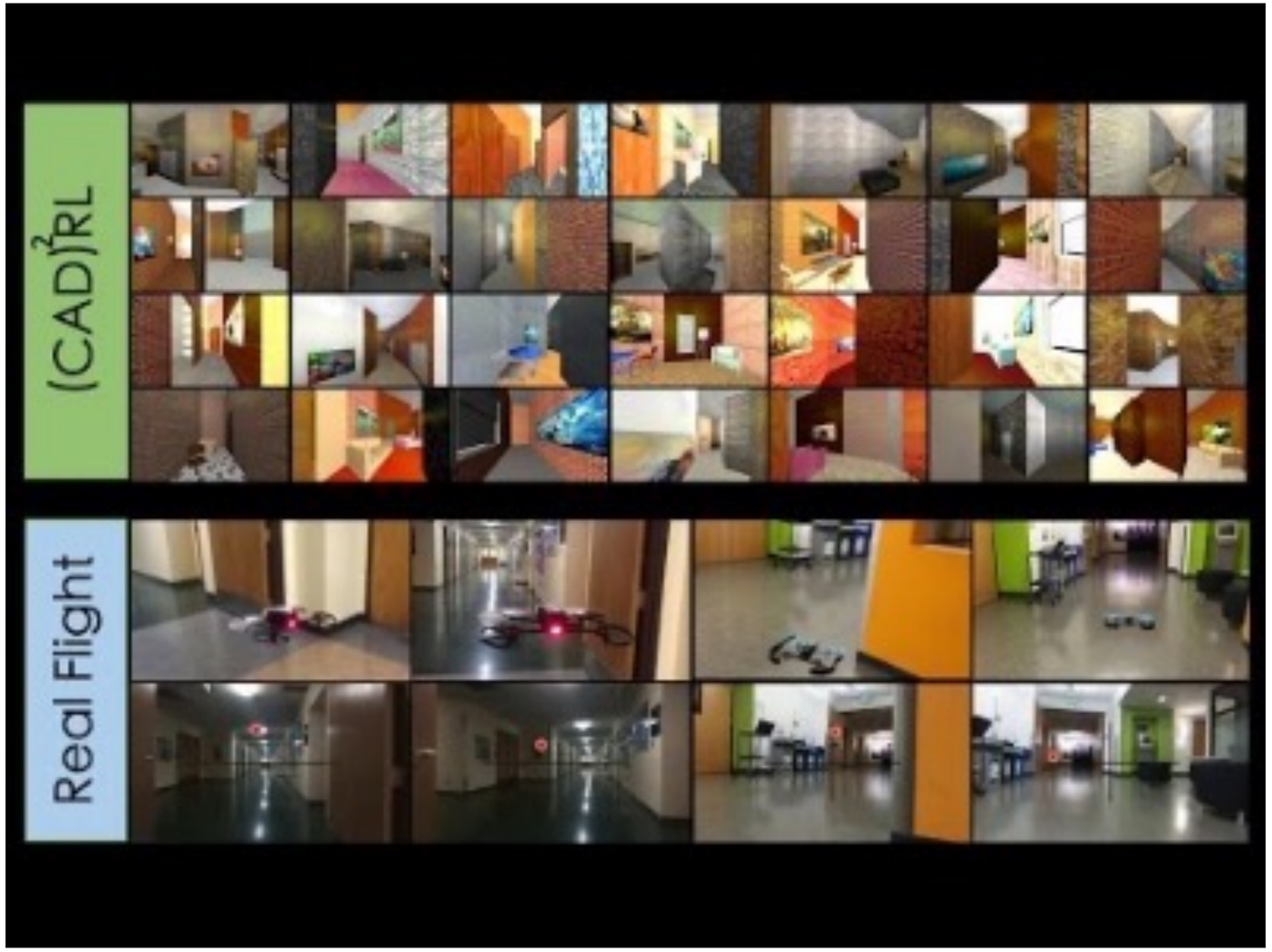
- Problem setting:
  - Given an image of the scene, decide the direction of travel that avoids collision in an indoor environment.
  - Learn a policy that outputs the action (direction of travel) with the best expected reward.
- Idea:
  - Learn the optimal policy from simulation data with ground truth depth.
  - Randomise the simulation data to enable sim2real adaptation.



Fereshteh Sadeghi and Sergey Levine. CAD2RL: Real Single-Image Flight without a Single Real Image. RSS 2017.

Domain randomisation II

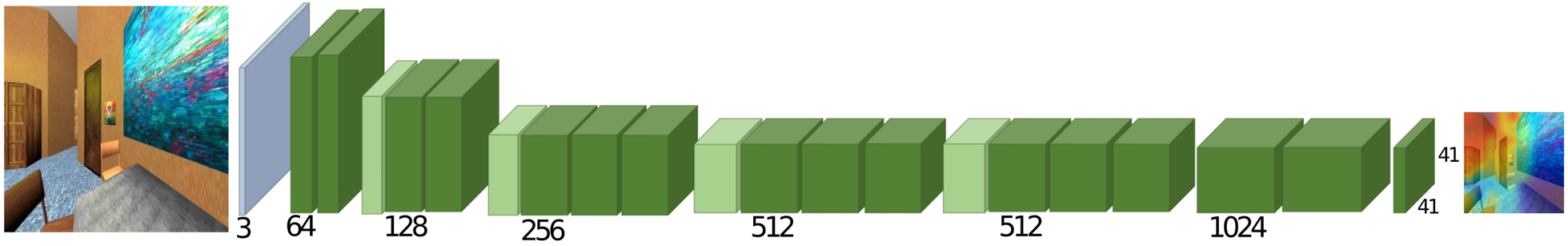
- Problem setting:
  - Given an image of the scene, decide the direction of travel that avoids collision in an indoor environment.
  - Learn a policy that outputs the action (direction of travel) with the best expected reward.
- Idea:
  - Learn the optimal policy from simulation data with ground truth depth.
  - Randomise the simulation data to enable sim2real adaptation.



Fereshteh Sadeghi and Sergey Levine. CAD2RL: Real Single-Image Flight without a Single Real Image. RSS 2017.

Method

- Use a CNN to learn the Q-function: $Q(\mathbf{I}_t, \mathbf{a}_t) = \sum_{s=t, \mathbf{a} \sim \pi}^{t+H} \gamma^{s-t} \mathcal{R}(\mathbf{I}_s, \mathbf{a}_s)$



- The Q-function gives the expected reward from executing action $\mathbf{a}_t$ at current $\mathbf{I}_t$, where
  - An action is the direction of travel obtained by backprojecting a pixel location.
  - H is a time horizon.
  - Reward is 0 if collision happens, and >0 otherwise.
- Based on the learned Q-function, adopt the greedy policy

$$\pi(\mathbf{I}) = \mathbf{a}^* = \arg \max_{\mathbf{a} \in \text{set of image pixels}} Q(\mathbf{I}, \mathbf{a})$$

- Note the recursive definition of Q-function based on the optimal policy.

Fereshteh Sadeghi and Sergey Levine. CAD2RL: Real Single-Image Flight without a Single Real Image. RSS 2017.

Training data

- Train the CNN using simulation environment (with ground truth depth) and images:

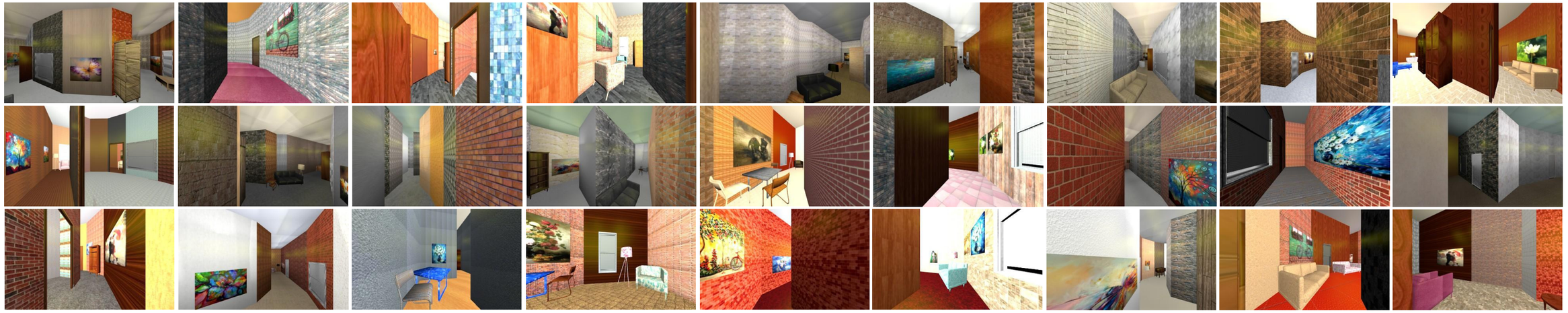


Fig. 2. Examples of rendered images using our simulator. We randomize textures, lighting and furniture placement to create a visually diverse set of scenes.

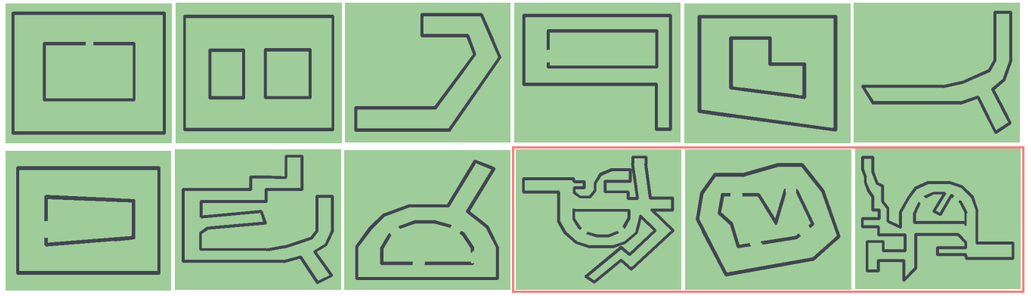
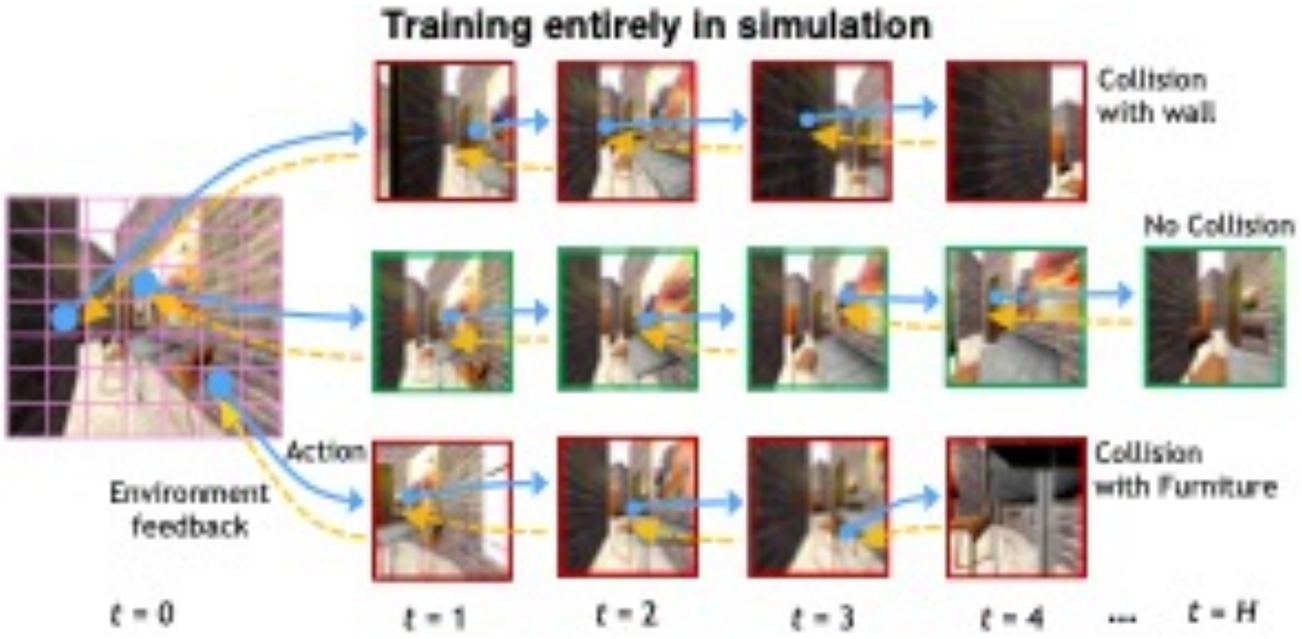


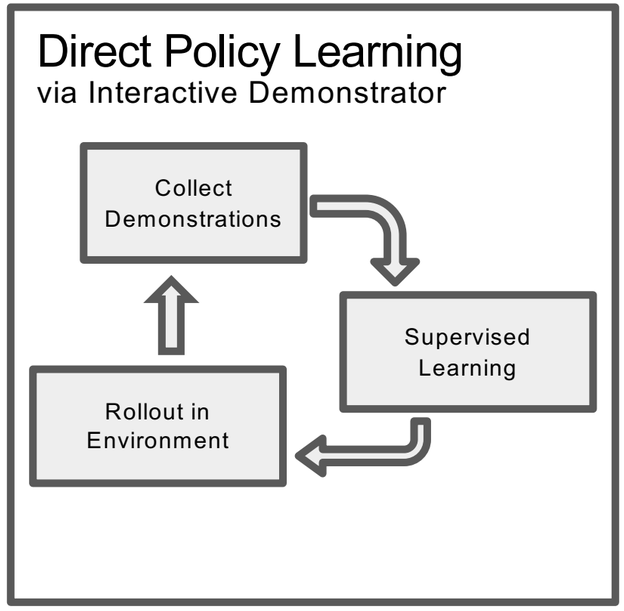
Fig. 4. Floor plans of the synthetic hallways. The last three hallways are used for evaluation while the first 9 are used during training.

Fereshteh Sadeghi and Sergey Levine. CAD2RL: Real Single-Image Flight without a Single Real Image. RSS 2017.

Training procedure

- Initialise random starting points in the environment rollout for each pixel location:





- The rollouts provide labelled training data $(\mathbf{I}_t, \mathbf{a}_t, Q(\mathbf{I}_t, \mathbf{a}_t))$ to train the CNN. Alternate between retraining and simulating (data collection).

Fereshteh Sadeghi and Sergey Levine. CAD2RL: Real Single-Image Flight without a Single Real Image. RSS 2017.

Results

- Synthetic indoor environment with “realistic textures”:

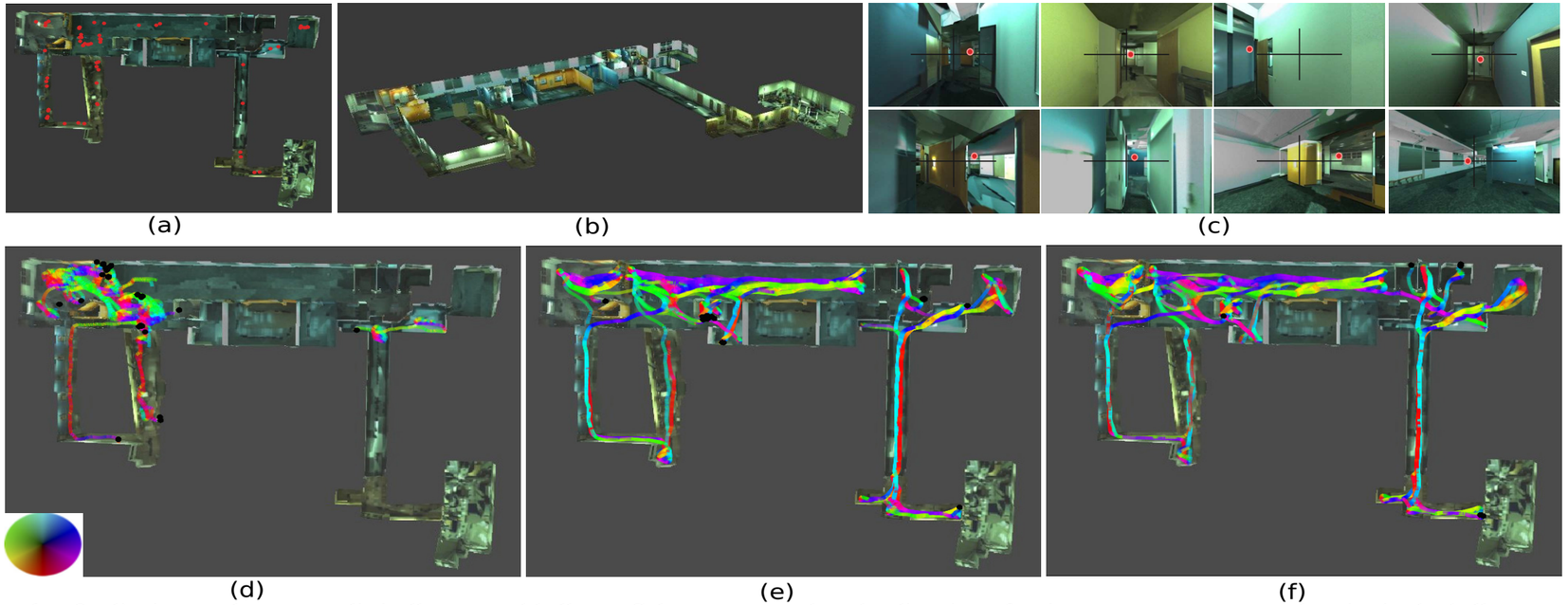


Fig. 6. Qualitative results on a realistically textured hallway. Colors correspond to the direction of trajectory movement at each point in the hallway as per the color wheel. (a) Red dots show flight initialization points (b) Overlook view of the hallway (c) Red dots show the control points produced by CAD^{2}RL. (d) LRS trajectories (e) Perception controller (FS-pred) trajectories (f) CAD^{2}RL trajectories.

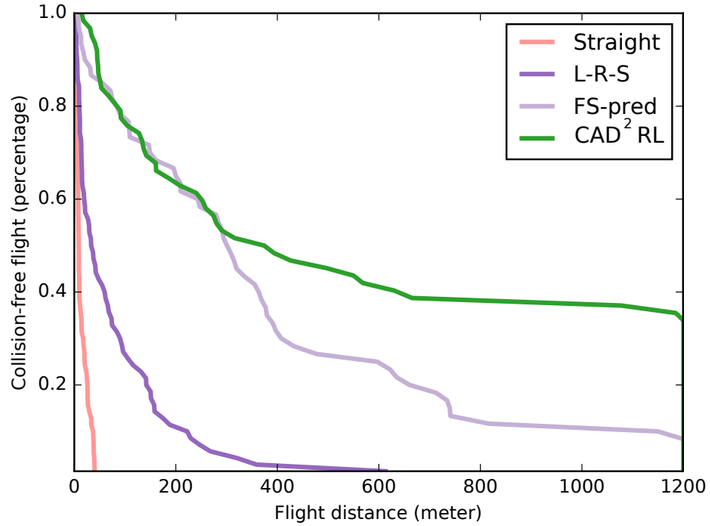


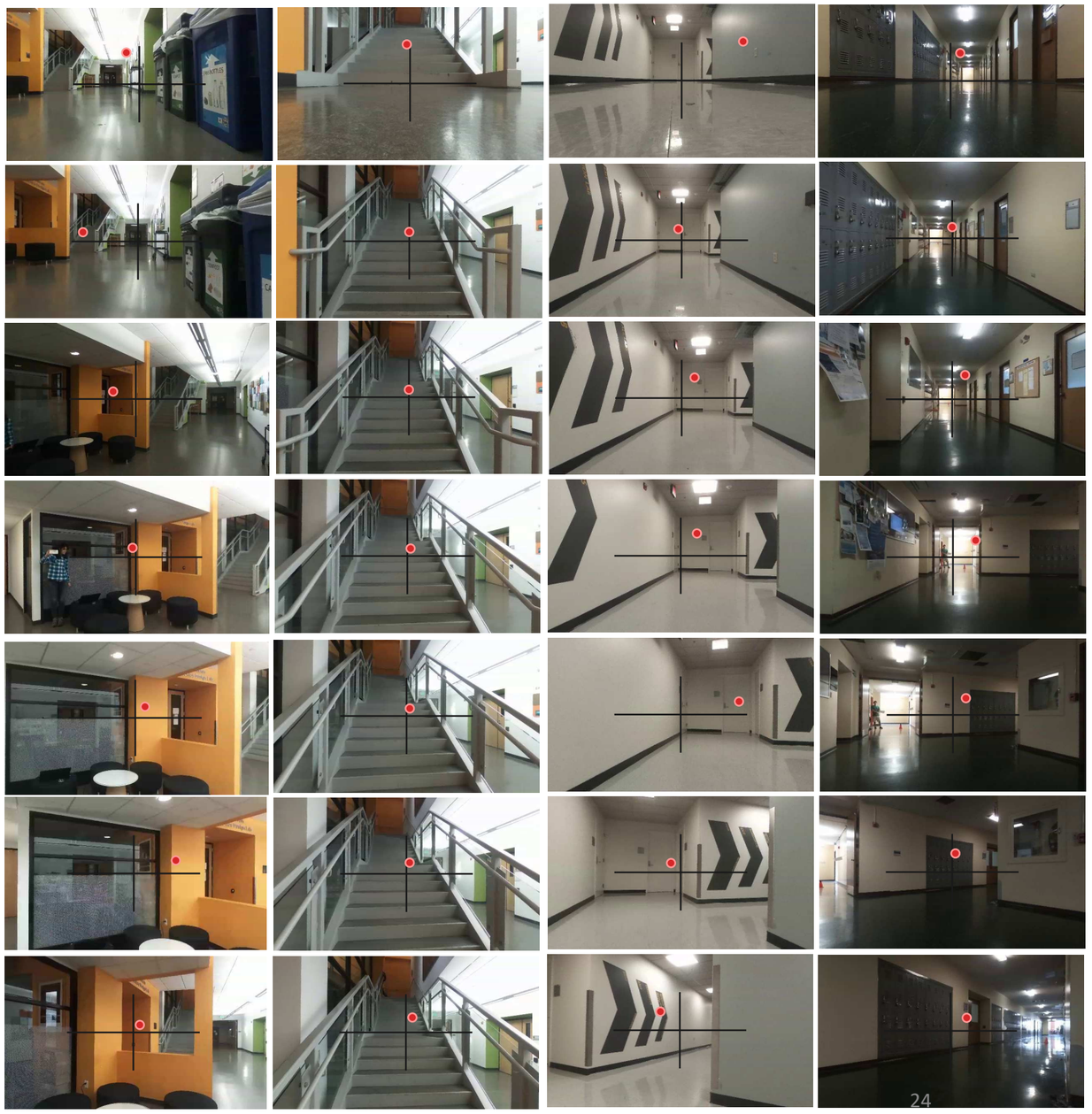
Fig. 5. Quantitative results on a realistically textured hallway. Our approach, CAD^{2}RL, outperforms the prior method (L-R-S) and other baselines.

Fereshteh Sadeghi and Sergey Levine. CAD2RL: Real Single-Image Flight without a Single Real Image. RSS 2017.

Results

TABLE I
REAL WORLD FLIGHT RESULTS.

Environment	Traveled Distance (meters)	Travel Time (minutes)	Collision (per meter)	Collision (per minute)	Safe Flight (meters)	Safe Flight (minutes)	Total Collisions
Cory FS-pred	162.458	12.01	0.080	1.081	12.496	0.924	13
Cory CAD^{2}RL	163.779	11.950	0.0366	0.502	27.296	1.991	6
SDH FS-pred	53.492	4.016	0.130	1.742	7.641	0.573	7
SDH CAD^{2}RL	54.813	4.183	0.072	0.956	13.703	1.045	4



Fereshteh Sadeghi and Sergey Levine. CAD2RL: Real Single-Image Flight without a Single Real Image. RSS 2017.

What acts as the bridge between sim and real?

The dataset is the bridge between sim and real.

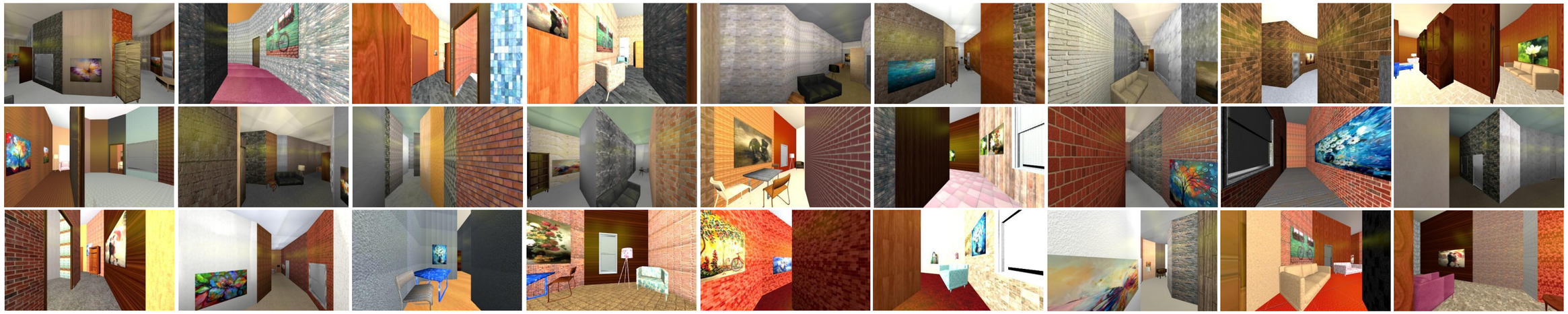


Fig. 2. Examples of rendered images using our simulator. We randomize textures, lighting and furniture placement to create a visually diverse set of scenes.

Fereshteh Sadeghi and Sergey Levine. CAD2RL: Real Single-Image Flight without a Single Real Image. RSS 2017.

Outline

- Domain and sim2real adaptation
- Domain randomisation I
- Domain randomisation II
- Recap on generative adversarial networks (GAN)
- Domain translation
- Modularisation and abstraction
- Latent space alignment
- Adversarial training

Generative adversarial networks (GAN)

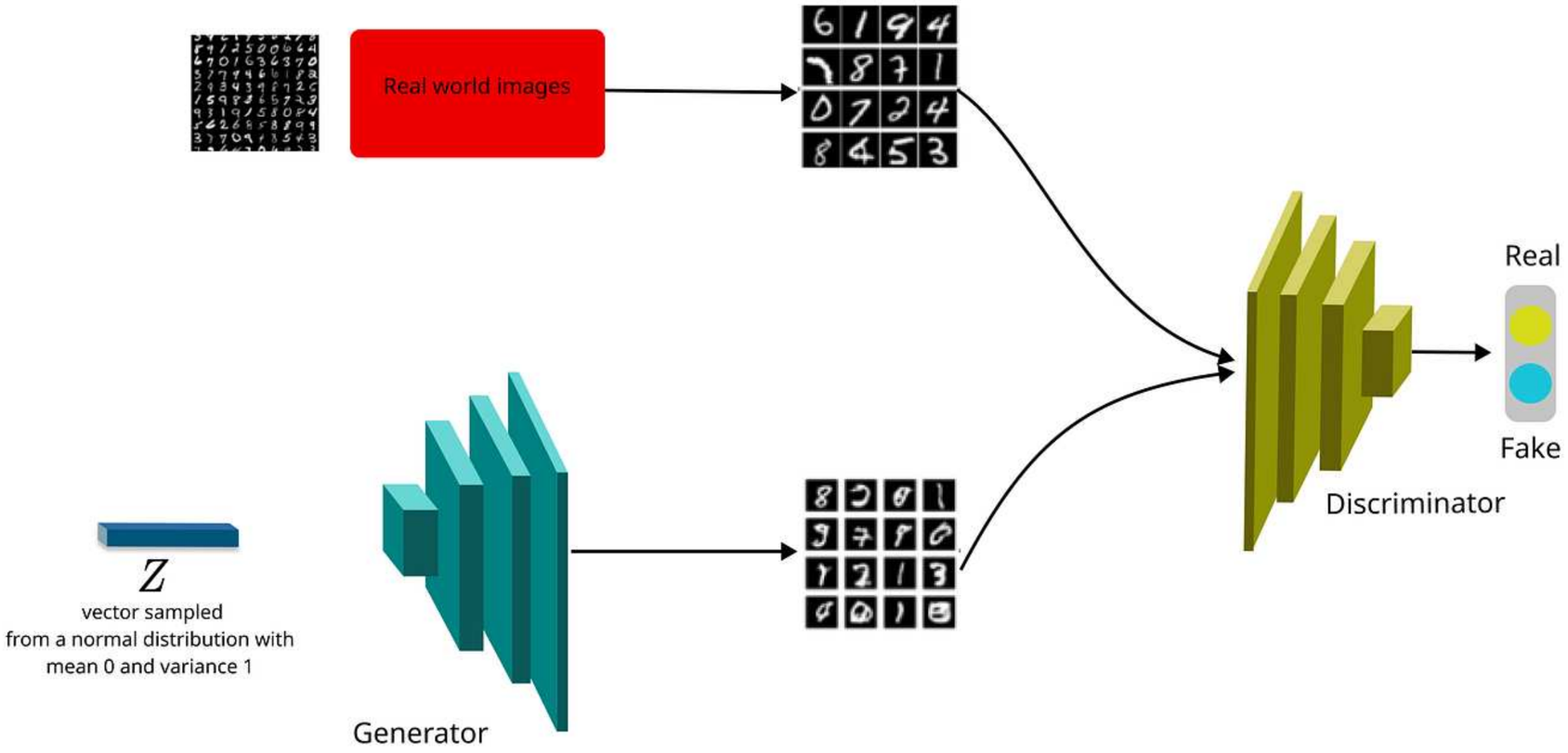
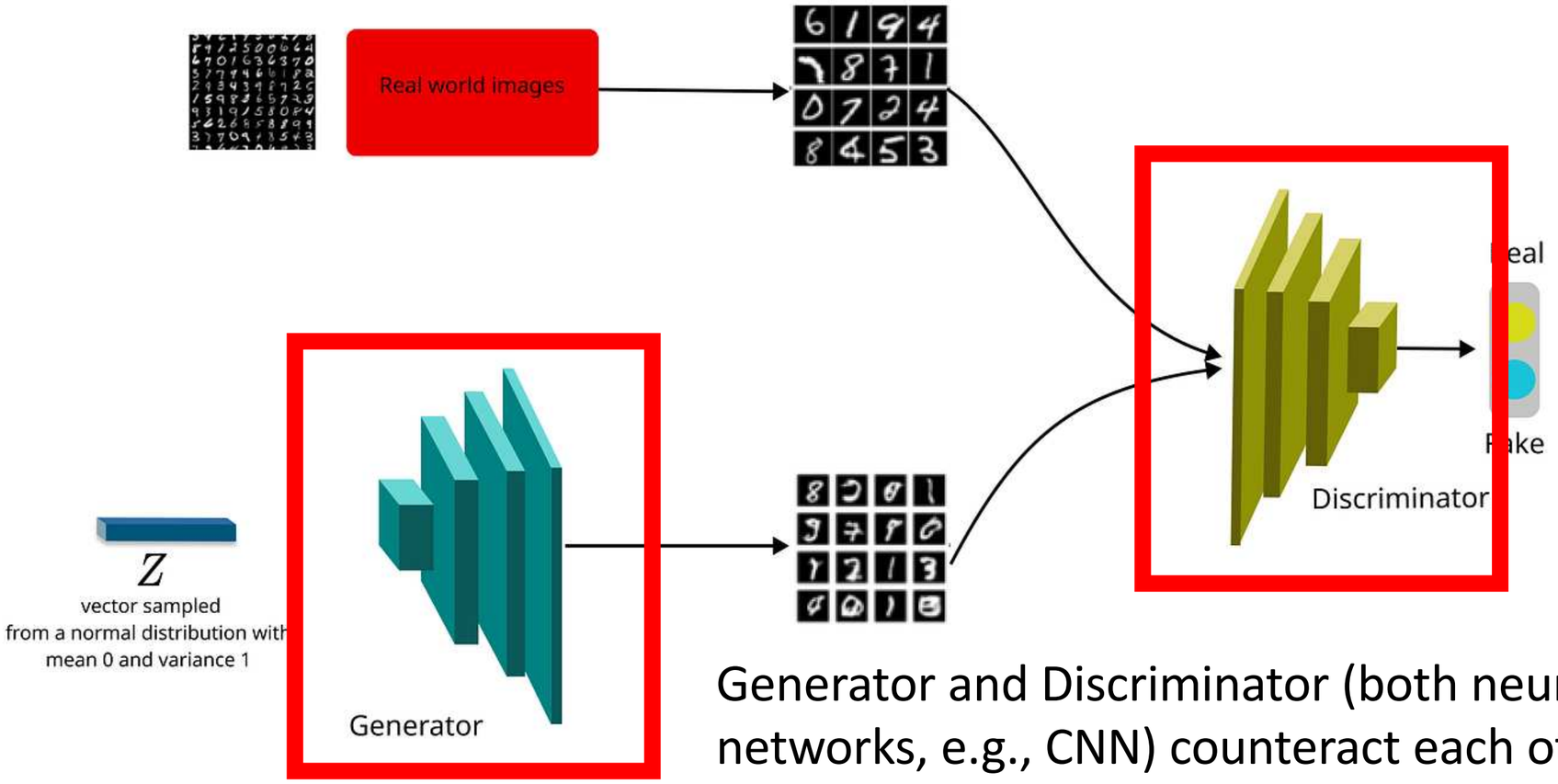


Diagram from https://chingisoinar.medium.com/generative-adversarial-networks-gan-introduction-and-example-3b66f5f235e9

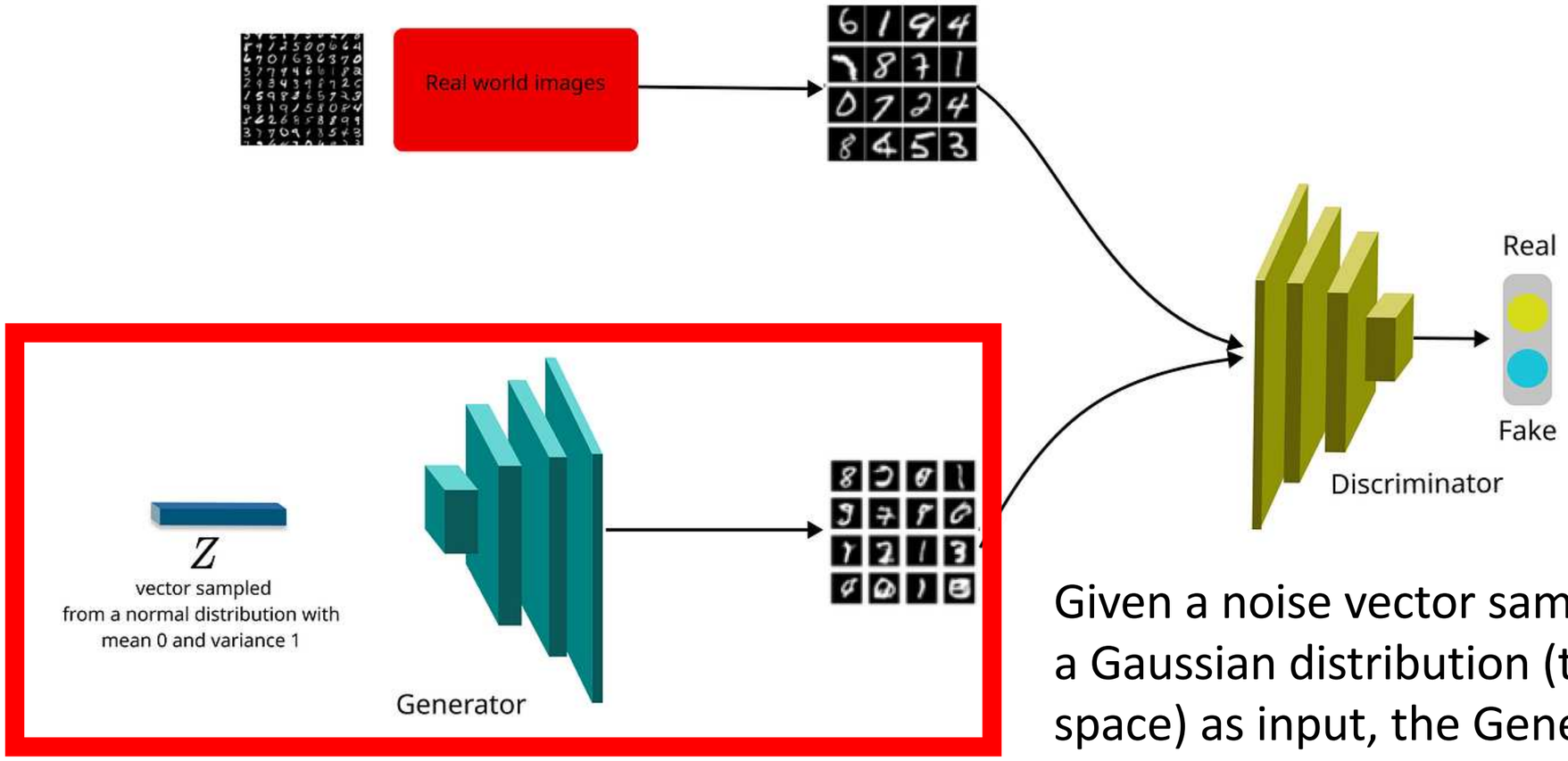
Generative adversarial networks (GAN)



Generator and Discriminator (both neural networks, e.g., CNN) counteract each other, === adversarial

Diagram from https://chingisoinar.medium.com/generative-adversarial-networks-gan-introduction-and-example-3b66f5f235e9

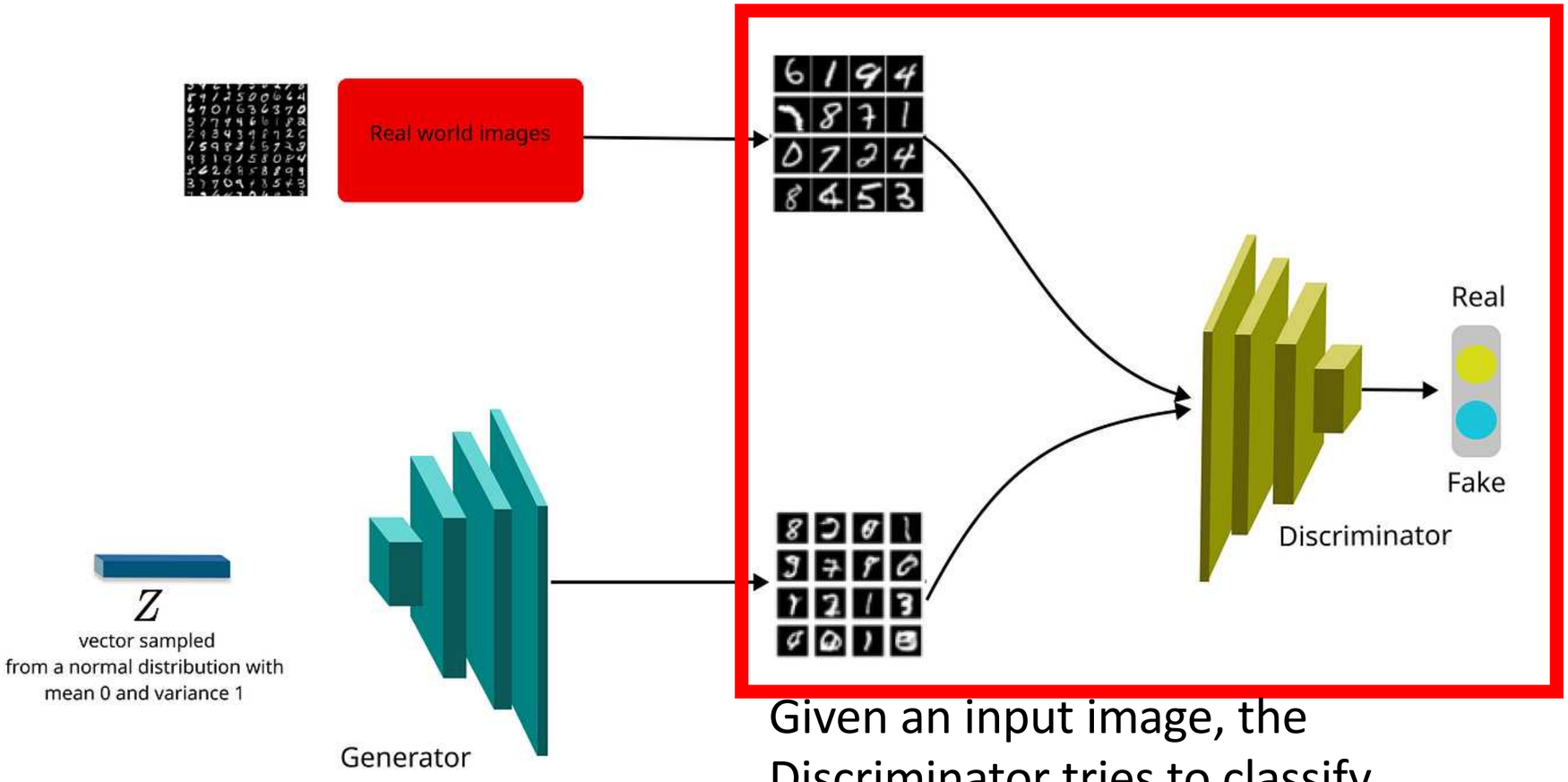
Generative adversarial networks (GAN)



Given a noise vector sampled from a Gaussian distribution (the latent space) as input, the Generator transforms it to a (fake) image.

Diagram from https://chingisoinar.medium.com/generative-adversarial-networks-gan-introduction-and-example-3b66f5f235e9

Generative adversarial networks (GAN)



Given an input image, the Discriminator tries to classify them as real or fake images.

Diagram from https://chingisoinar.medium.com/generative-adversarial-networks-gan-introduction-and-example-3b66f5f235e9

Training of GANs

- Objective function in training of GANs:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Training of GANs

- Objective function in training of GANs:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Indicates the adversarial relationship between Generator and Discriminator.

Training of GANs

- Objective function in training of GANs:

$$\min_{G} \square V(\square G) = \square \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Assuming Discriminator is well-trained….

… we train the Generator to produce images that the Discriminator thinks is real (target label = 1).

Recall that the Generator takes as input a random noise vector z.

Training of GANs

- Objective function in training of GANs:

$$\square \max_{D} V(D\square) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Assuming Generator is well-trained….

… we train the Discriminator to tell apart real images (target label = 1) and fake images (target label = 0).

Training of GANs

- Objective function in training of GANs:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

In the beginning, neither Generator nor Discriminator are well-trained, so we alternate between training Generator and Discriminator.

If the Discriminator fails half of the time, we conclude that the Generatoris well-trained.

Training of GANs

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Ian J. Goodfellow et al., Generative Adversarial Nets, Neurips 2014.

Inference/online phase

https://blog.acolyer.org/2018/05/10/progressive-growing-of-gans-for-improved-quality-stability-and-variation/

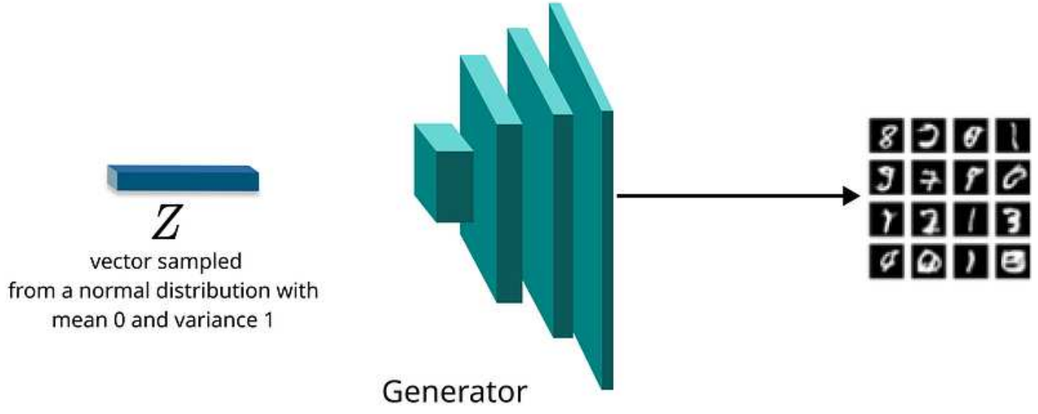


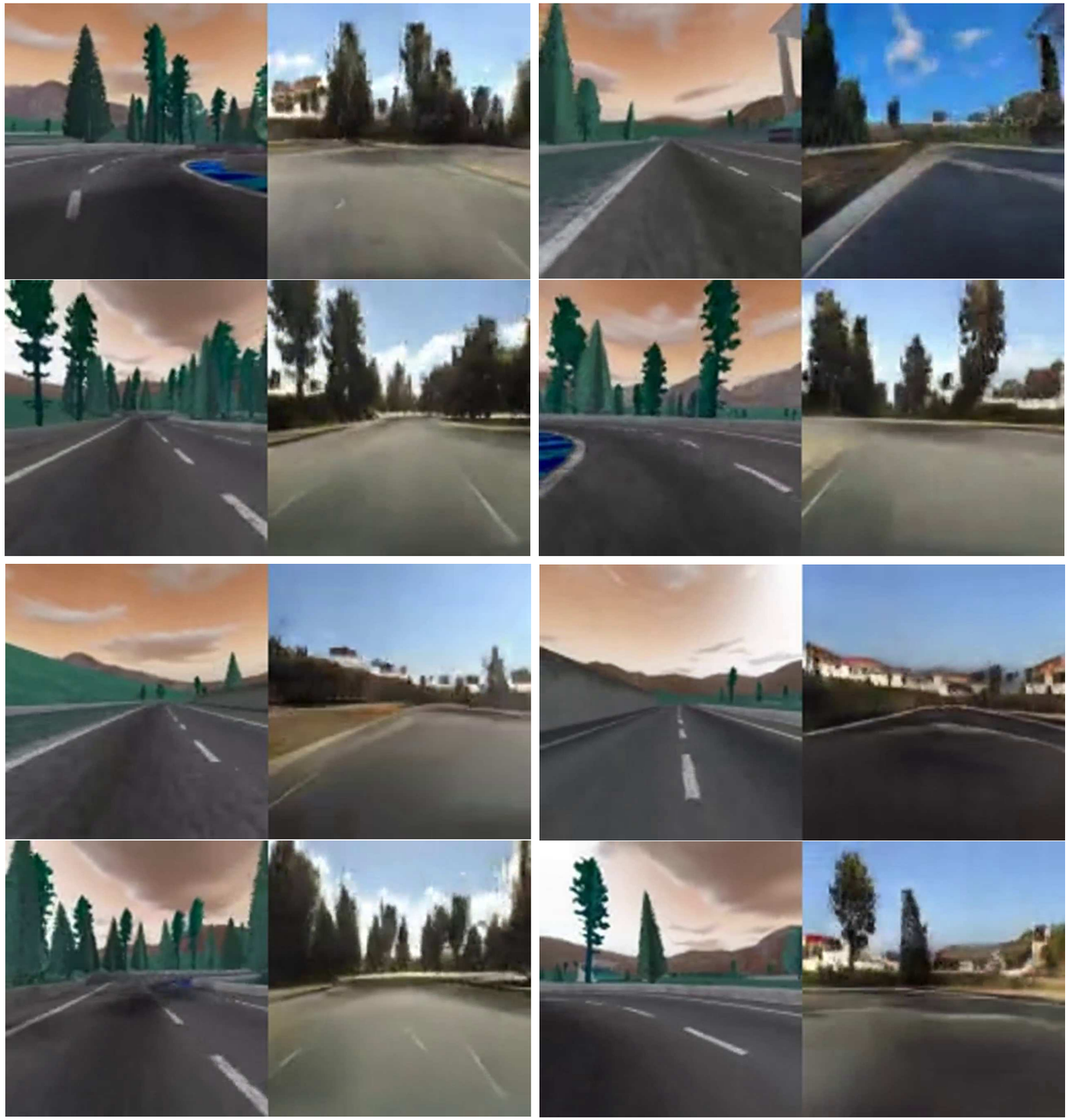
Diagram from https://chingisoinar.medium.com/generative-adversarial-networks-gan-introduction-and-example-3b66f5f235e9

Outline

- Domain and sim2real adaptation
- Domain randomisation I
- Domain randomisation II
- Recap on generative adversarial networks (GAN)
- Domain translation
- Modularisation and abstraction
- Latent space alignment
- Adversarial training

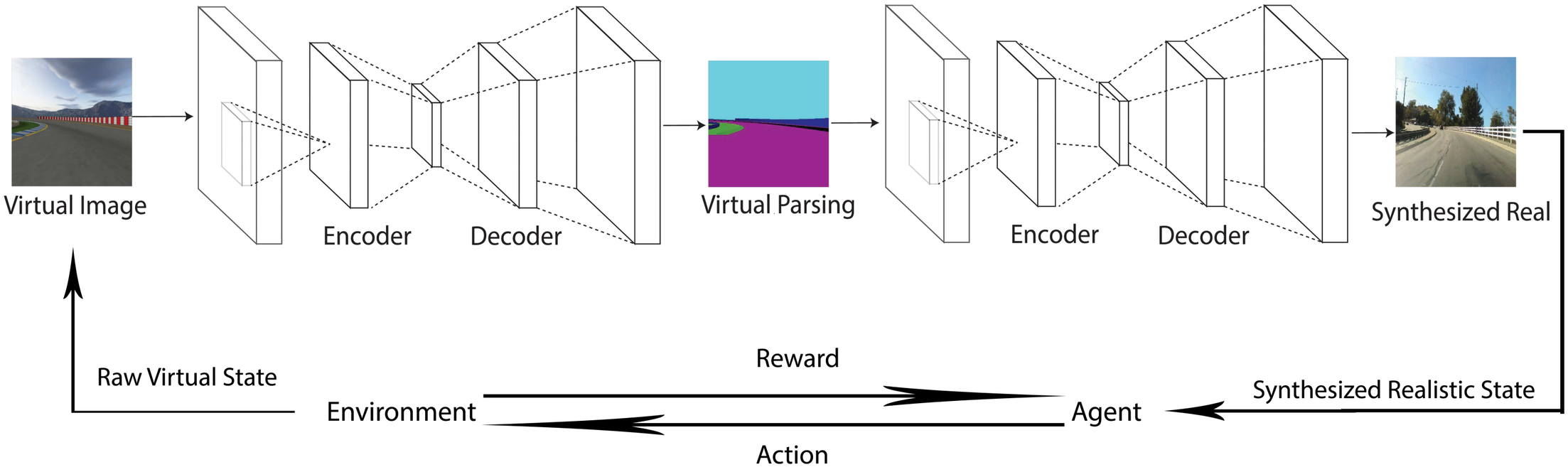
Domain translation

- Problem setting:
  - Train a driving agent using reinforcement learning in a virtual environment.
  - Driving agent takes as image as input and outputs steering angle.
- Key ideas:
  - Learn an image translation network that can convert synthetic images to real images.



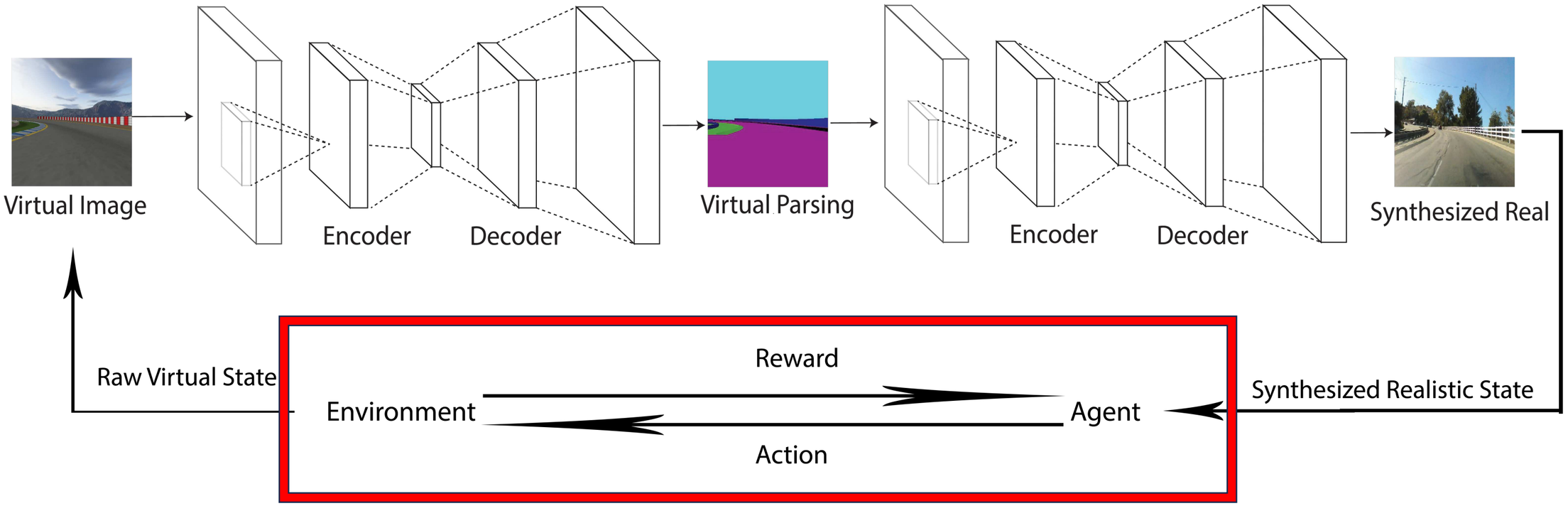
Yurong You, Xinlei Pan, Ziyan Wang, Cewu Lu. Virtual to Real Reinforcement Learning for Autonomous Driving. BMVC 2017.

Method



Yurong You, Xinlei Pan, Ziyan Wang, Cewu Lu. Virtual to Real Reinforcement Learning for Autonomous Driving. BMVC 2017.

Method

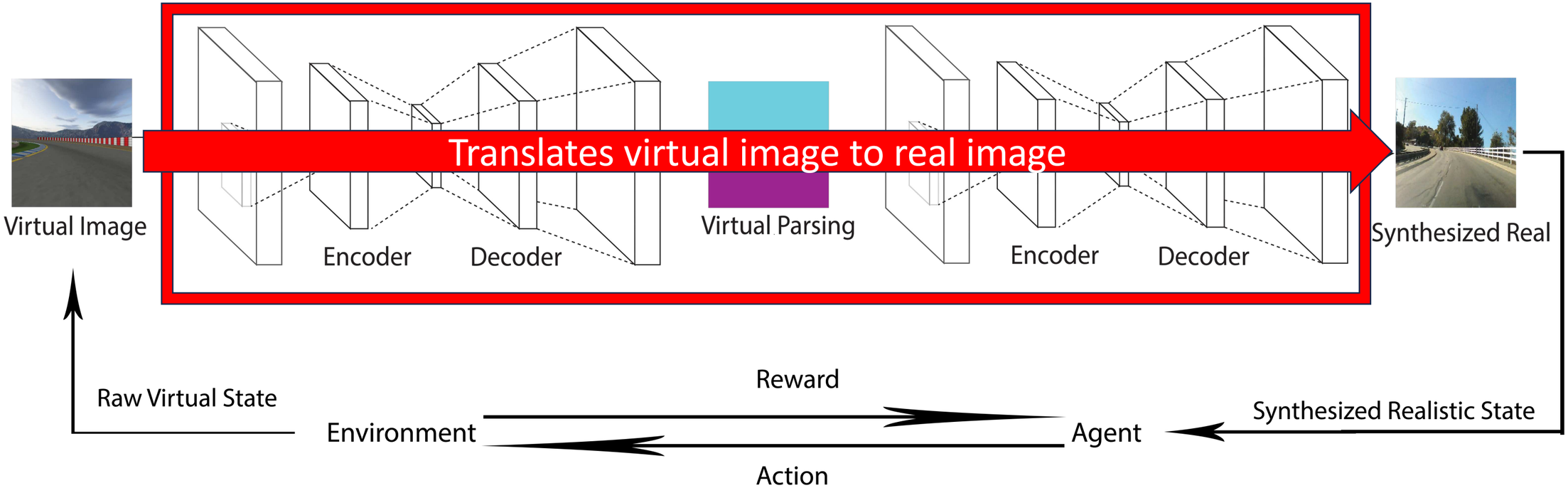


Optimise driving policy via RL using ground truth steering angles

Yurong You, Xinlei Pan, Ziyan Wang, Cewu Lu. Virtual to Real Reinforcement Learning for Autonomous Driving. BMVC 2017.

Method

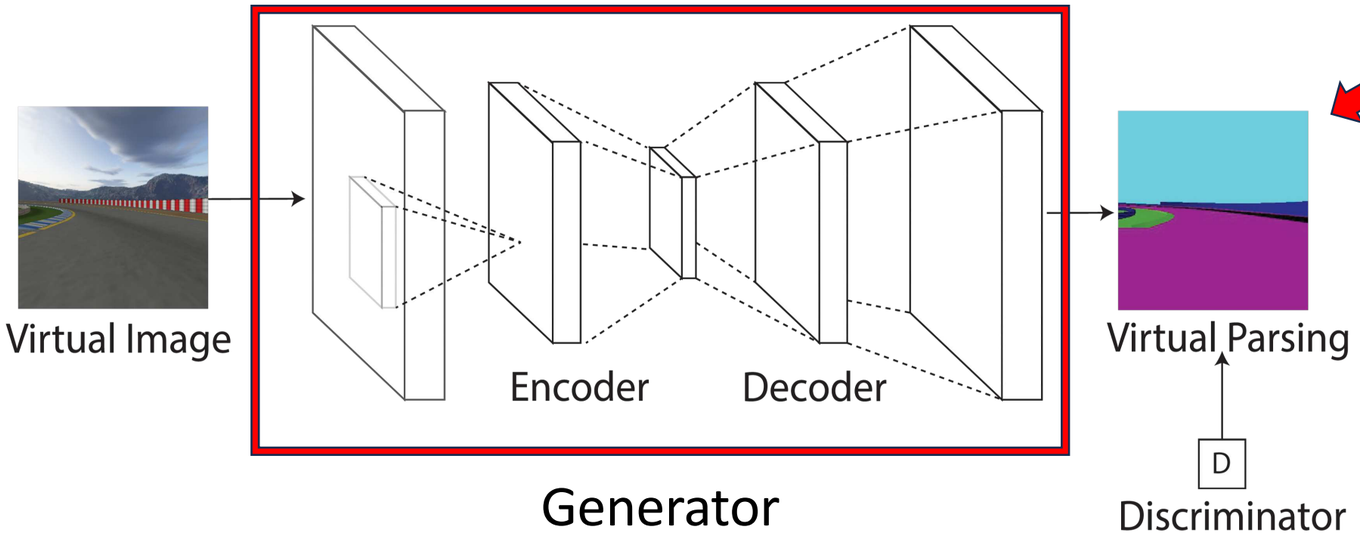
Image translation network:
Consists two cGANS



Yurong You, Xinlei Pan, Ziyan Wang, Cewu Lu. Virtual to Real Reinforcement Learning for Autonomous Driving. BMVC 2017.

Conditional Generative Adversarial Network (cGAN) #1

Image from virtual environment (from TORCS car racing simulator)



Semantic segmentation: obtain for "free" from the virtual environment

Generator:

$$G : \{x, z\} \rightarrow s.$$

maps image and noise vector to another image.

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,s\sim p_{data}(x,s)}[\log D(x, s)] + \mathbb{E}_{x\sim p_{data}(x),z\sim p_z(z)}[\log(1 - D(x, G(x, z)))]$$

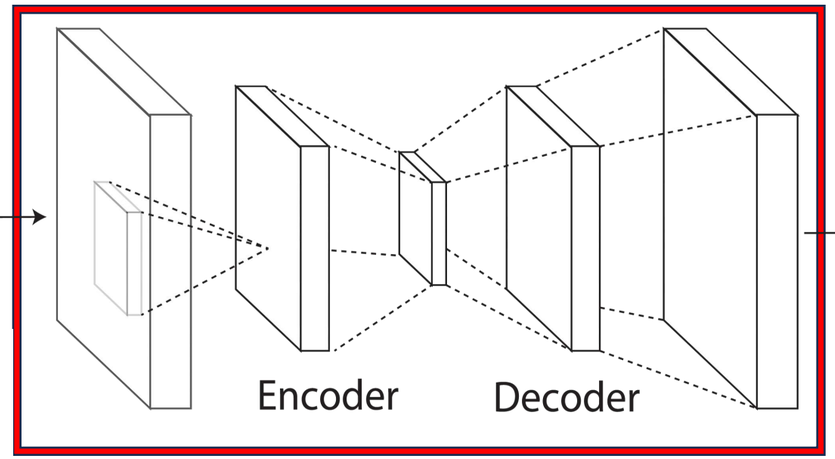
$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,s\sim p_{data}(x,s),z\sim p_z(z)}[\|s - G(x, z)\|_1]$$

Train cGAN using <virtual image, segmentation> image pairs.

Yurong You, Xinlei Pan, Ziyan Wang, Cewu Lu. Virtual to Real Reinforcement Learning for Autonomous Driving. BMVC 2017.

Conditional Generative Adversarial Network (cGAN) #2

Segmentation results on CityScape images (real)



Generator

Synthesized Real

D

Discriminator

CityScape images (real)

Generator:

$G : \{x, z\} \rightarrow s$

maps image and noise vector to another image.

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda \mathcal{L}_{L1}(G)$$

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,s\sim p_{data}(x,s)}[\log D(x,s)] + \mathbb{E}_{x\sim p_{data}(x),z\sim p_z(z)}[\log(1 - D(x, G(x,z)))]$$

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,s\sim p_{data}(x,s),z\sim p_z(z)}[\|s - G(x,z)\|_1]$$

Train cGAN using <real image, segmentation> image pairs.

Yurong You, Xinlei Pan, Ziyan Wang, Cewu Lu. Virtual to Real Reinforcement Learning for Autonomous Driving. BMVC 2017.

Semantic segmentation as the sim2real bridge

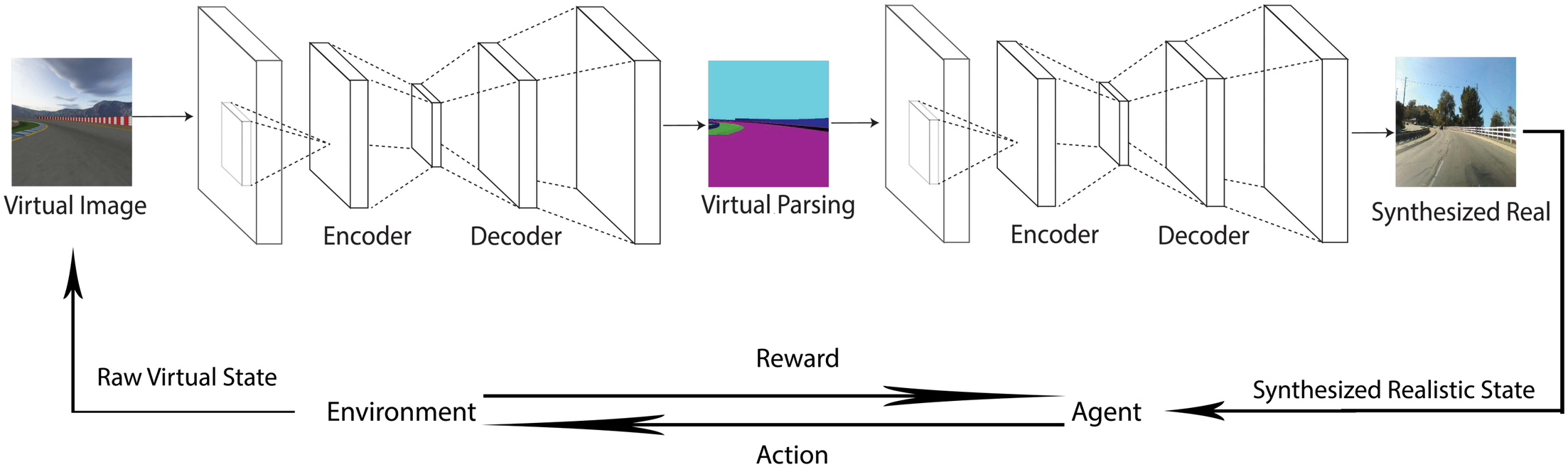
Virtual

Real

Why not train a cGAN to translate virtual to real images directly?

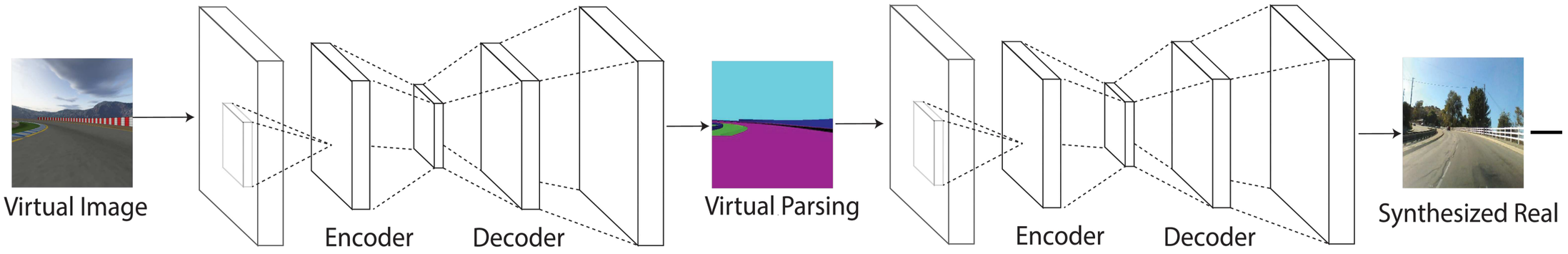
Yurong You, Xinlei Pan, Ziyan Wang, Cewu Lu. Virtual to Real Reinforcement Learning for Autonomous Driving. BMVC 2017.

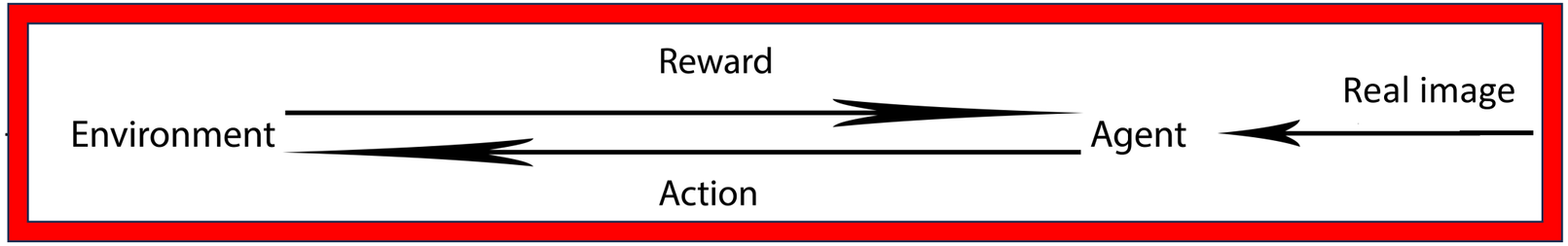
Inference/online phase?



Yurong You, Xinlei Pan, Ziyan Wang, Cewu Lu. Virtual to Real Reinforcement Learning for Autonomous Driving. BMVC 2017.

Inference/online phase?





Yurong You, Xinlei Pan, Ziyan Wang, Cewu Lu. Virtual to Real Reinforcement Learning for Autonomous Driving. BMVC 2017.

Results

Steering angle prediction on real images:

B-RL: Baseline with no adaptation

SV: Supervised learning with true labels

Table 1: Action prediction accuracy for the three methods.

Accuracy	Ours	B-RL	SV
Dataset in [5]	43.40%	28.33%	53.60%

Adapting between different virtual environments:

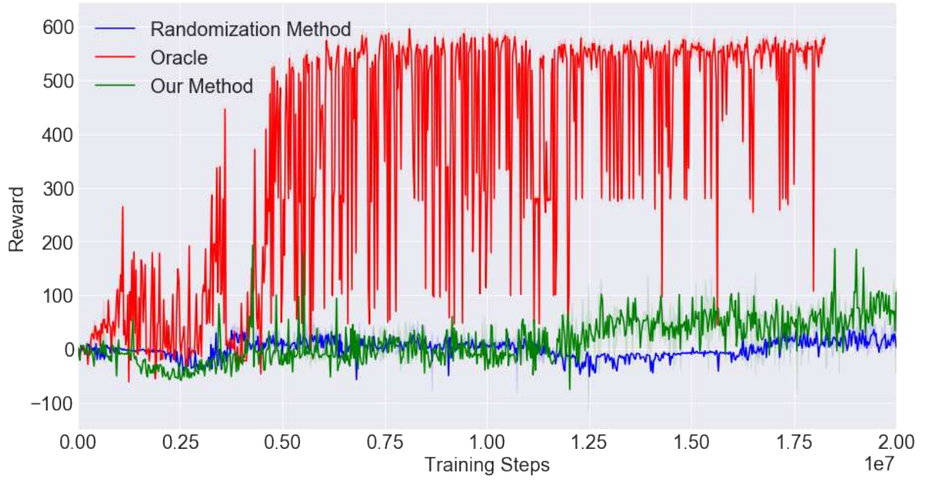
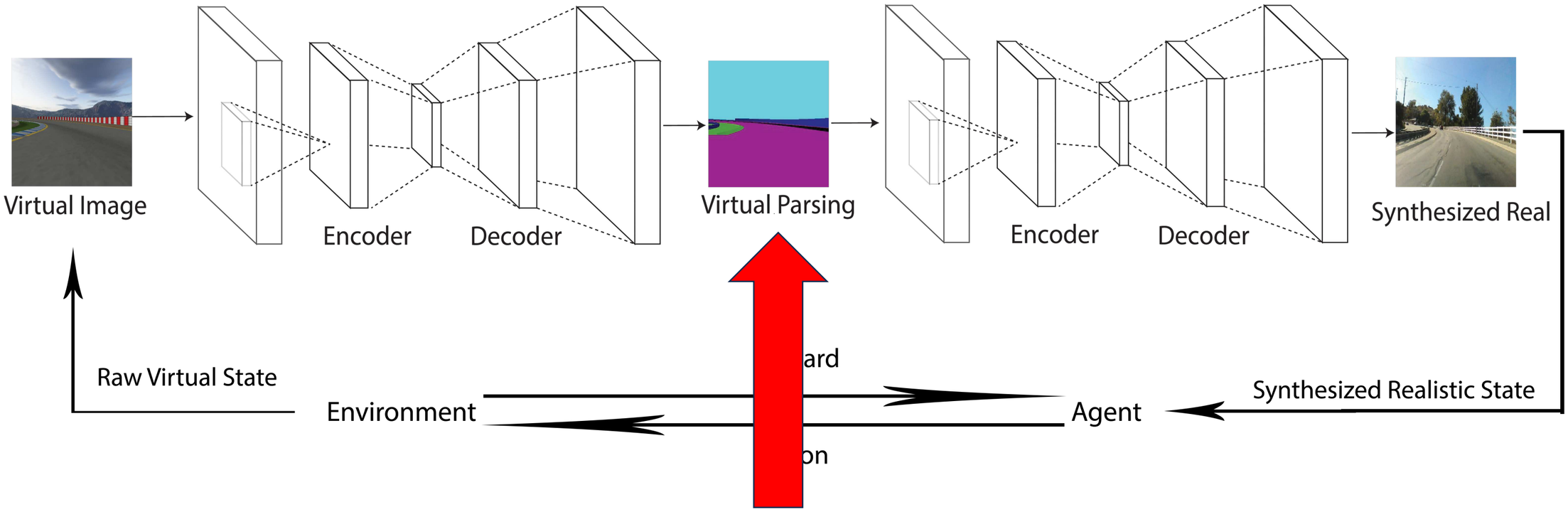


Figure 5: Transfer learning between different environments. Oracle was trained in Cgtrack2 and tested in Cgtrack2, so its performance is the best. Our model works better than the domain randomization RL method. Domain randomization method requires training in multiple virtual environments, which imposes significant manual engineering work.

Yurong You, Xinlei Pan, Ziyan Wang, Cewu Lu. Virtual to Real Reinforcement Learning for Autonomous Driving. BMVC 2017.

What acts as the bridge between sim and real?



Segmentation representation is the <u>bridge</u> between sim and real.

Yurong You, Xinlei Pan, Ziyan Wang, Cewu Lu. Virtual to Real Reinforcement Learning for Autonomous Driving. BMVC 2017.

Outline

- Domain and sim2real adaptation
- Domain randomisation I
- Domain randomisation II
- Recap on generative adversarial networks (GAN)
- Domain translation
- Modularisation and abstraction
- Latent space alignment
- Adversarial training

Modularisation and abstraction

- Problem setting:
  - Train an end-to-end waypoint predictor that takes images and high-level commands as inputs (i.e., conditional imitation learning).
  - Convert predicted waypoints into control signals using PID controller.
- Key ideas:
  - Split the waypoint predictor into perception module and driving policy module.
  - Expose the perception module to real images, while train the driving policy using simulation.

Figure 11: Controlled evaluation of road following performance in environment 2. Top: a composite image showing five starting positions as seen from the finish line. Bottom: close-ups of the five starting positions. The positions are clearly marked for consistency across episodes.

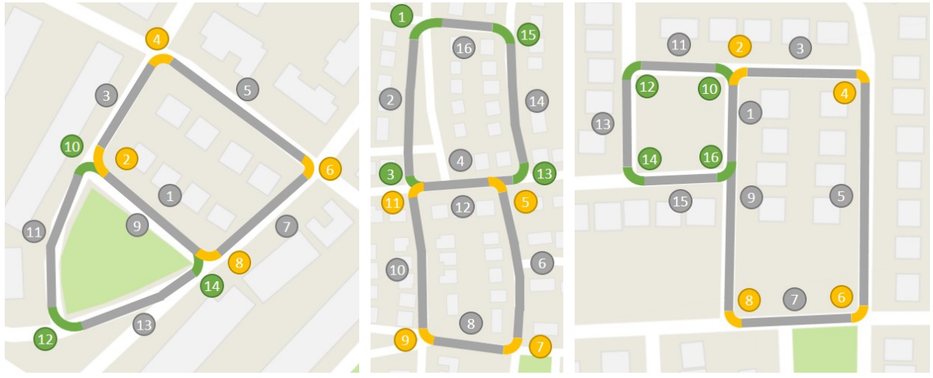
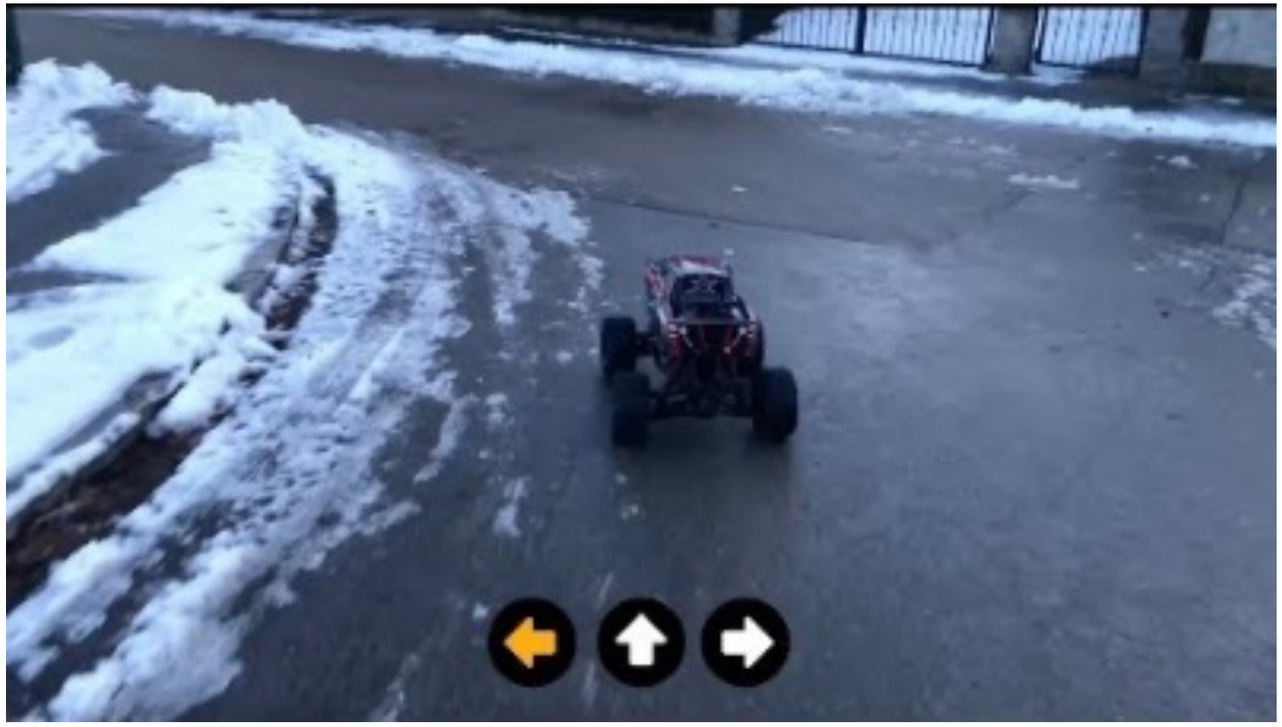


Figure 12: Three routes used for evaluating the driving policy. Right turns marked in yellow, left turns in green.

Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, V. Koltun. Driving Policy Transfer via Modularity and Abstraction. CORL 2018.

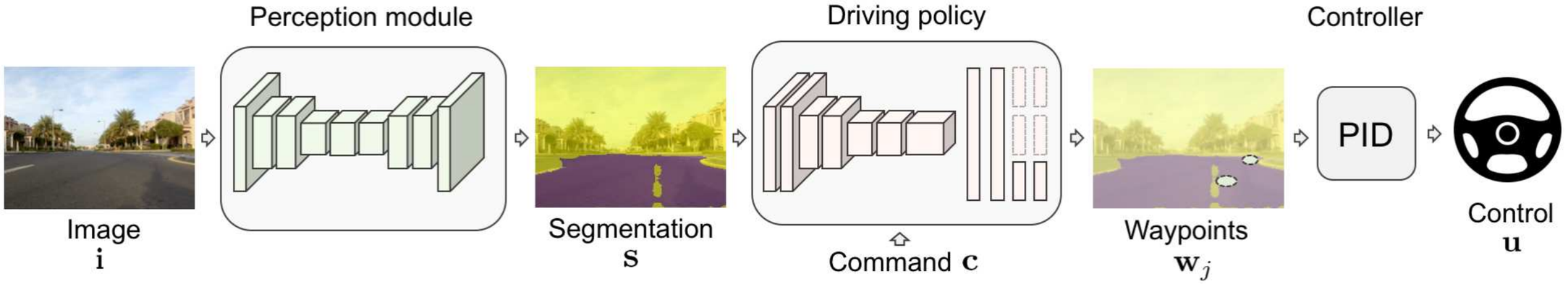
Modularisation and abstraction

- Problem setting:
  - Train an end-to-end waypoint predictor that takes images and high-level commands as inputs (i.e., conditional imitation learning).
  - Convert predicted waypoints into control signals using PID controller.
- Key ideas:
  - Split the waypoint predictor into perception module and driving policy module.
  - Expose the perception module to real images, while train the driving policy using simulation.



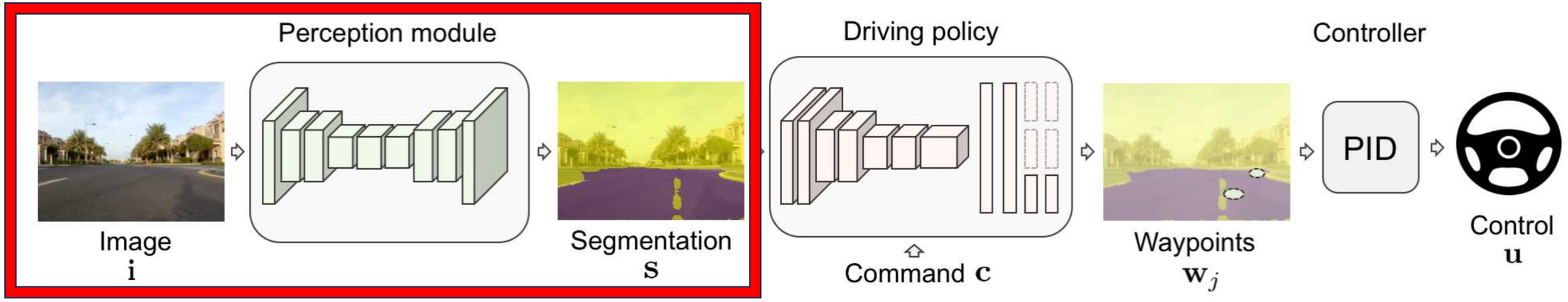
Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, V. Koltun. Driving Policy Transfer via Modularity and Abstraction. CORL 2018.

Method



Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, V. Koltun. Driving Policy Transfer via Modularity and Abstraction. CORL 2018.

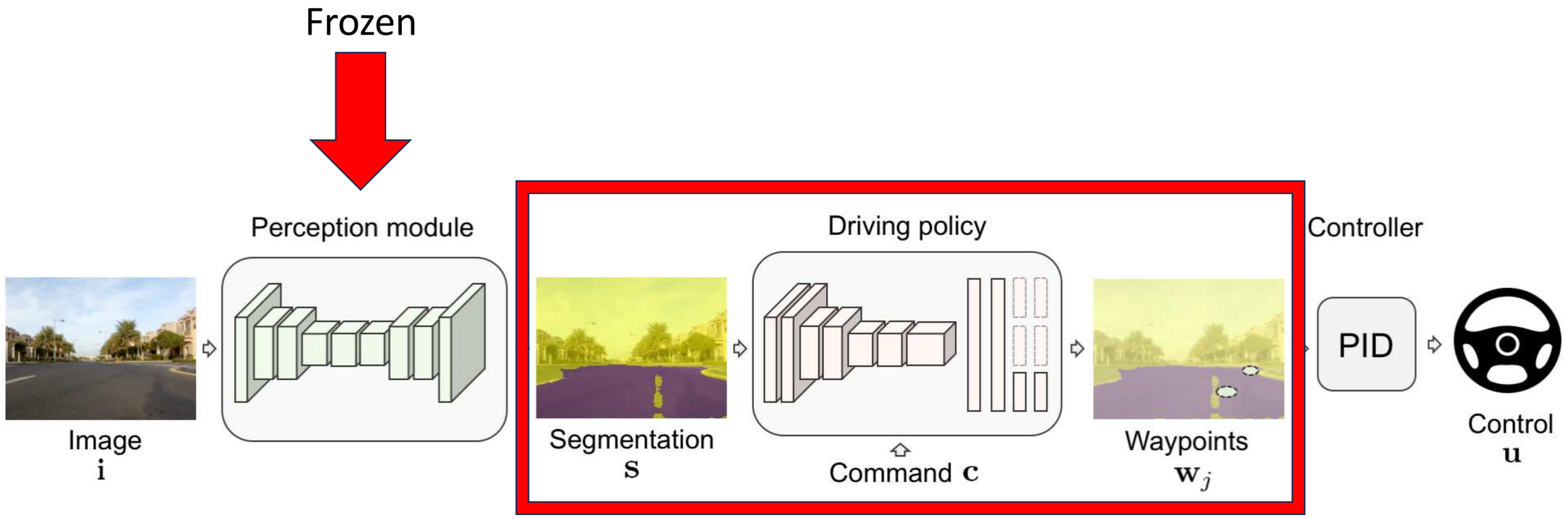
Method

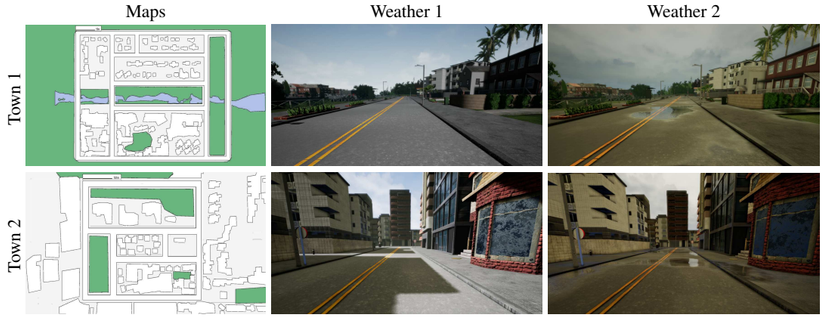


Trained on Cityscapes (real images)

Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, V. Koltun. Driving Policy Transfer via Modularity and Abstraction. CORL 2018.

Method





Trained in simulation environment with:

- Ground truth waypoints from expert driver.
- Semantic segmentation results on synthetic images predicted using perception module trained using real images.

Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, V. Koltun. Driving Policy Transfer via Modularity and Abstraction. CORL 2018.

Semantic segmentation on synthetic and real images

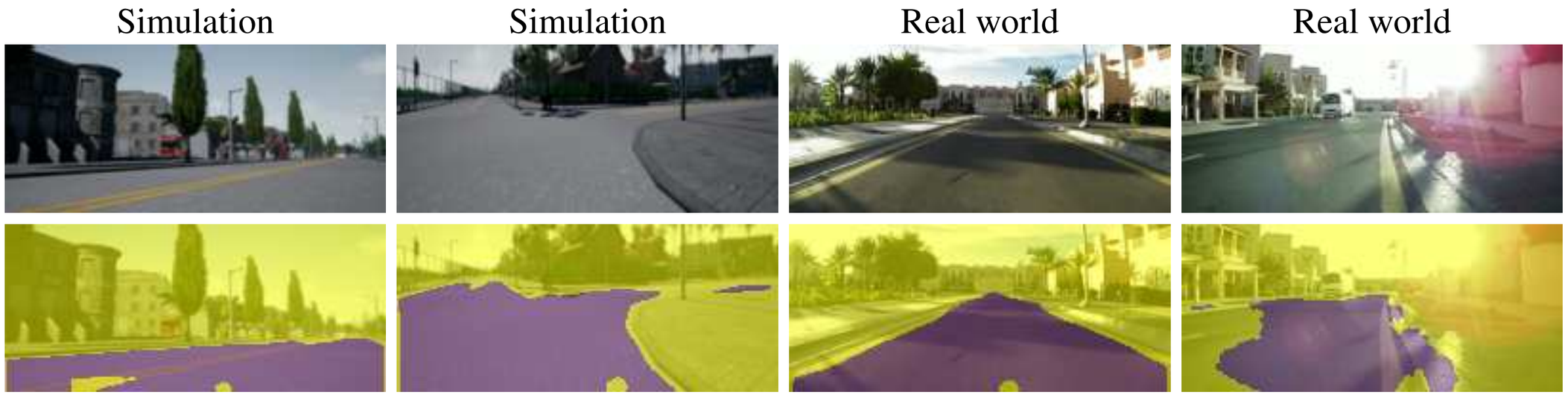


Figure 7: Sample outputs of the segmentation network trained on Cityscapes and tested in simulation and in the real world. The images are at the resolution used by the network – 200×88 pixels. The network works well in typical scenes both in simulation and in the real world, but accuracy drops under complex lighting conditions and in unusual situations.

Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, V. Koltun. Driving Policy Transfer via Modularity and Abstraction. CORL 2018.

Definition of waypoints

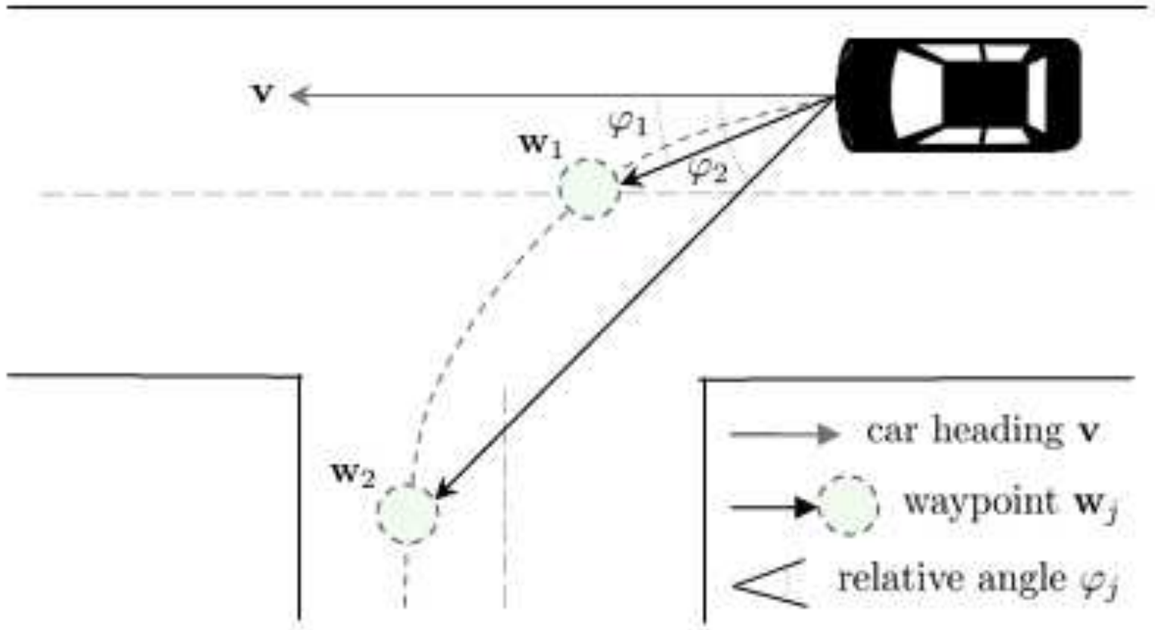
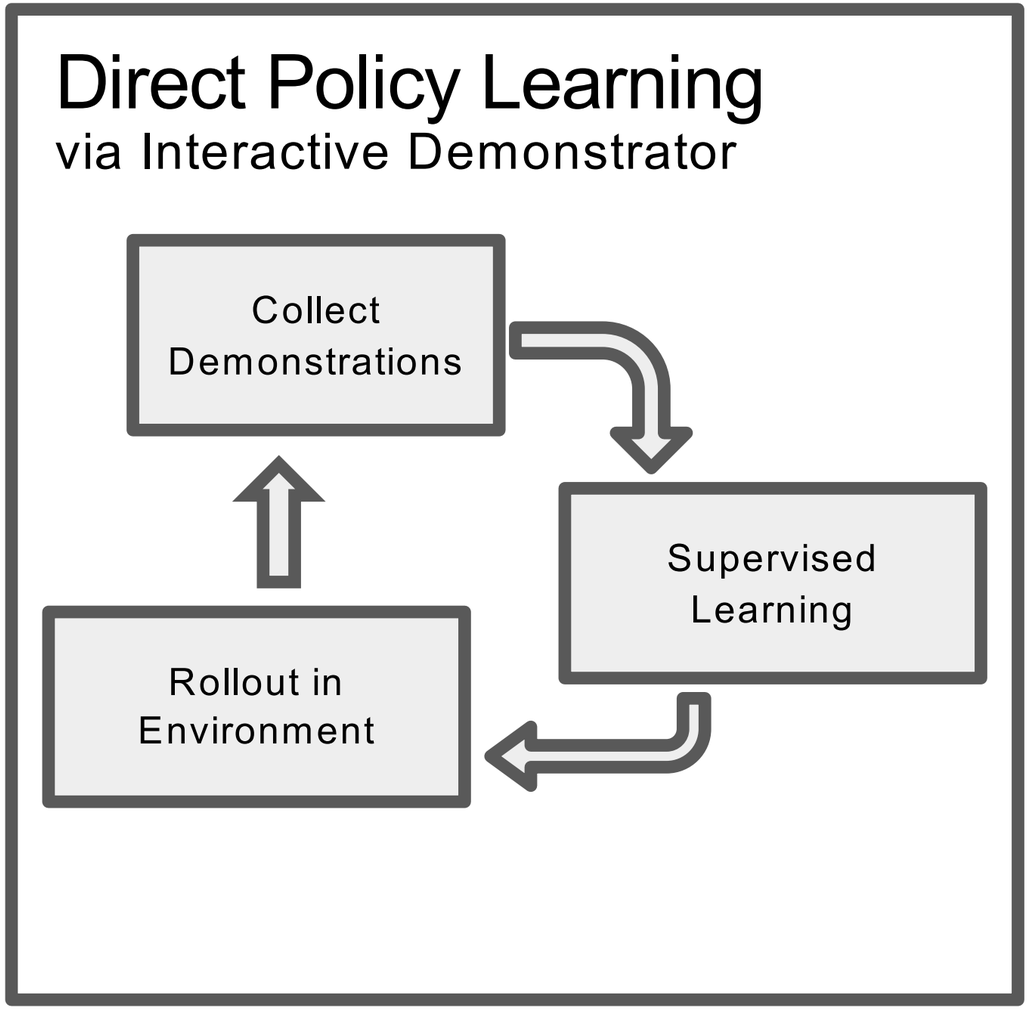


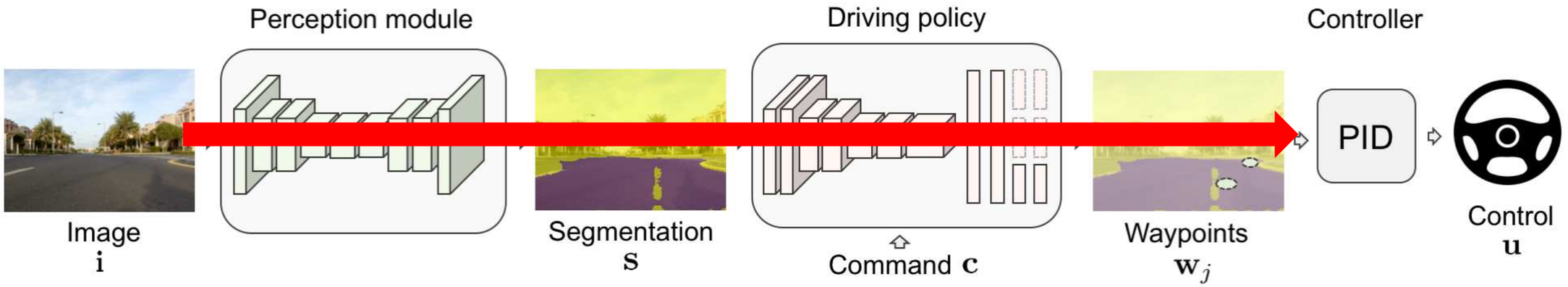
Figure 2: Waypoints are encoded by the distance to the vehicle and the relative angle to the vehicle's heading.

Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, V. Koltun. Driving Policy Transfer via Modularity and Abstraction. CORL 2018.

Rollout in virtual environment



Inference/online phase?



Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, V. Koltun. Driving Policy Transfer via Modularity and Abstraction. CORL 2018.

Results

- Driving in simulation

- **Image to control (img2ctrl):** Predicts low-level control directly from color images.
- **Image to waypoint (img2wp):** Predicts waypoints directly from color images.
- **Segmentation to control (seg2ctrl):** We pre-train the perception module on Cityscapes and fix it. We then train a driving policy to predict low-level control from segmentation maps produced by the perception module.
- **Segmentation to waypoint (ours):** Our full model predicts waypoints from segmentation maps produced by the perception module.

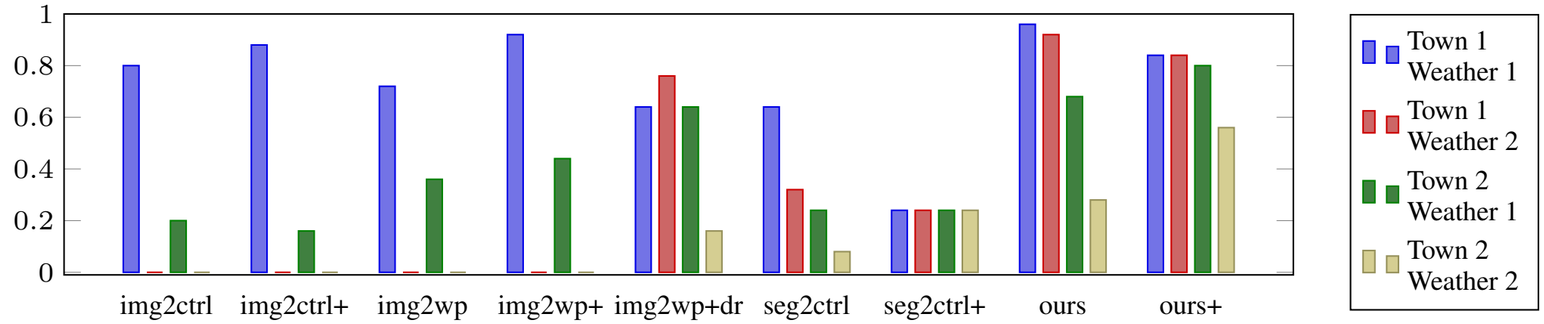


Figure 4: Quantitative evaluation of goal-directed navigation in simulation. We report the success rate over 25 navigation trials in four town-weather combinations. The models have been trained in Town 1 and Weather 1. The evaluated models are: *img2ctrl* – predicting low-level control from color images; *img2wp* – predicting waypoints from color images; *seg2ctrl* – predicting low-level control from the segmentation produced by the perception module; *ours* – predicting waypoints from the segmentation produced by the perception module. Suffix '+' denotes models trained with data augmentation, and '+dr' denotes the model trained with domain randomization.

Data augmentation. To improve the robustness of the learned policy, we follow Codevilla et al. [8] and introduce noise into the controls in approximately 20% of the data. We additionally randomize the camera parameters and perform data augmentation, as described in the supplement.

Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, V. Koltun. Driving Policy Transfer via Modularity and Abstraction. CORL 2018.

Results

- Driving in the real world

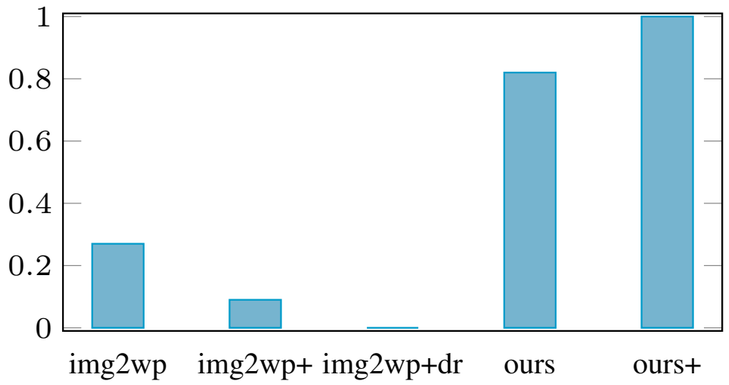


Figure 5: Quantitative evaluation of road following in the real world. We report the average success rate over a total of 11 navigation trials, with distance to be driven varying from 10 to 50 meters. Notation follows Figure 4.

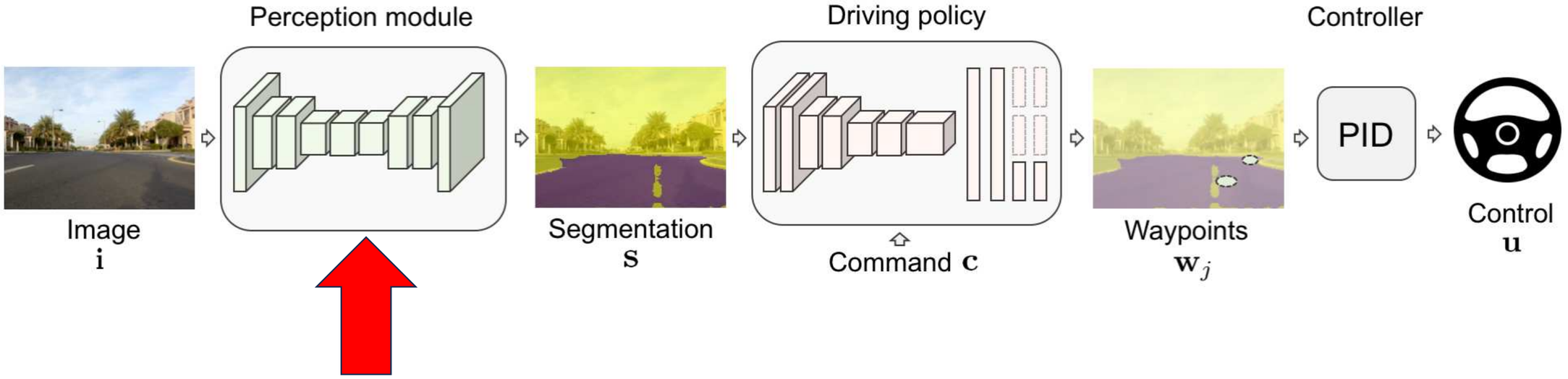
Figure 6: Some of the environments the truck was tested in. Note the variation in scene structure, weather, and lighting. Qualitative results are shown in the supplementary video.

Table 1: Evaluation of our method on three long routes in an urban environment.

				Infractions	
Route	Length	Time	Missed turns	Severe	Mild
1	1.0 km	4:12	1/7	0	2
2	0.7 km	3:05	1/8	0	3
3	1.1 km	5:08	2/8	1	5

Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, V. Koltun. Driving Policy Transfer via Modularity and Abstraction. CORL 2018.

What acts as the bridge between sim and real?



The trained perception module (CNN) is the bridge between sim and real.

Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, V. Koltun. Driving Policy Transfer via Modularity and Abstraction. CORL 2018.

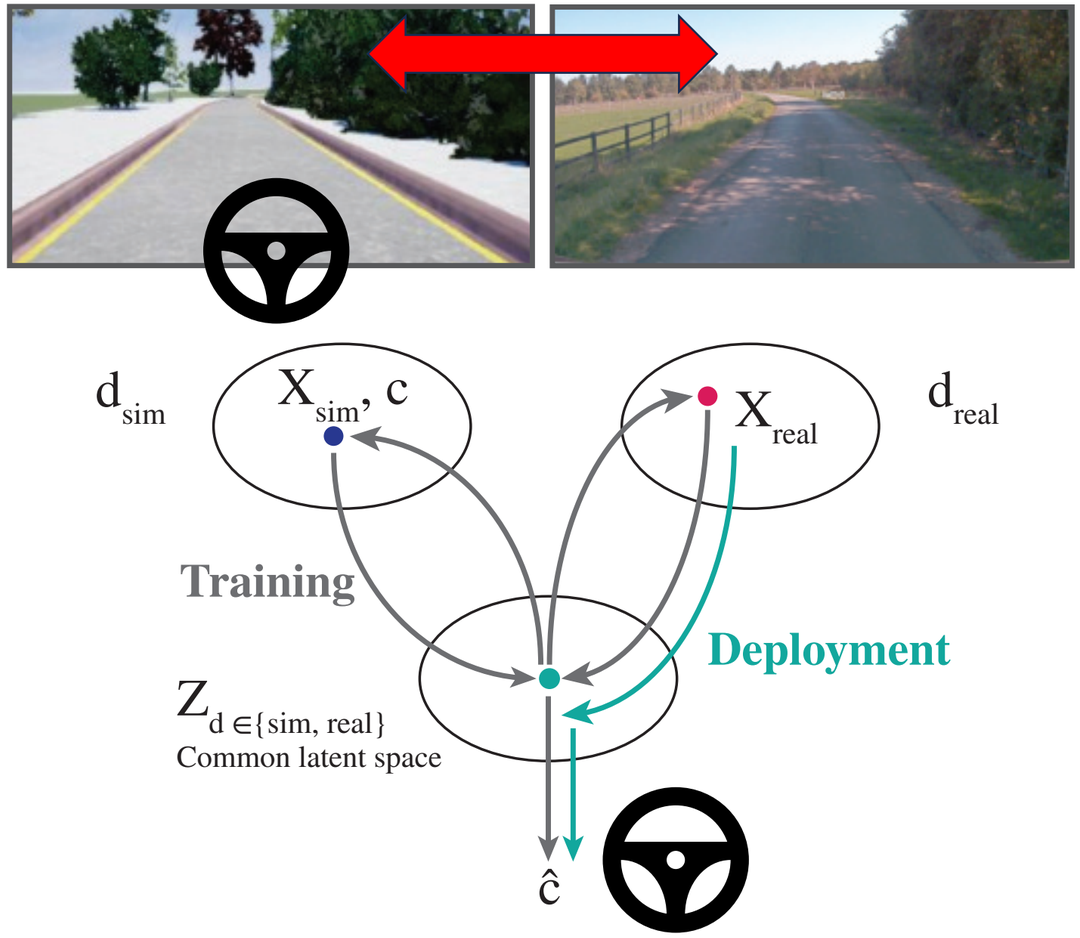
Outline

- Domain and sim2real adaptation
- Domain randomisation I
- Domain randomisation II
- Recap on generative adversarial networks (GAN)
- Domain translation
- Modularisation and abstraction
- Latent space alignment
- Adversarial training

Latent space alignment

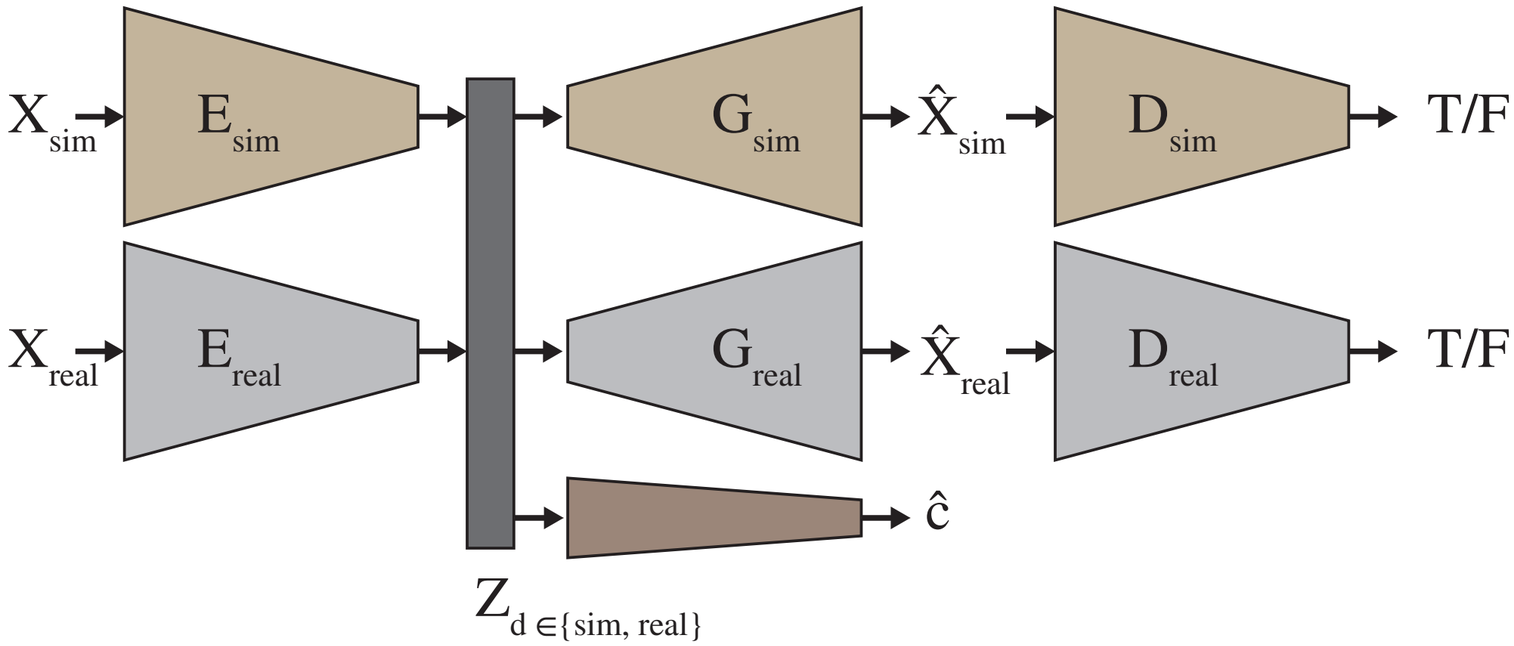
- Problem setting:
  - Train a driving controller that predicts a steering angle from input images.
  - The controller is trained using labelled synthetic images and unlabelled real images.
    - ➔Unsupervised sim2real DA.
    - ➔Synthetic and real images are unpaired.
- Key ideas:
  - Train a latent representation that is common for synthetic and real images.

This example shows paired synthetic-real images, but it is not a requirement.



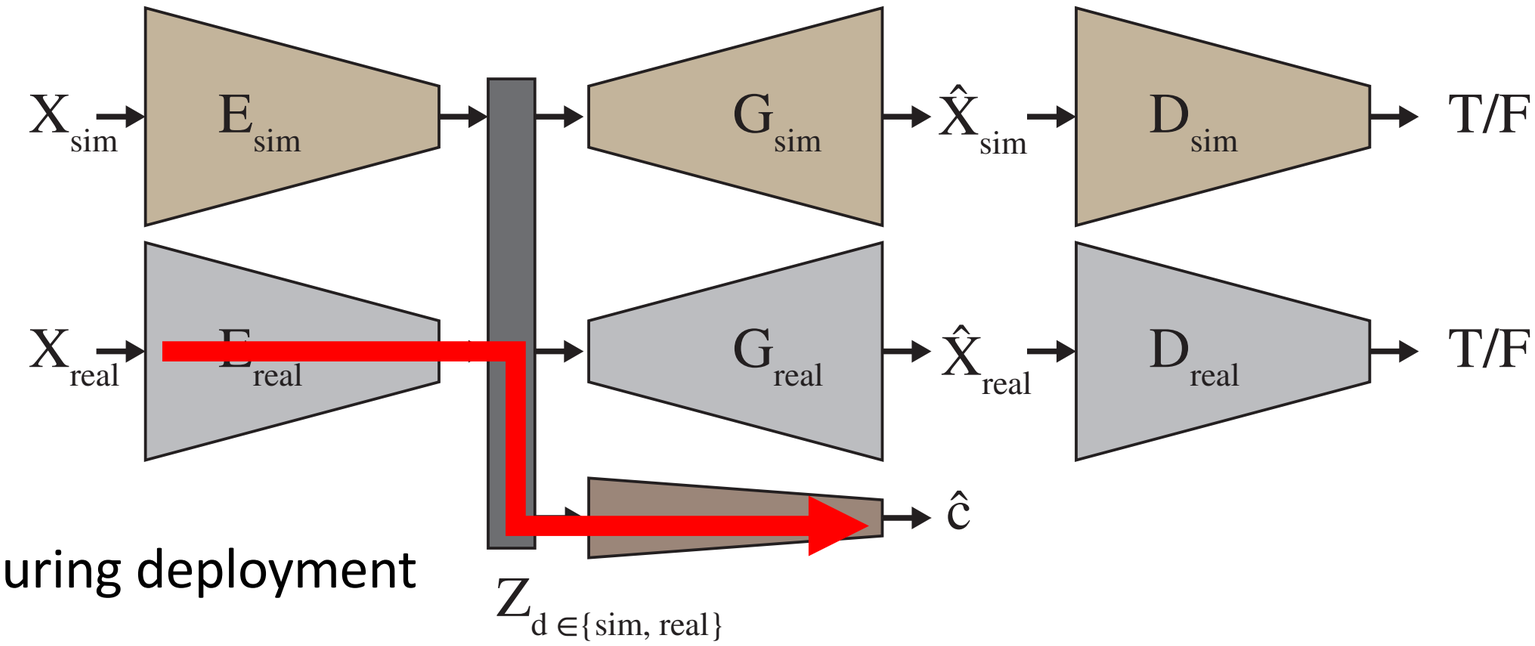
A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

Architecture



A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

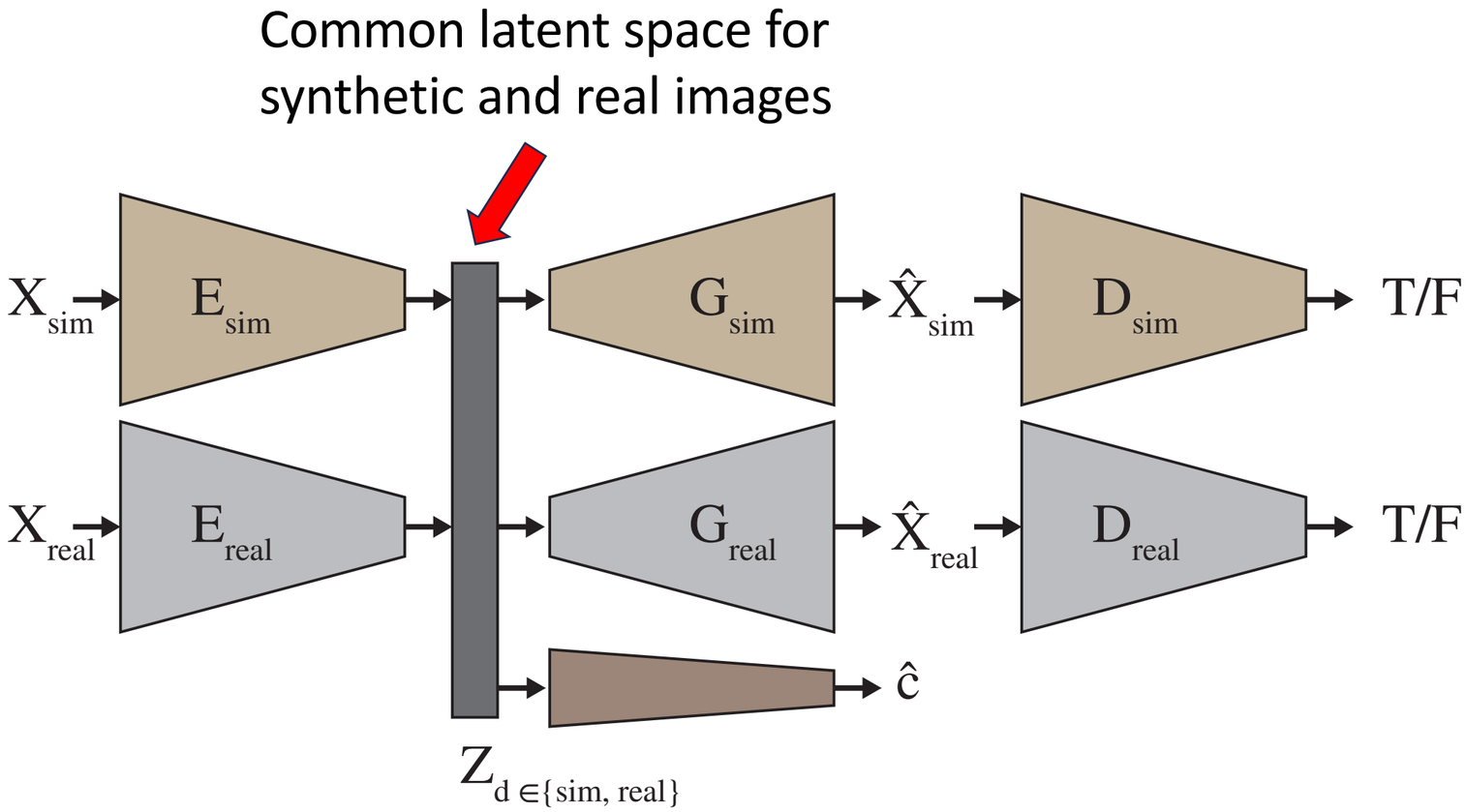
Architecture



Used during deployment

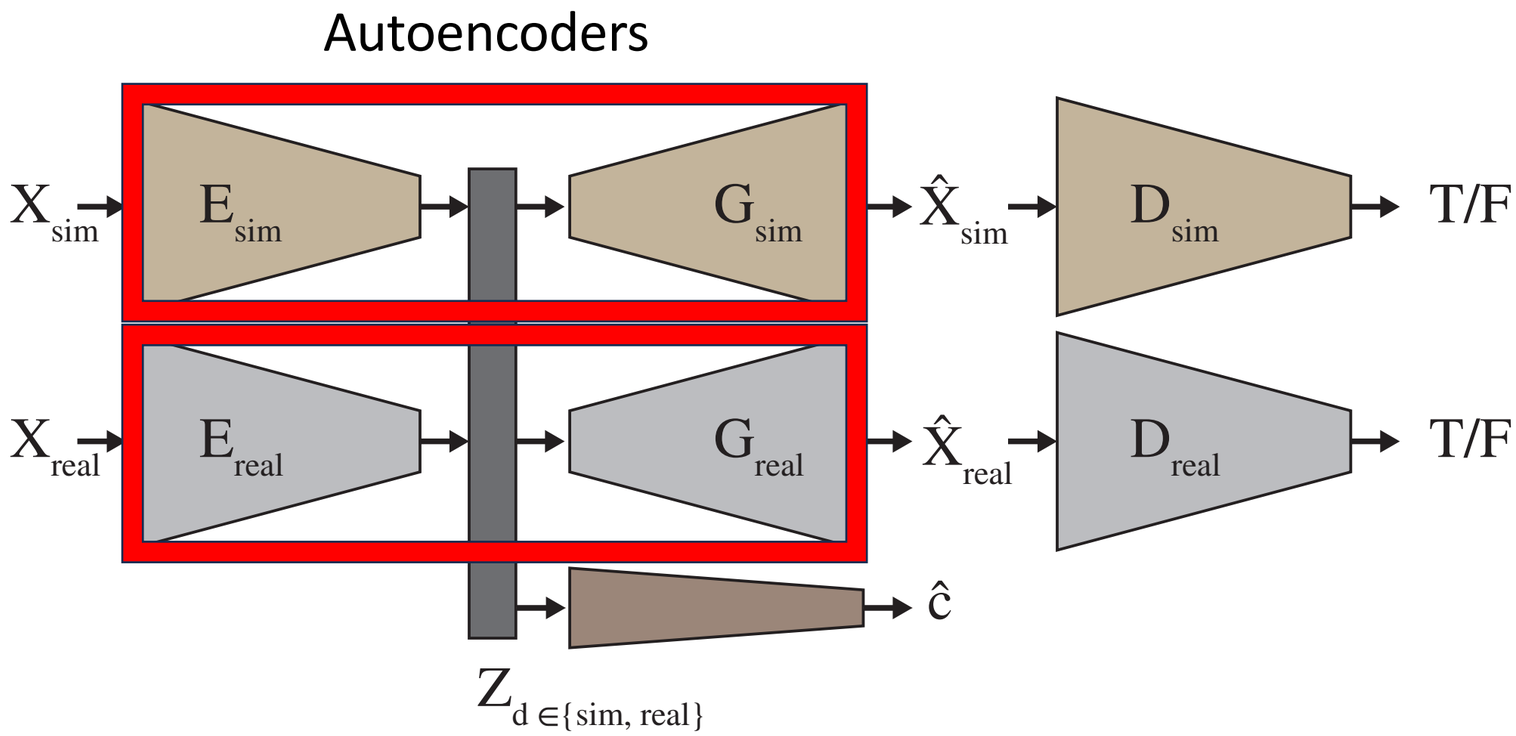
A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

Architecture



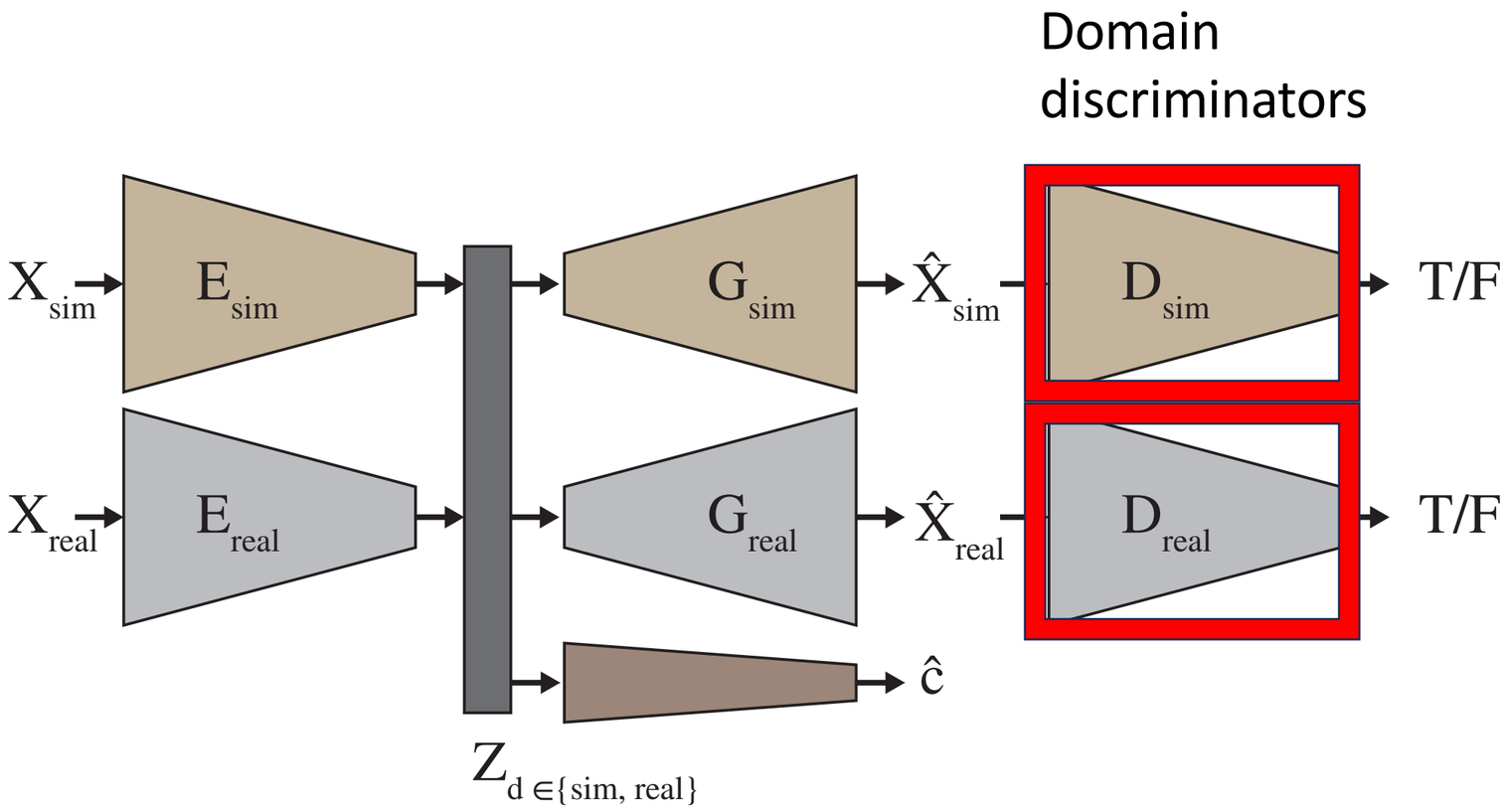
A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

Architecture



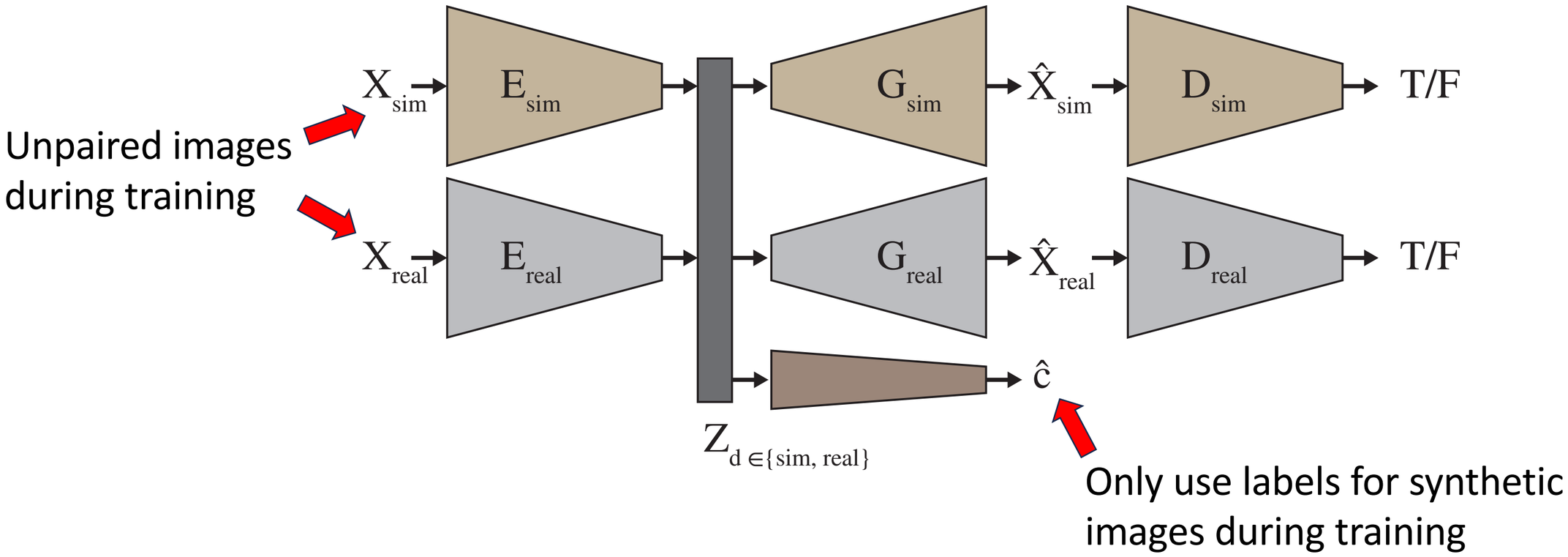
A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

Architecture



A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

Architecture



A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

Loss functions

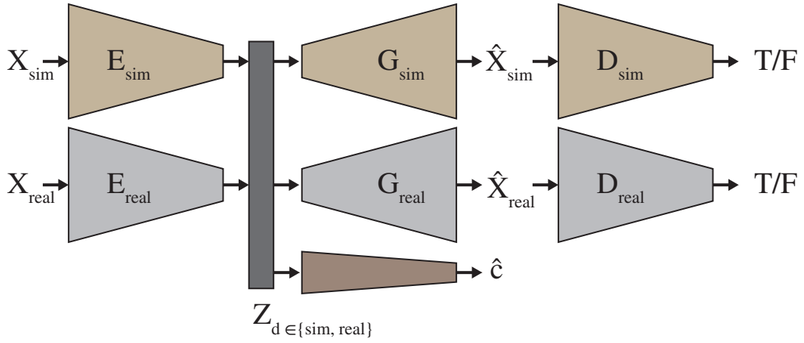
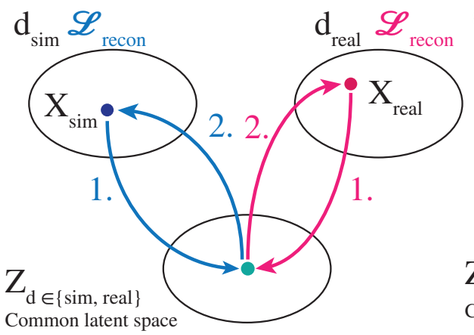


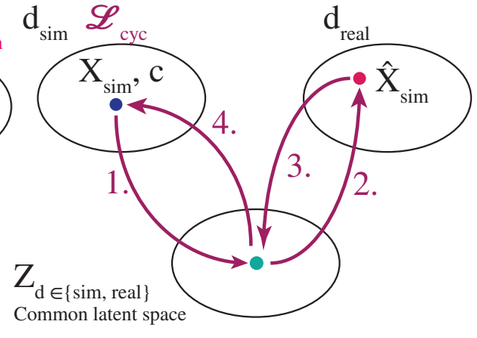
Image reconstruction loss



c

(a) Reconstruction loss $\mathcal{L}_{recon}$

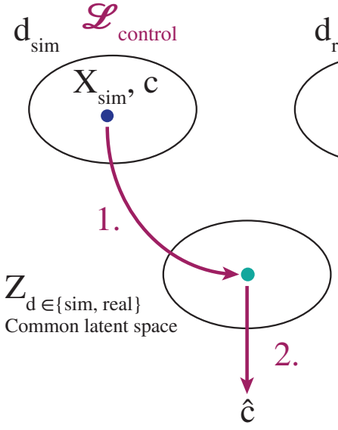
Cyclic reconstruction loss



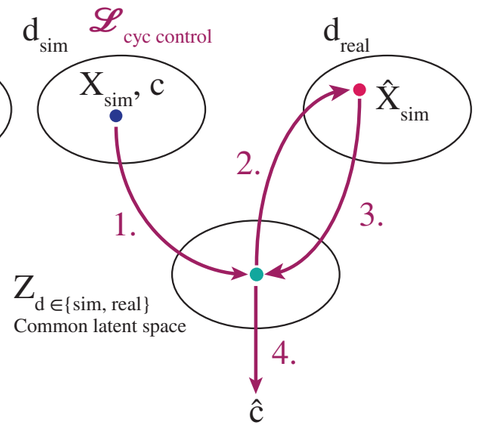
ĉ

(b) Cyclic reconstruction loss $\mathcal{L}_{cyc}$

Control loss



(c) Control loss $\mathcal{L}_{control}$



(d) Cyclic control loss $\mathcal{L}_{cyc\ control}$

Cyclic control loss

A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

Loss functions

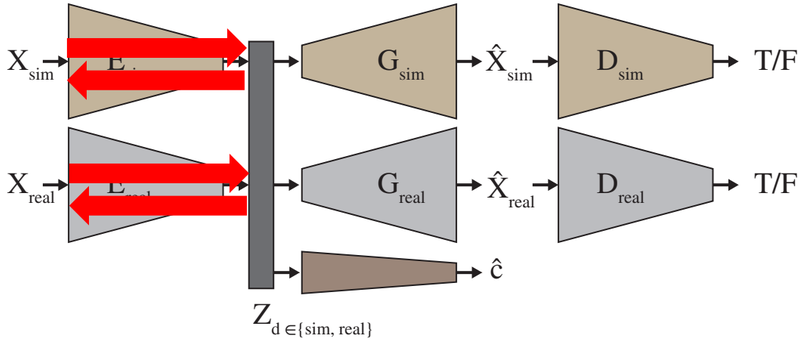
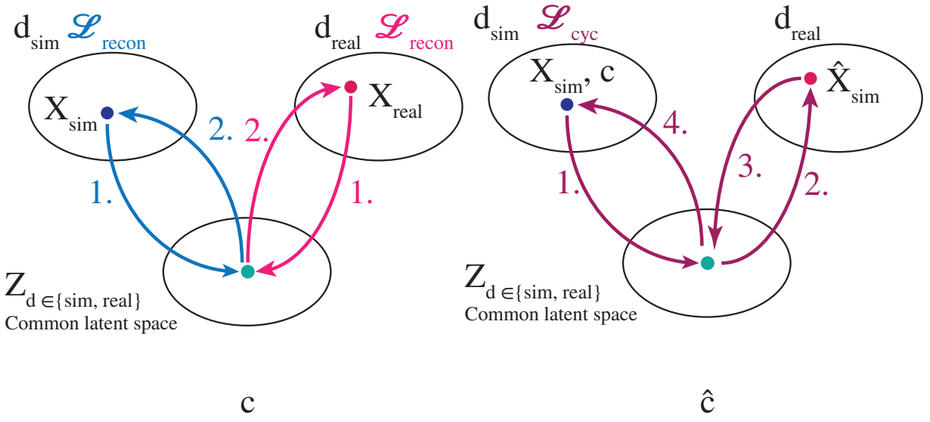


Image reconstruction loss

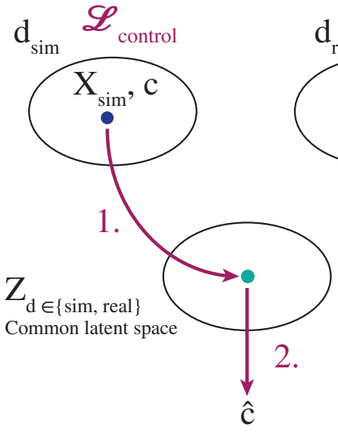


(a) Reconstruction loss $\mathcal{L}_{recon}$

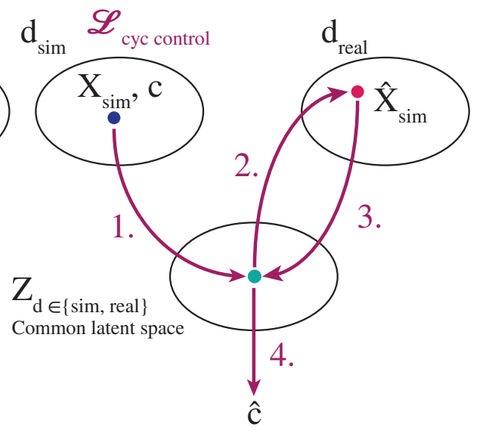
(b) Cyclic reconstruction loss $\mathcal{L}_{cyc}$

Cyclic reconstruction loss

Control loss



(c) Control loss $\mathcal{L}_{control}$



(d) Cyclic control loss $\mathcal{L}_{cyc\ control}$

Cyclic control loss

A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

Loss functions

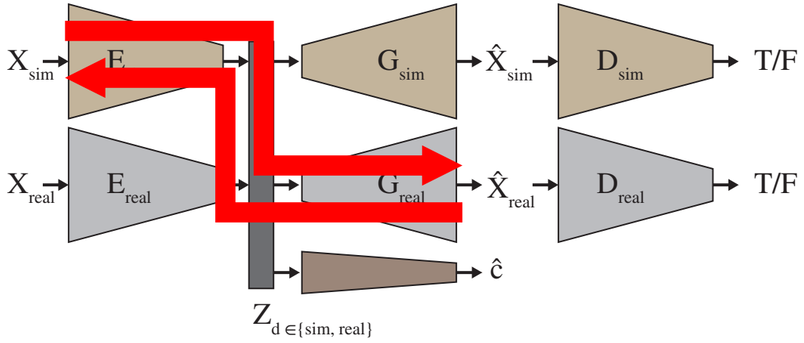
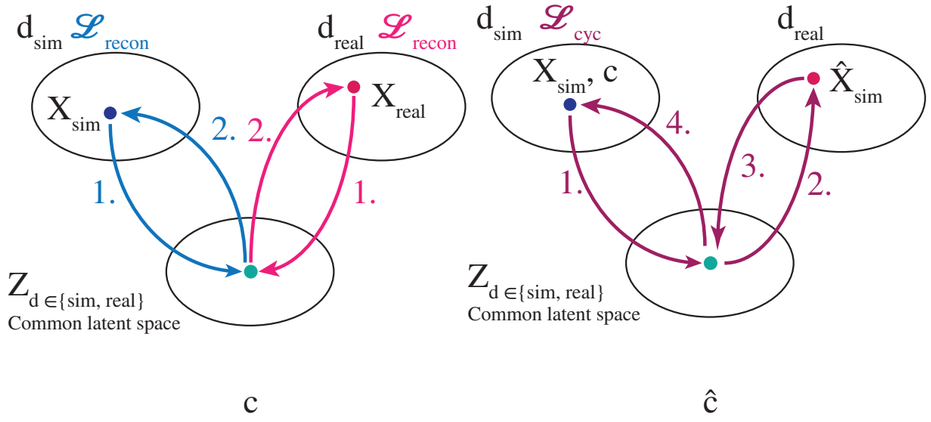


Image reconstruction loss

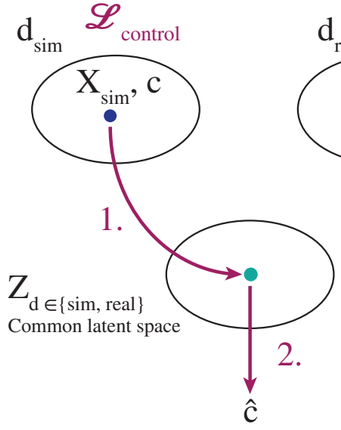


(a) Reconstruction loss $\mathcal{L}_{recon}$

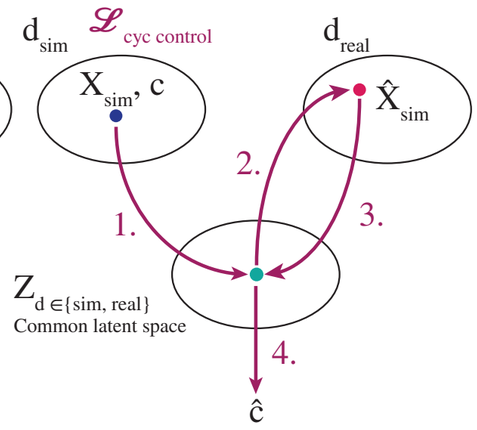
(b) Cyclic reconstruction loss $\mathcal{L}_{cyc}$

Cyclic reconstruction loss

Control loss



(c) Control loss $\mathcal{L}_{control}$



(d) Cyclic control loss $\mathcal{L}_{cyc\ control}$

Cyclic control loss

A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

Loss functions

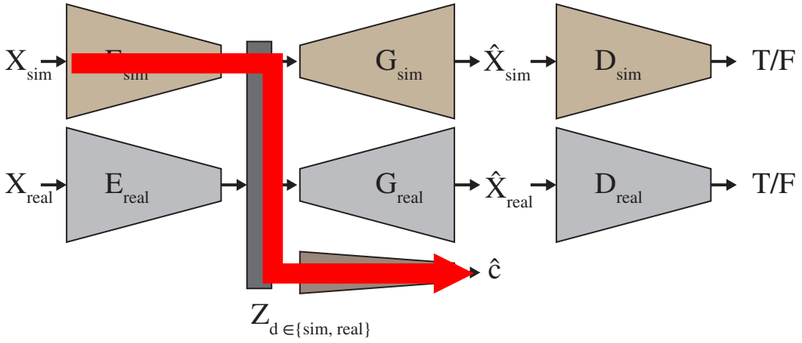
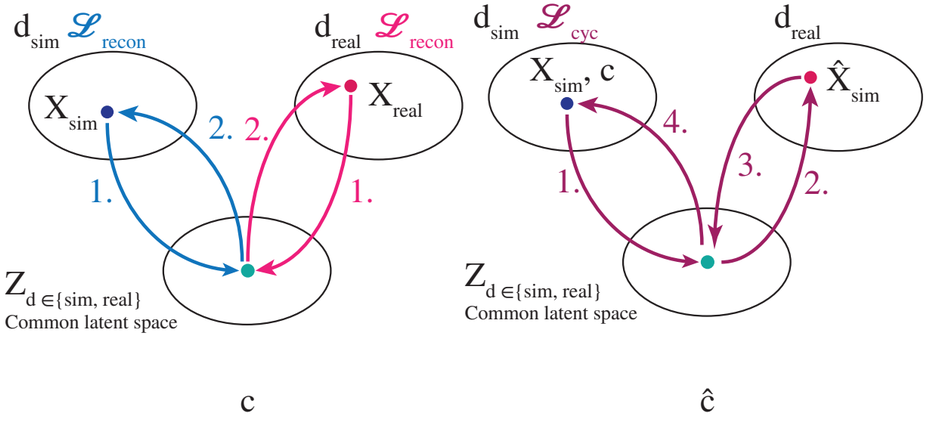


Image reconstruction loss

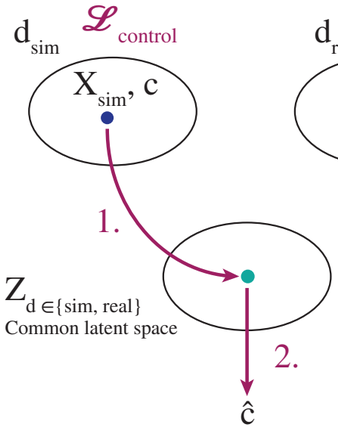


(a) Reconstruction loss $\mathcal{L}_{recon}$

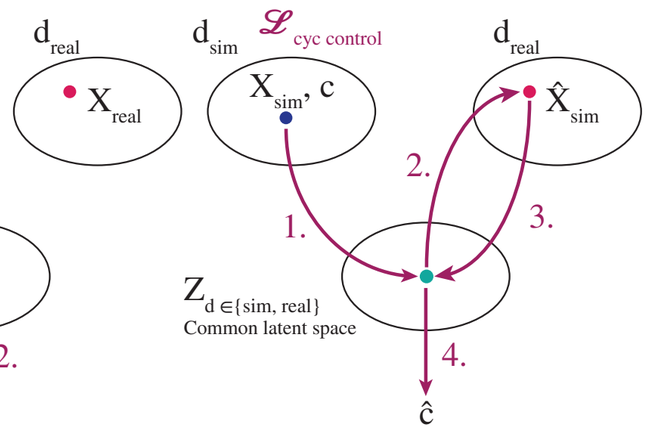
Cyclic reconstruction loss

(b) Cyclic reconstruction loss $\mathcal{L}_{cyc}$

Control loss



(c) Control loss $\mathcal{L}_{control}$



(d) Cyclic control loss $\mathcal{L}_{cyc\ control}$

Cyclic control loss

A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

Loss functions

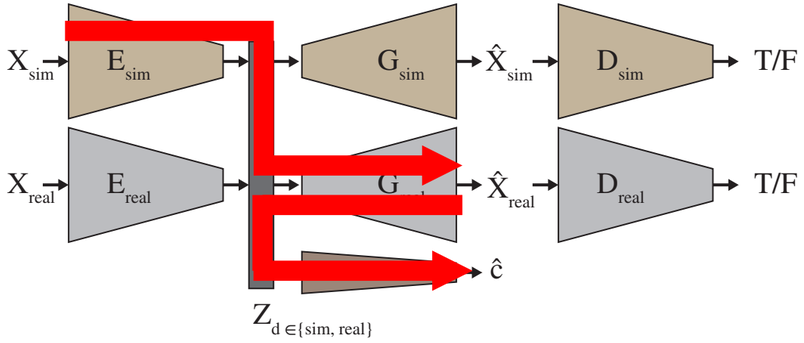
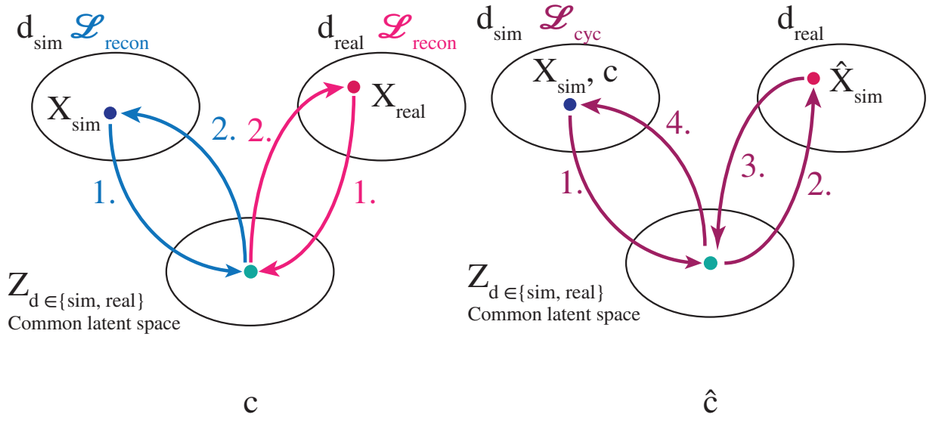


Image reconstruction loss

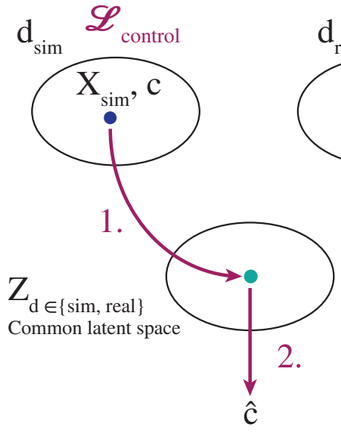


(a) Reconstruction loss $\mathcal{L}_{recon}$

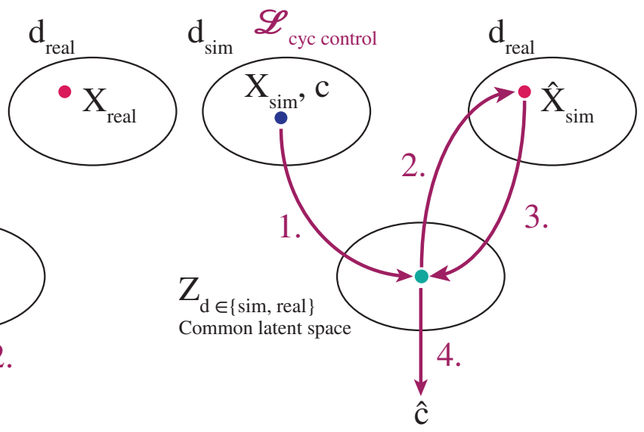
Cyclic reconstruction loss

(b) Cyclic reconstruction loss $\mathcal{L}_{cyc}$

Control loss



(c) Control loss $\mathcal{L}_{control}$

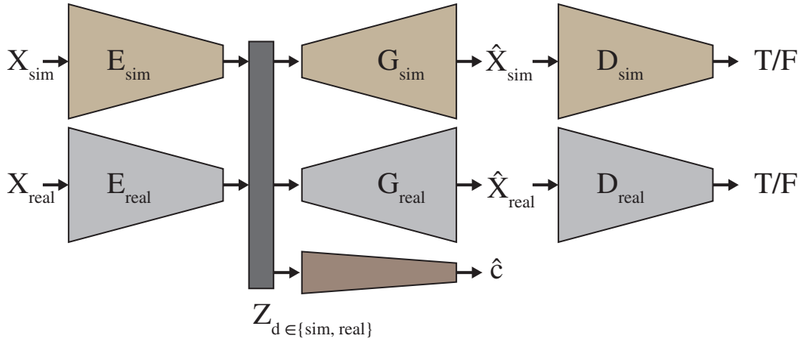


(d) Cyclic control loss $\mathcal{L}_{cyc\ control}$

Cyclic control loss

A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

Loss functions



- Adversarial loss $\mathcal{L}_{LSGAN}(D) = \mathbb{E}_{X \sim p_{data}}[(D(X)-1)^2] + \mathbb{E}_{Z \sim p_Z(Z)}[(D(G(Z))-0)^2]$

  $\mathcal{L}_{LSGAN}(G) = \mathbb{E}_{Z \sim p_Z(Z)}[(D(G(Z))-1)^2]$

- Perceptual loss $\mathcal{L}_{perceptual}$

  Difference between the features from the last convolutional layer of a pretrained VGG16 model for a given input image and its translated counterpart.

- Latent reconstruction loss $\mathcal{L}_{Zrecon}$

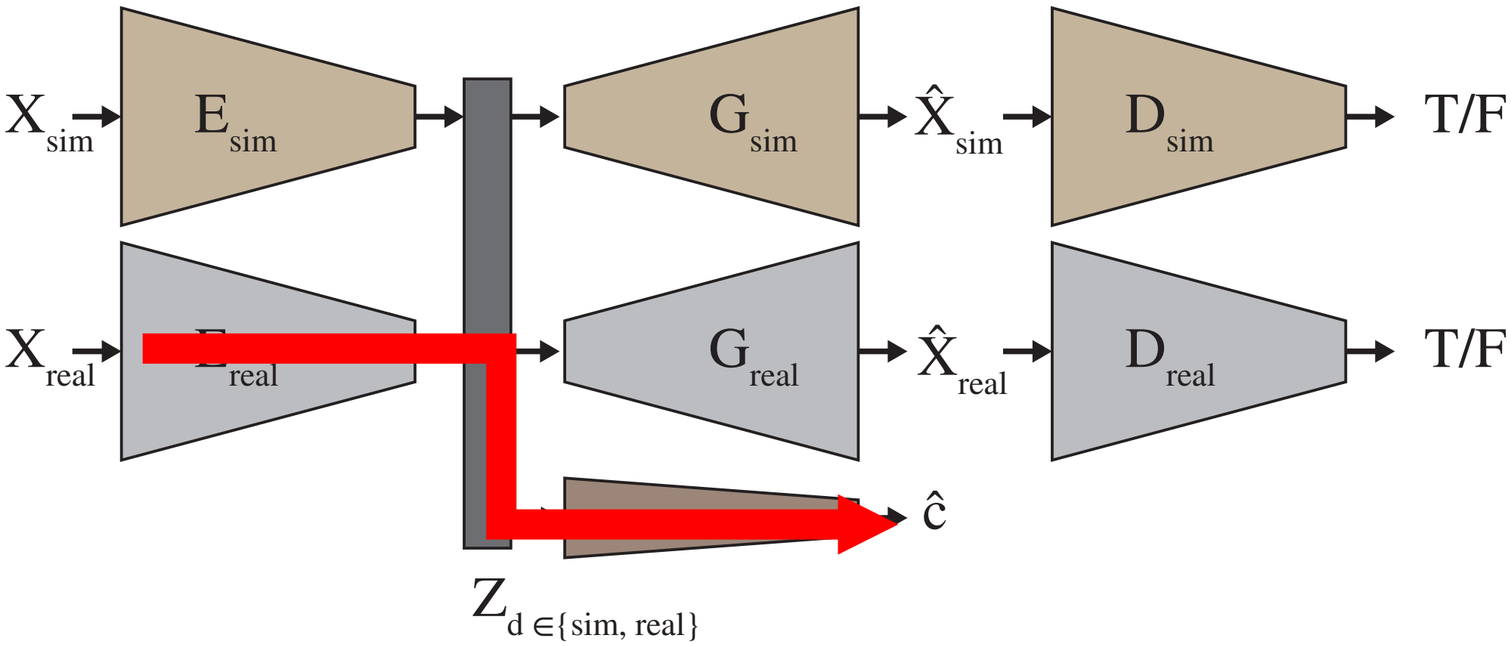
  Difference between the latent representation of an image and the reconstructed latent representation after it was decoded to the other domain and encoded once more.

- Overall loss

$$\mathcal{L}_{tot} = \lambda_0 \mathcal{L}_{recon} + \lambda_1 \mathcal{L}_{cyc} + \lambda_2 \mathcal{L}_{control} + \lambda_3 \mathcal{L}_{cyc\ control} + \lambda_4 \mathcal{L}_{LSGAN}(G) + \lambda_5 \mathcal{L}_{perceptual} + \lambda_6 \mathcal{L}_{Zrecon}.$$

A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

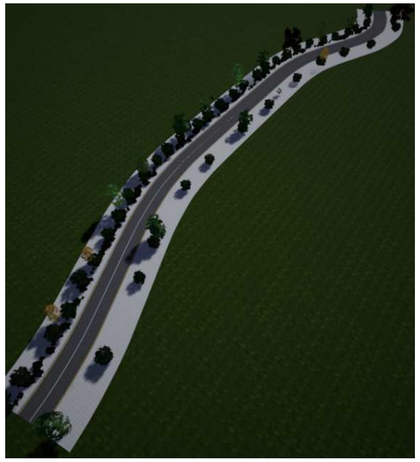
Inference/online phase?



A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

Dataset

(a) The 250m real-world rural driving route, coloured in blue.



(b) A procedurally generated curvy driving route in simulation.

(c) Real-world rural road.

(d) Simulated road.

Fig. 4: Rural setting: aerial views of the real-world (a) and simulated driving routes (b), along with example images from each domain showing the perspective from the vehicle (c, d).

(a) Simulated urban world.

(b) The real-world autonomous vehicle used for all experiments.

(c) An example of the real-world urban roads used for closed loop testing in Cambridge, UK.

Fig. 6: Urban setting: illustrations of the simulated and real-world domains for urban road scenes. Despite quite a large appearance change from the cartoon-like simulated world to the real-world, our model is able to successfully transfer the learned control policy to drive in in the real world, with no real world labels.

TABLE I: Datasets used for driving policy domain adaptation. Each simulation frame has associated expert control labels.

	Training Frames	Test Frames
Simulation	60014	17741
Real	57916	19141

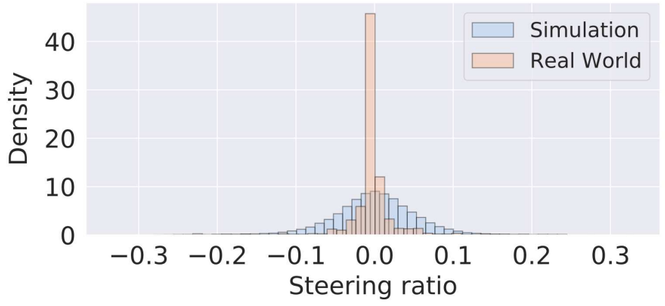


Fig. 5: The steering data distribution (in the range ± 1.0) differs dramatically between the simulation and real world, where simulation has much greater variance. The real-world data has shorter tails and is heavily concentrated around the zero position, due to the profile of the rural driving route.

A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

Results

TABLE II: Open-loop control metrics on the simulation and real test datasets from Table I.

	Simulation		Real	
	MAE	Bal-MAE	MAE	Bal-MAE
Drive-Straight	0.043	0.087	**0.019**	0.093
Simple Transfer	0.05	0.055	0.265	0.272
Real-to-Sim Translation	-	-	0.261	0.234
Sim-to-Real Translation	-	-	0.059	**0.045**
Latent Feature ADA	0.040	0.047	0.032	0.071
Ours	**0.017**	**0.018**	0.081	0.087

TABLE III: On-vehicle performance, driving 3km on the rural driving route in 4a. For policies unable to drive a lap with ≤ 1 intervention, we terminated after one 250m lap (†).

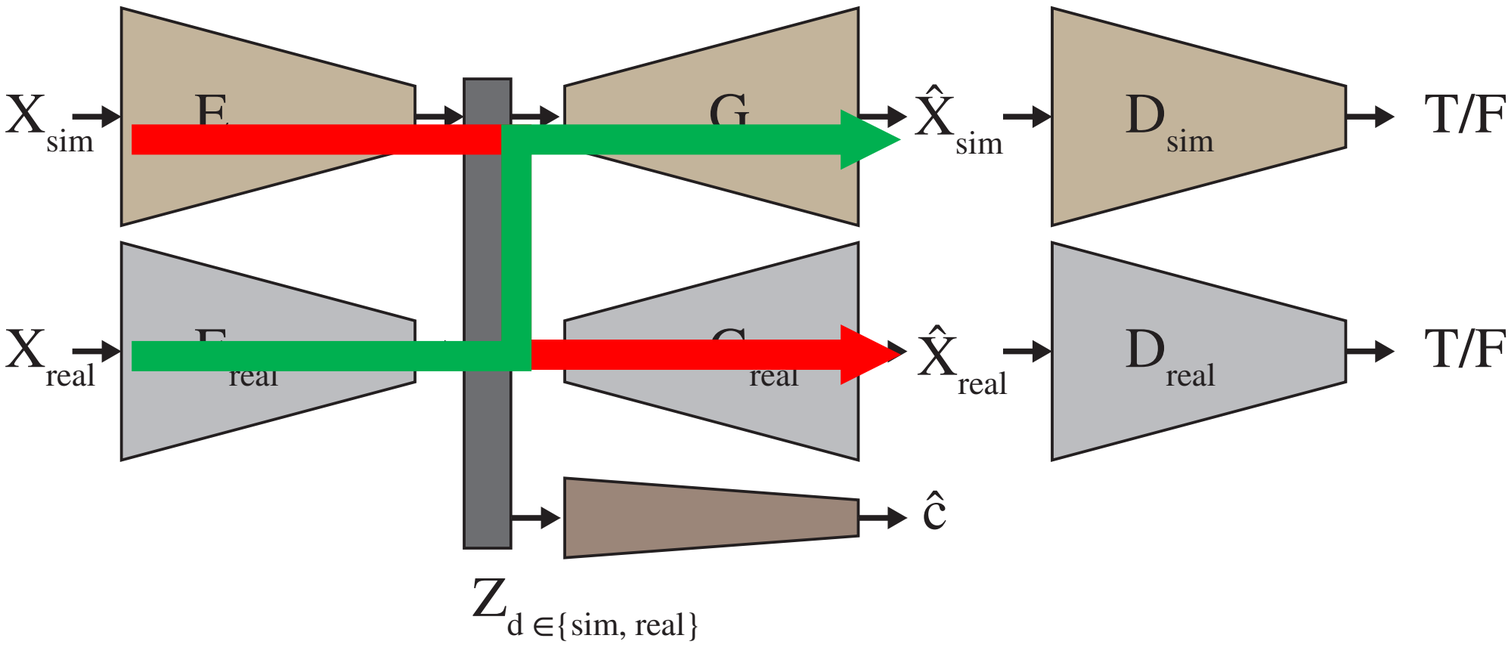
	Mean distance / intervention (metres)
Drive-straight	$23^{\dagger}$
Simple Transfer	$9^{\dagger}$
Real-to-Sim Translation	$10^{\dagger}$
Sim-to-Real Translation	$28^{\dagger}$
Latent Feature ADA	$15^{\dagger}$
Ours	No intervention over 3km

TABLE IV: Ablation by removing single terms in eqn. 5. Values expressed as the average error multiplier across three runs for the open-loop metrics when transferred to the real domain test dataset. For the Control term the gradients applied to the image translator for this loss are removed (‡).

	MAE	Bal-MAE
Recon. L1	1.602	1.259
Recon. Cyclic L1	2.980	2.395
Z Recon.	1.766	1.270
Perceptual	1.693	1.435
Cyc Control	0.923	1.068
GAN	1.893	1.507
Control-grad(‡)	1.516	1.433

A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

Domain translation (inference phase)



A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

Results

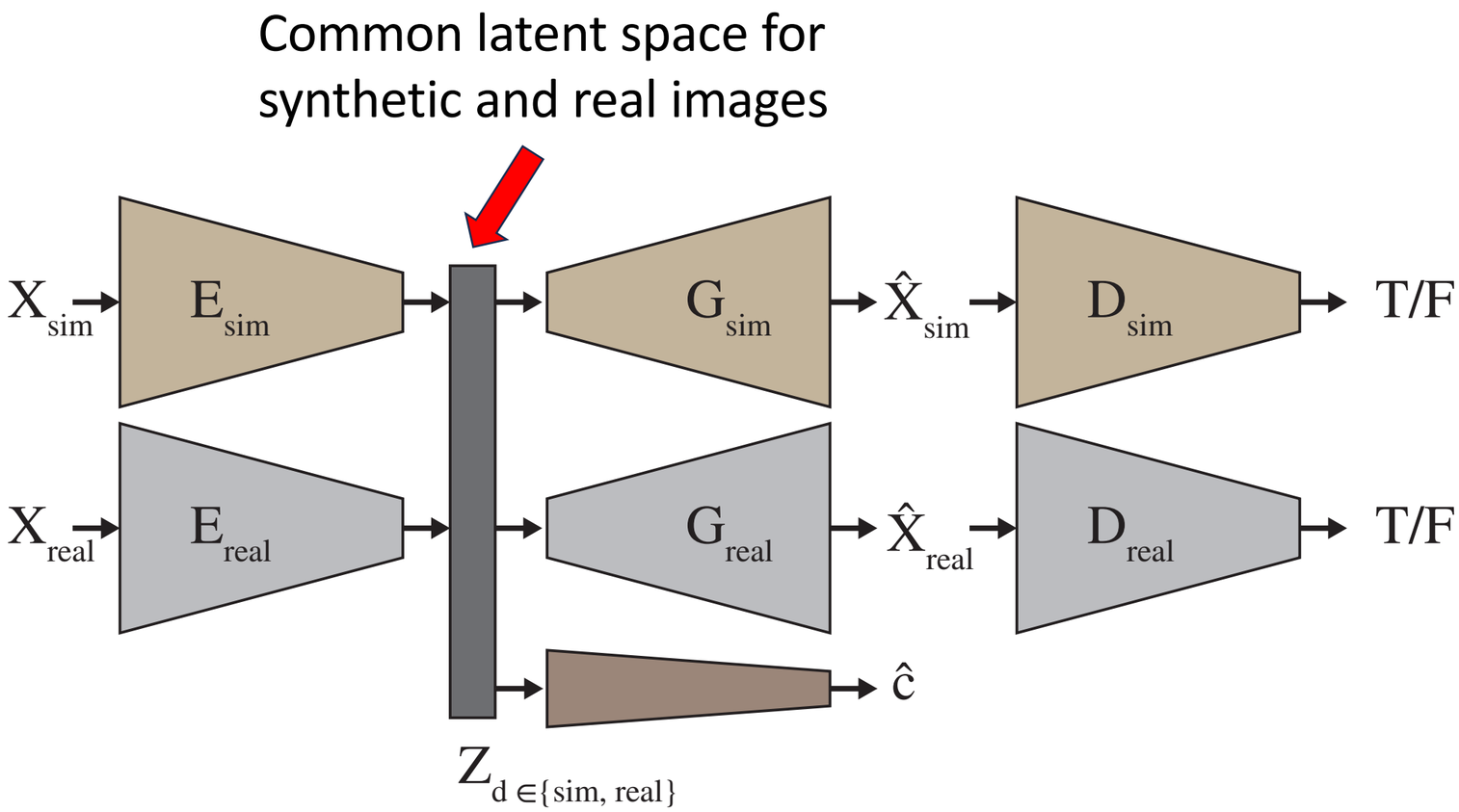
(a) Rural environment.

(b) Urban environment.

Fig. 7: Qualitative results of the image translation model for (a) rural and (b) urban environments. In each example, the top four rows show images originating in our simulated environment, and the bottom half showing translation from real world images. The columns left to right are as follows: original, image reconstructed from latent back to original domain, translated to other domain, and taking the translated image and translating it back to the original domain.

A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

What acts as the bridge between sim and real?



A. Bewley, J. Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, Alex Kendall. Learning to Drive from Simulation without Real World Labels. ICRA 2018.

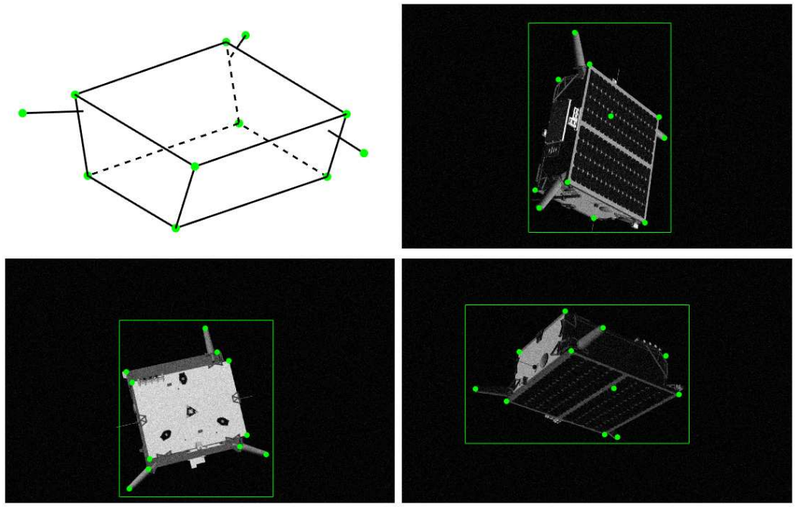
Outline

- Domain and sim2real adaptation
- Domain randomisation I
- Domain randomisation II
- Recap on generative adversarial networks (GAN)
- Domain translation
- Modularisation and abstraction
- Latent space alignment
- Adversarial training

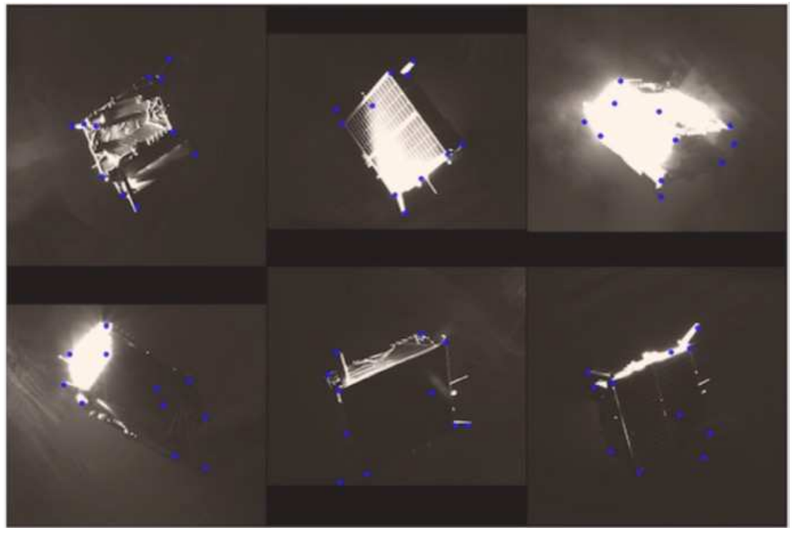
Adversarial training

- Problem setting:
  - Train an object pose estimation pipeline using <u>labelled</u> synthetic images and <u>unlabelled</u> real images.
    - ➔Unsupervised sim2real DA.
    - ➔Synthetic and real images are <u>unpaired</u>.
  - Real images are under unseen before (harsh) lighting conditions.
- Idea:
  - Train a latent representation that is common for synthetic and real images.
  - Multiscale representation to reduce the effects of lighting differences.

Synthetic

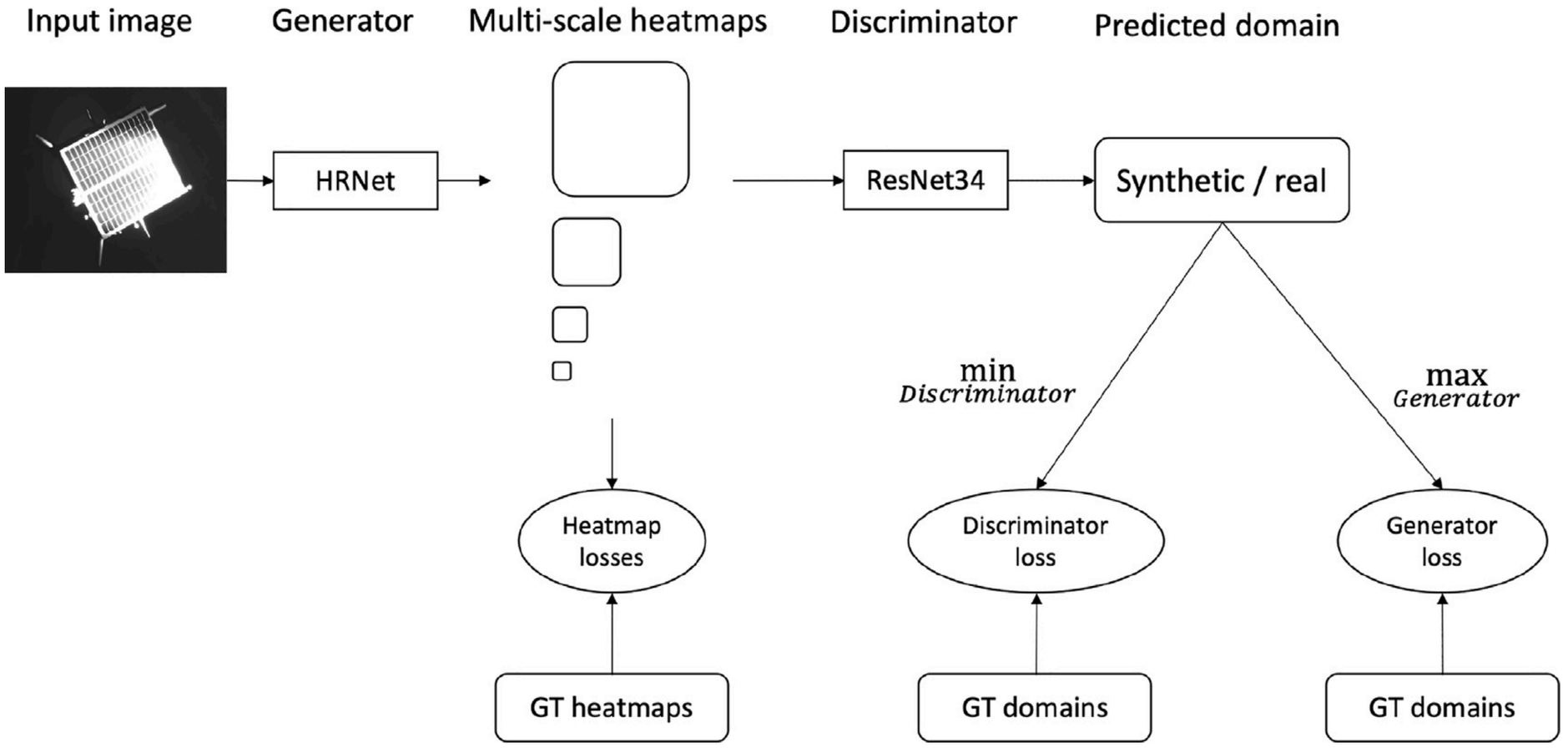


Real



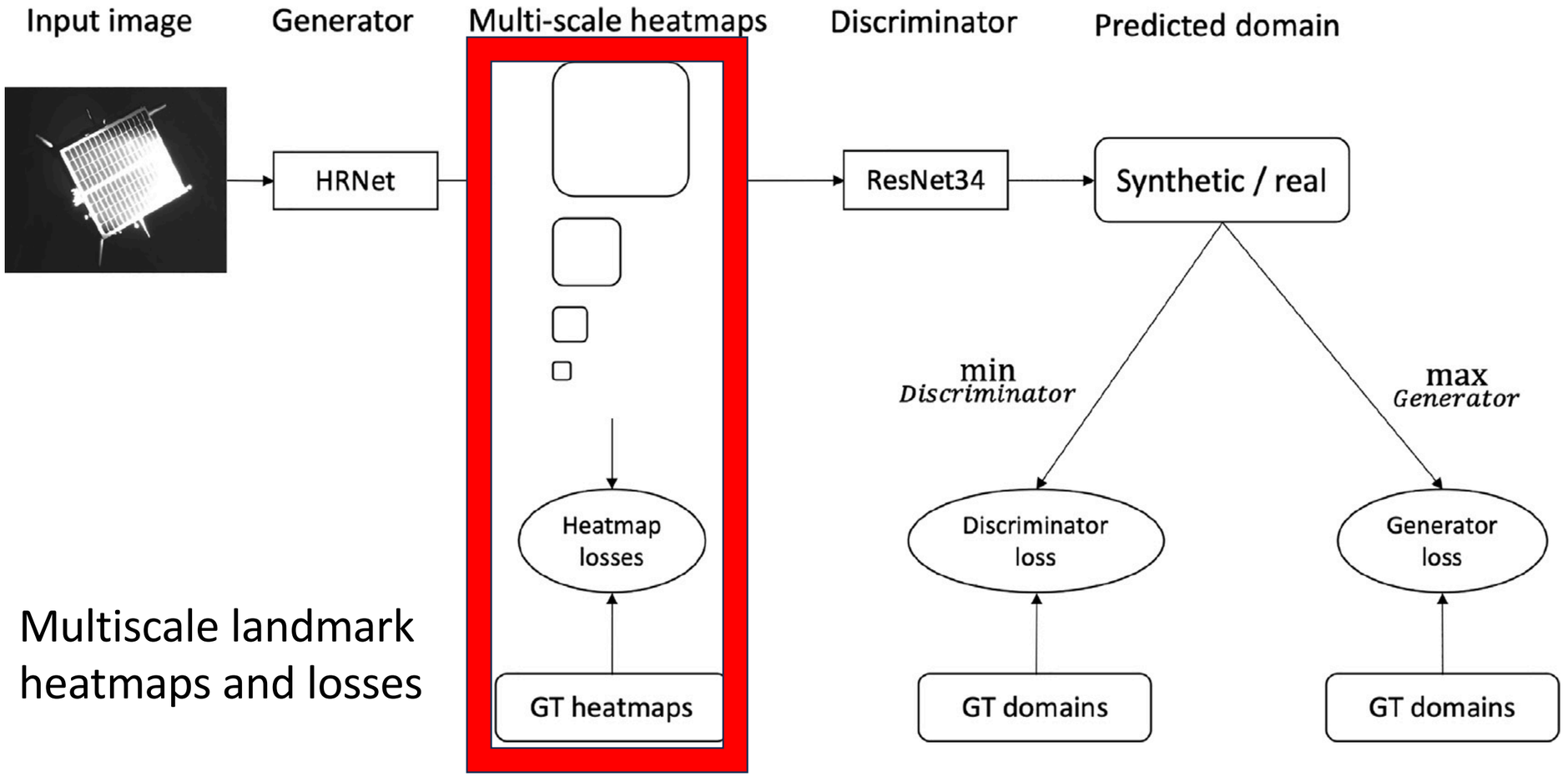
Tae Ha Park, Marcus Märtens, Mohsi Jawaid, Zi Wang, Bo Chen, Tat-Jun Chin, Dario Izzo, Simone D'Amico. Satellite Pose Estimation Competition 2021: Results and Analyses. Acta Astronautica. Volume 204, 2023, Pages 640-665.

Method



Tae Ha Park, Marcus Märtens, Mohsi Jawaid, Zi Wang, Bo Chen, Tat-Jun Chin, Dario Izzo, Simone D'Amico. Satellite Pose Estimation Competition 2021: Results and Analyses. Acta Astronautica. Volume 204, 2023, Pages 640-665.

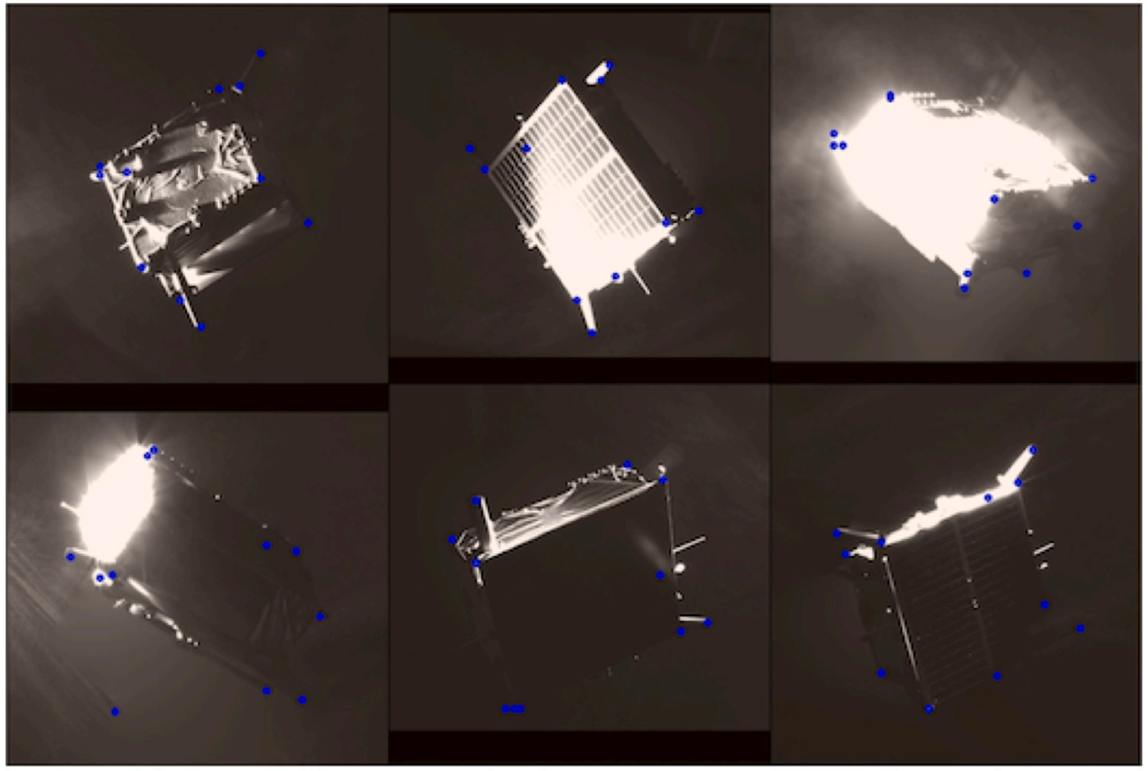
Method



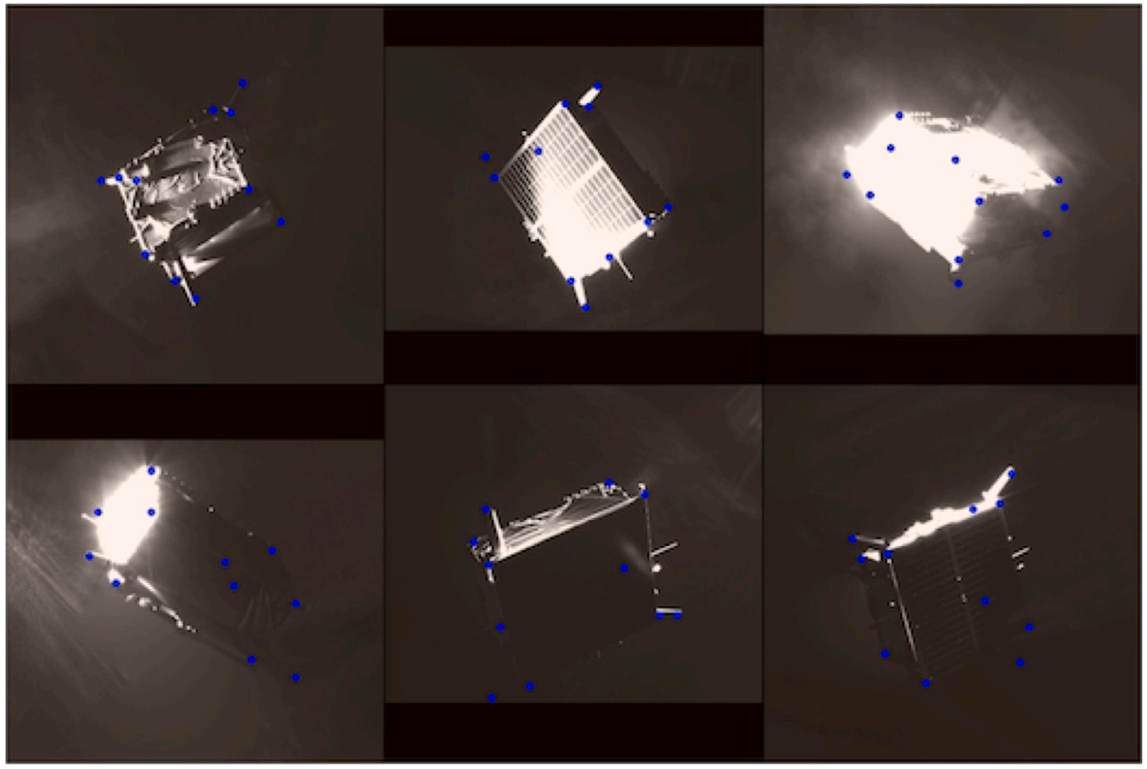
Multiscale landmark heatmaps and losses

Tae Ha Park, Marcus Märtens, Mohsi Jawaid, Zi Wang, Bo Chen, Tat-Jun Chin, Dario Izzo, Simone D'Amico. Satellite Pose Estimation Competition 2021: Results and Analyses. Acta Astronautica. Volume 204, 2023, Pages 640-665.

Ablation test – multiscale heatmap



(a) Without multi-scale heatmap supervision.

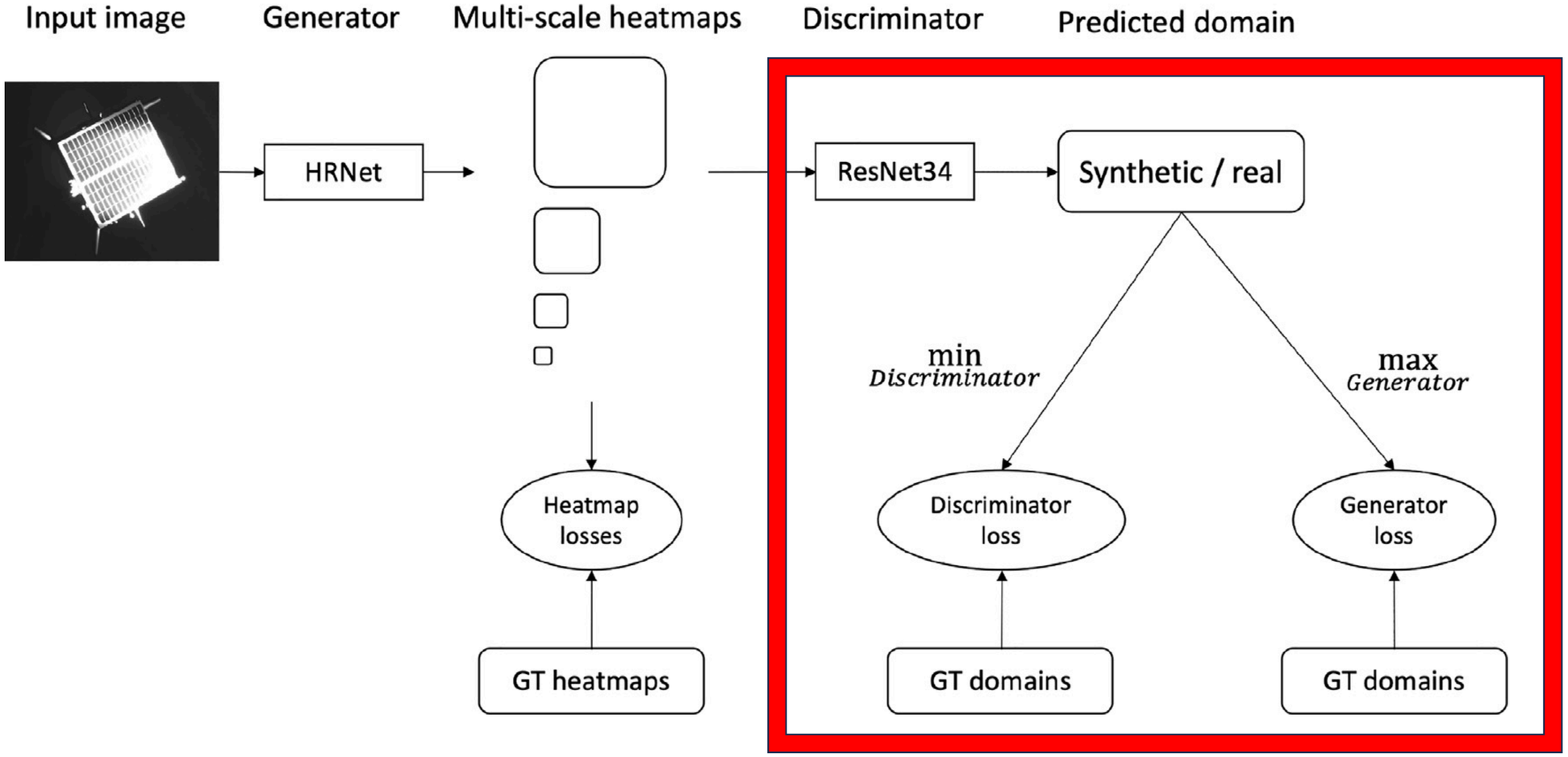


(b) With multi-scale heatmap supervision.

Tae Ha Park, Marcus Märtens, Mohsi Jawaid, Zi Wang, Bo Chen, Tat-Jun Chin, Dario Izzo, Simone D'Amico. Satellite Pose Estimation Competition 2021: Results and Analyses. Acta Astronautica. Volume 204, 2023, Pages 640-665.

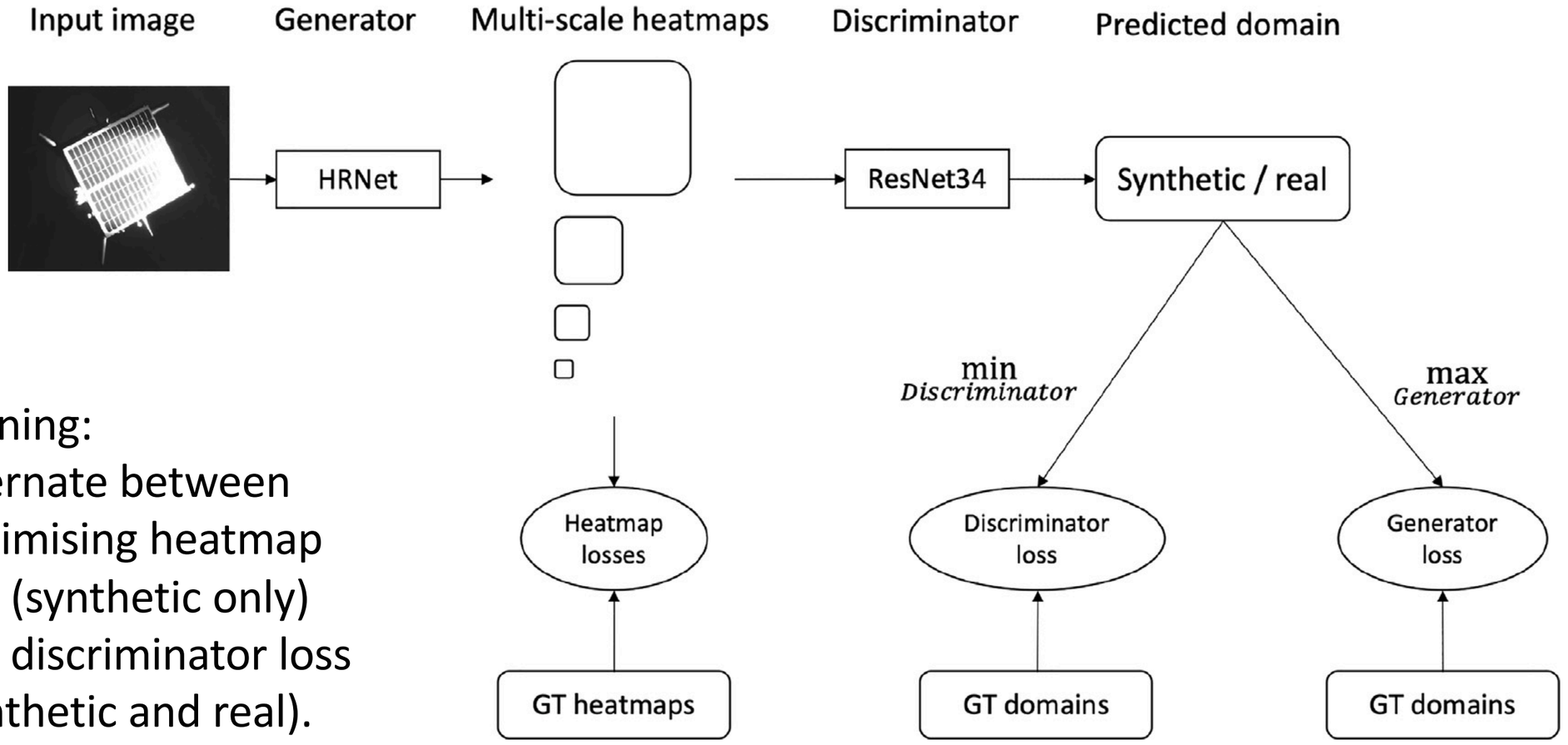
Method

Try to discriminate real and synthetic images based on their landmark heatmaps



Tae Ha Park, Marcus Märtens, Mohsi Jawaid, Zi Wang, Bo Chen, Tat-Jun Chin, Dario Izzo, Simone D'Amico. Satellite Pose Estimation Competition 2021: Results and Analyses. Acta Astronautica. Volume 204, 2023, Pages 640-665.

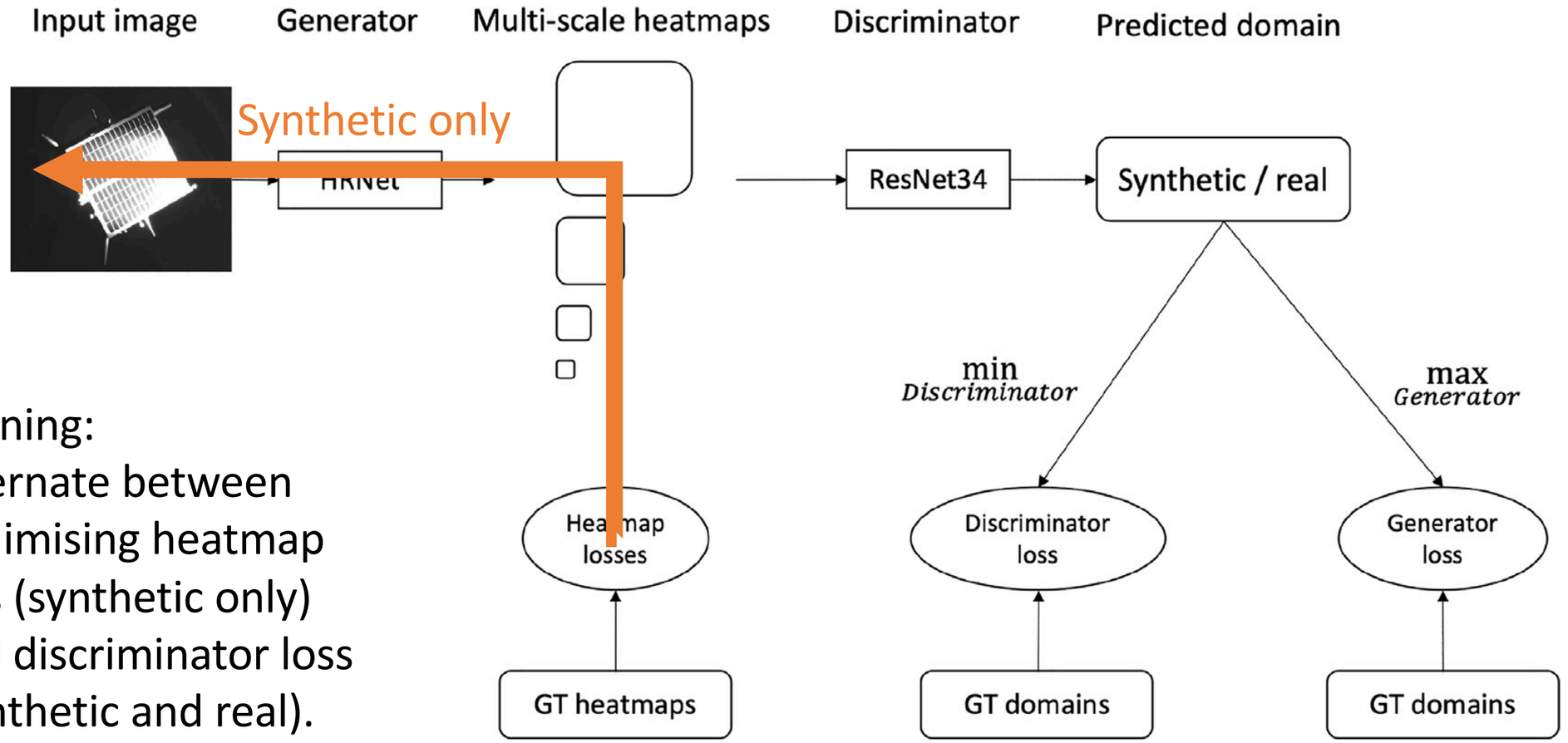
Method



Training:
Alternate between minimising heatmap loss (synthetic only) and discriminator loss (synthetic and real).

Tae Ha Park, Marcus Märtens, Mohsi Jawaid, Zi Wang, Bo Chen, Tat-Jun Chin, Dario Izzo, Simone D'Amico. Satellite Pose Estimation Competition 2021: Results and Analyses. Acta Astronautica. Volume 204, 2023, Pages 640-665.

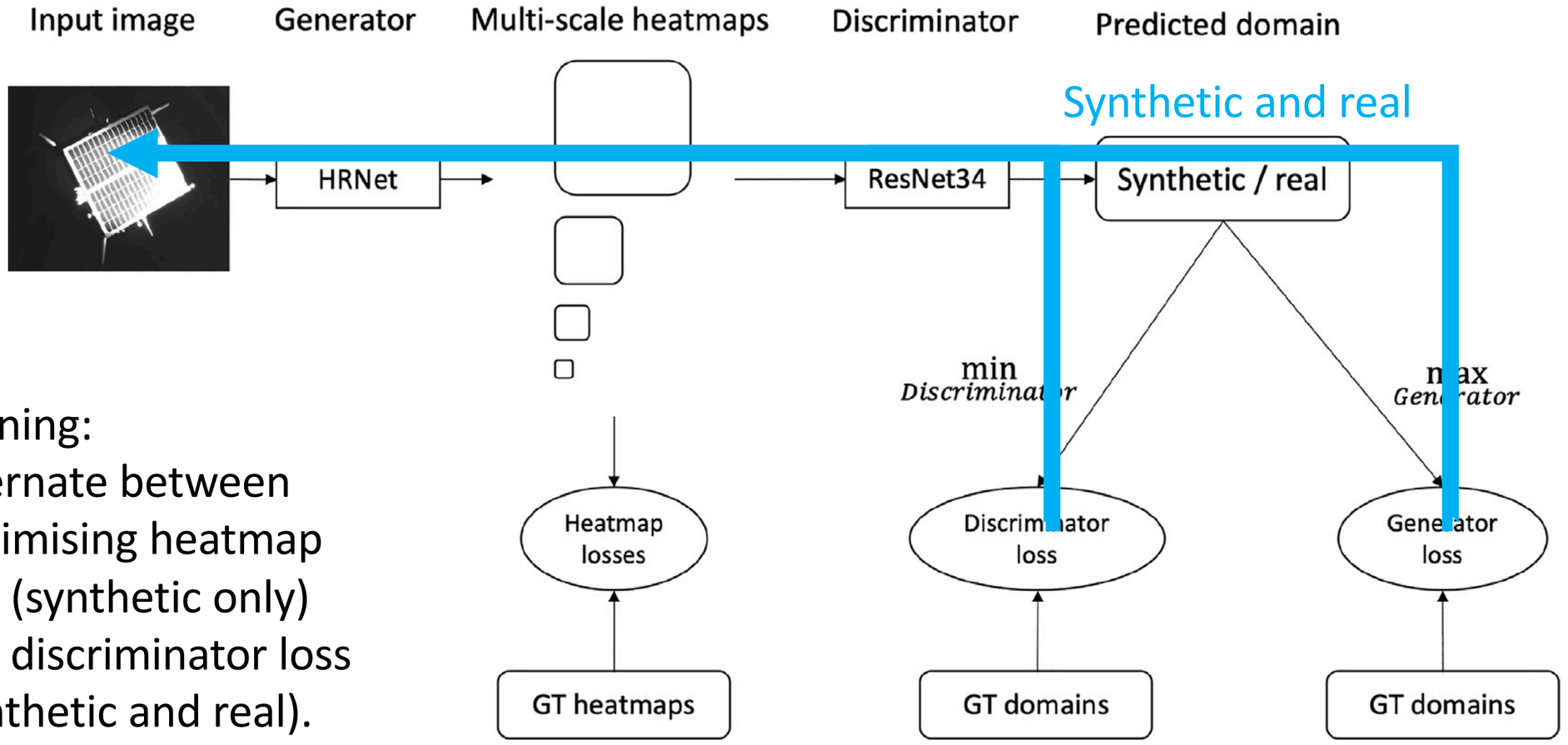
Method



Training:
Alternate between minimising heatmap loss (synthetic only) and discriminator loss (synthetic and real).

Tae Ha Park, Marcus Märtens, Mohsi Jawaid, Zi Wang, Bo Chen, Tat-Jun Chin, Dario Izzo, Simone D'Amico. Satellite Pose Estimation Competition 2021: Results and Analyses. Acta Astronautica. Volume 204, 2023, Pages 640-665.

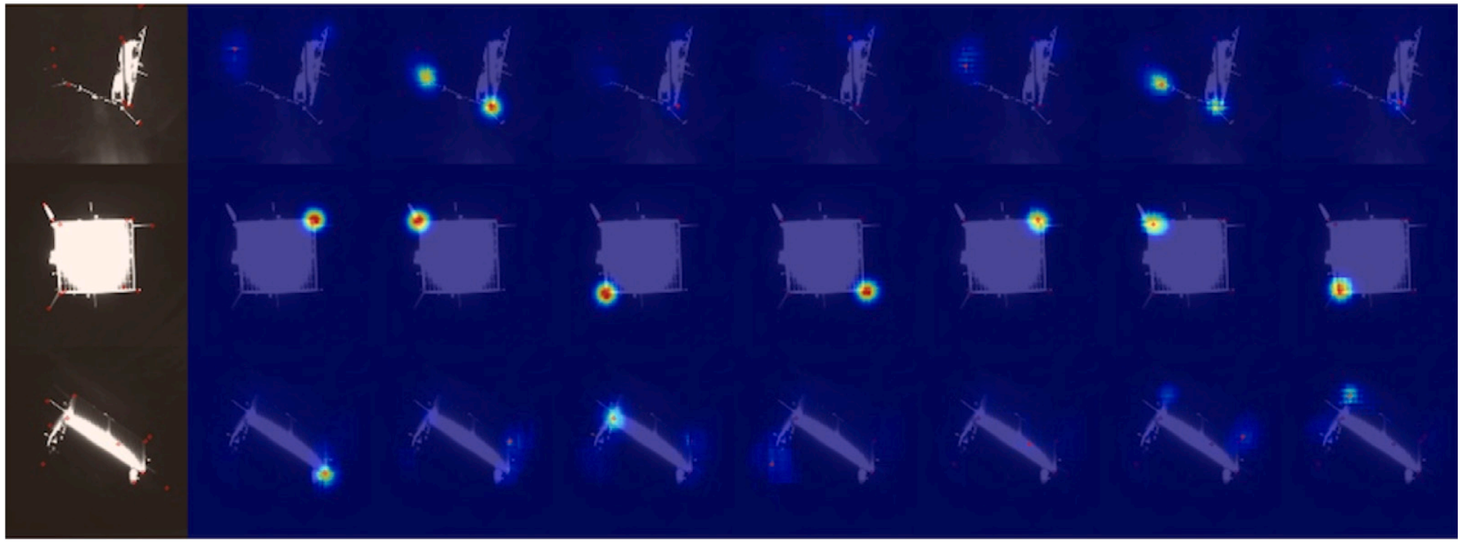
Method



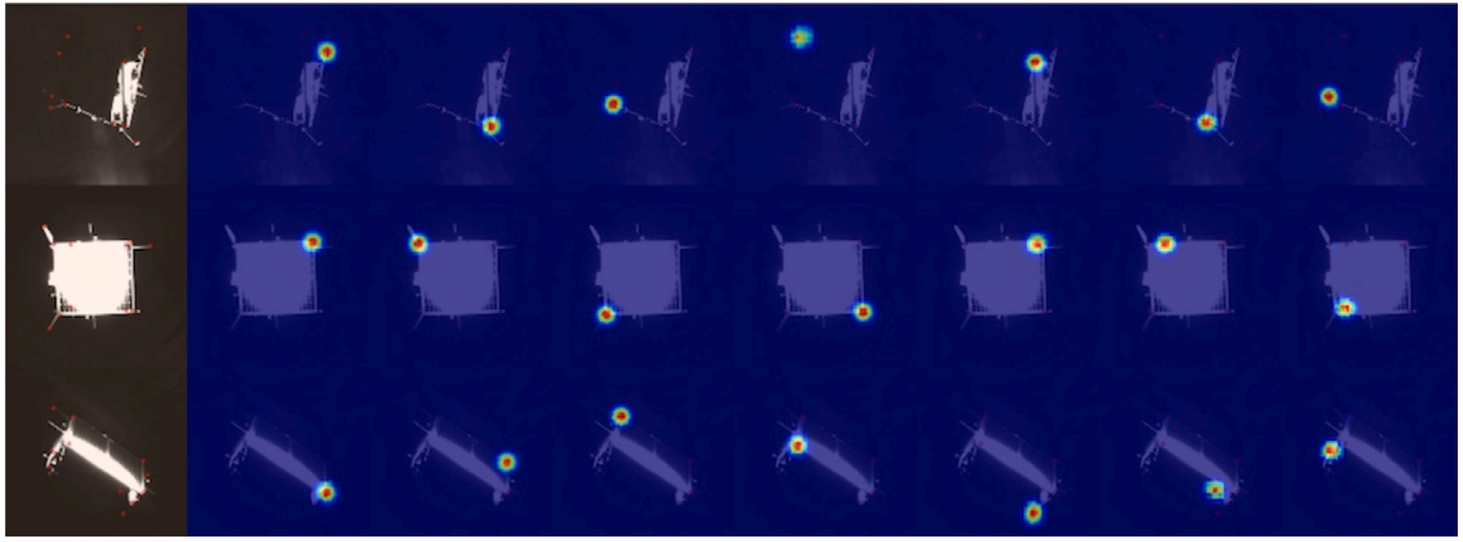
Training:
Alternate between minimising heatmap loss (synthetic only) and discriminator loss (synthetic and real).

Tae Ha Park, Marcus Märtens, Mohsi Jawaid, Zi Wang, Bo Chen, Tat-Jun Chin, Dario Izzo, Simone D'Amico. Satellite Pose Estimation Competition 2021: Results and Analyses. Acta Astronautica. Volume 204, 2023, Pages 640-665.

Ablation test - adversarial training



(a) Without adversarial training.

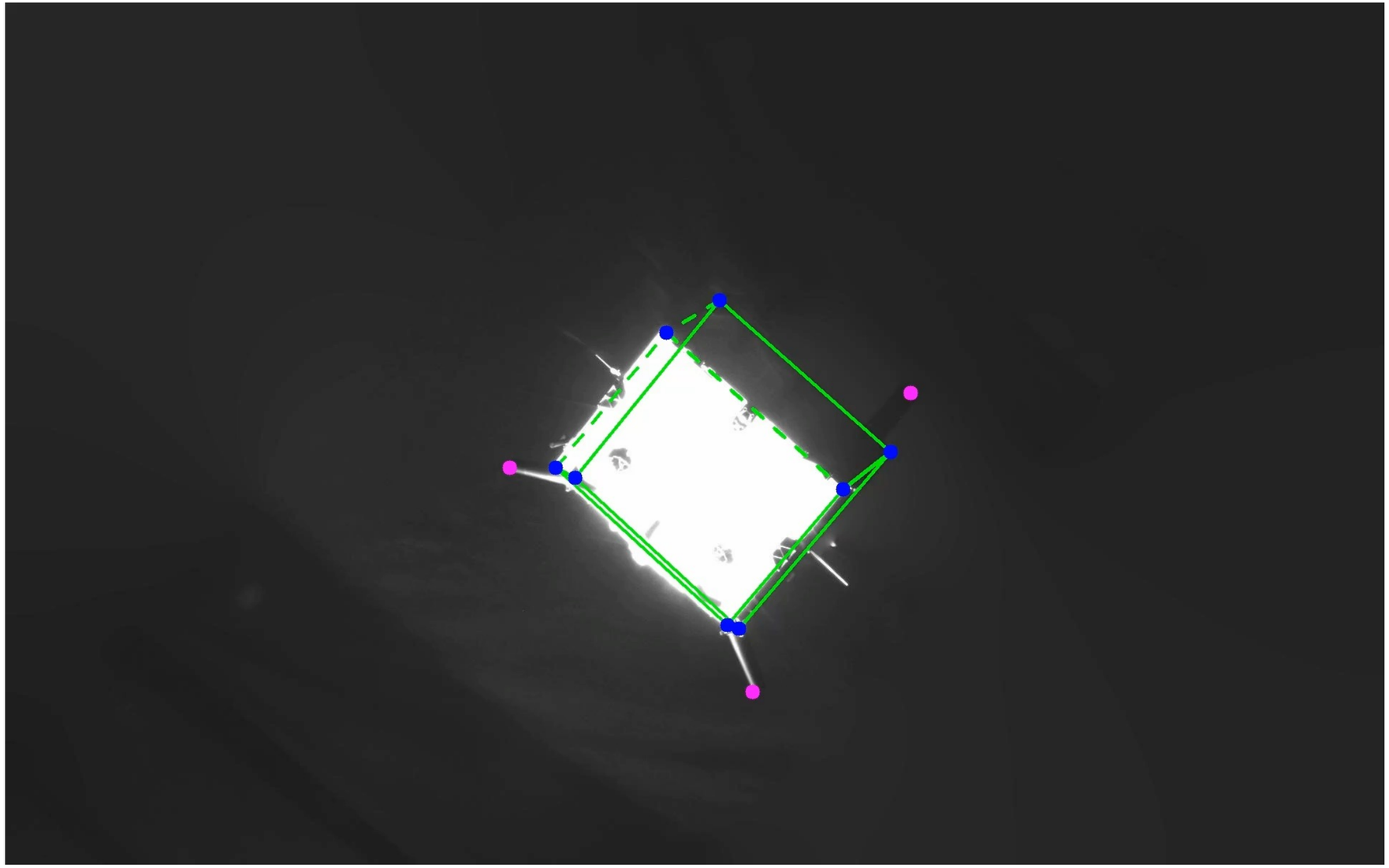


(b) With adversarial training.

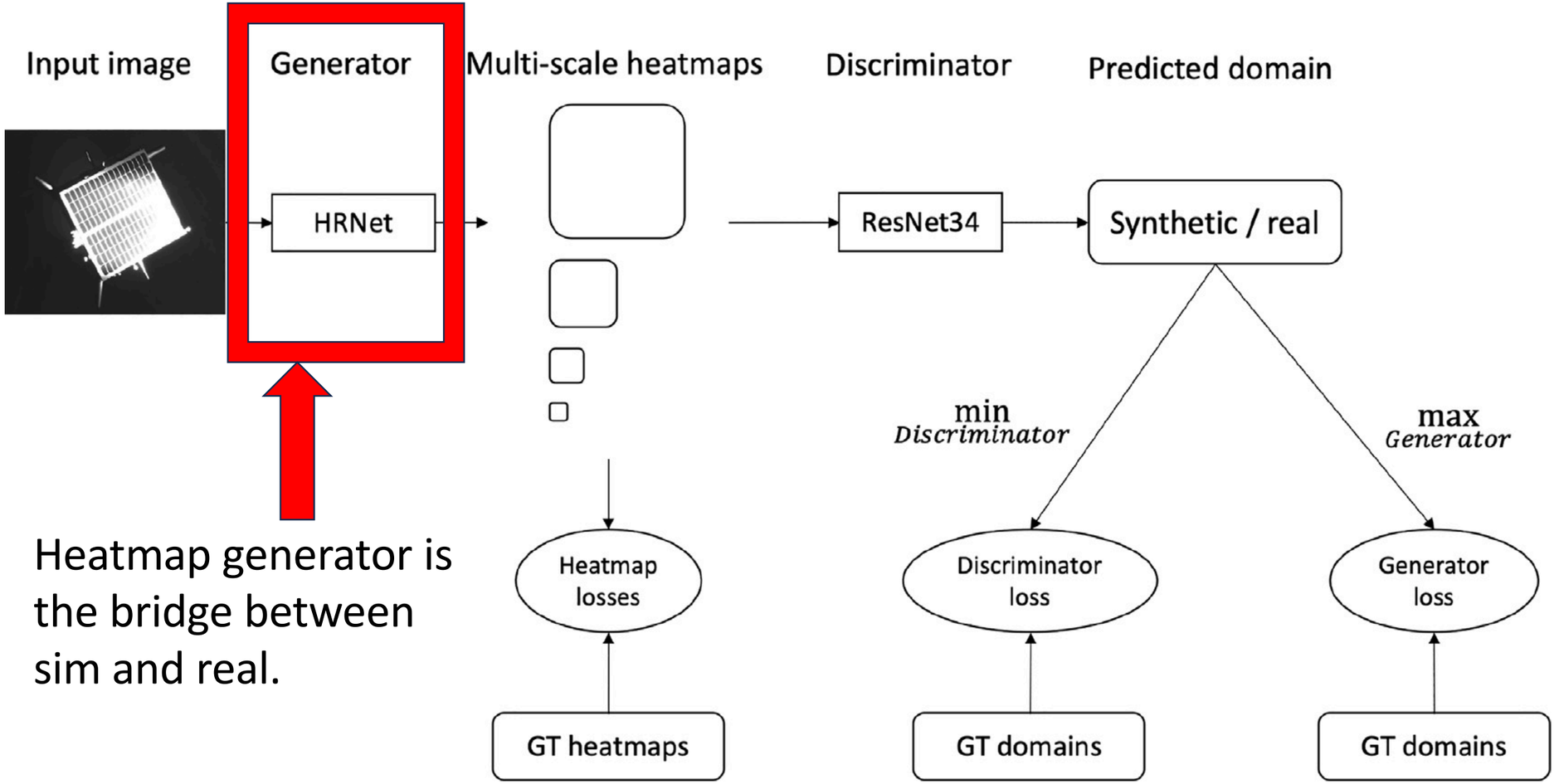
Tae Ha Park, Marcus Märtens, Mohsi Jawaid, Zi Wang, Bo Chen, Tat-Jun Chin, Dario Izzo, Simone D'Amico. Satellite Pose Estimation Competition 2021: Results and Analyses. Acta Astronautica. Volume 204, 2023, Pages 640-665.

Sim2real adaptation results

Source: https://www.youtube.com/@mohsi.j



What acts as the bridge between sim and real?



Heatmap generator is the bridge between sim and real.

Tae Ha Park, Marcus Märtens, Mohsi Jawaid, Zi Wang, Bo Chen, Tat-Jun Chin, Dario Izzo, Simone D'Amico. Satellite Pose Estimation Competition 2021: Results and Analyses. Acta Astronautica. Volume 204, 2023, Pages 640-665.

make history.

THE UNIVERSITY of ADELAIDE