



# S31649 - Bridging Sim2Real Gap: Simulation Tuning for Training Deep Learning Robotic Perception Models

Peter Dykas, NVIDIA Solutions Architect  
April 12, 2021

# Agenda

- Synthetic Data Basics
  - What is the sim2real gap and how do we overcome it?
- Tuning simulators techniques and pitfalls
  - How do we setup our simulator to generalize to the real world?
- Advanced Topics and the Future
  - How do we build end to end pipelines for training perception networks?

The background of the slide is a dark, almost black, field. It is populated with numerous thin, light green lines that crisscross the frame in various directions. Interspersed among these lines are several small, bright green circular dots or nodes. Some of these dots are slightly larger and more prominent than others. The overall effect is a complex, web-like pattern that suggests a network or a dynamic system. In the bottom right corner, the text "The Basics" is written in a clean, white, sans-serif font.

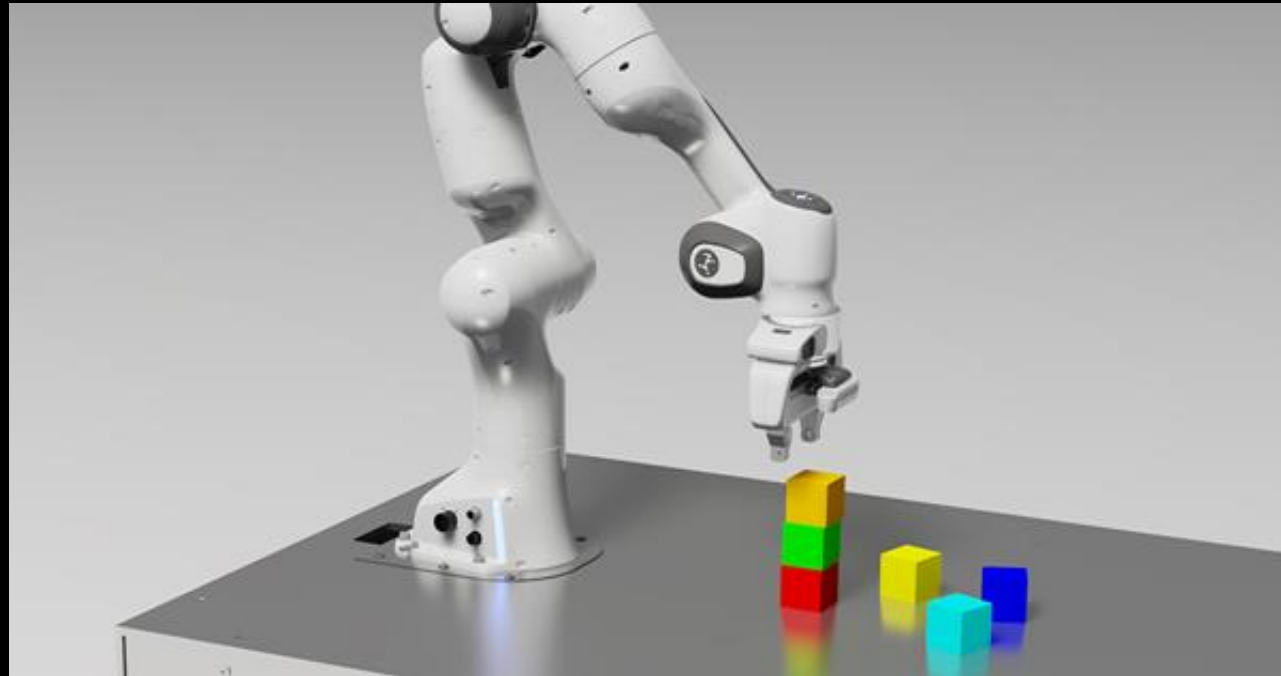
# The Basics

# Robotic Perception

## DNN Explosion

DNNs are quickly becoming the standard for SOTA perception systems

With this comes an exponential hunger for labeled data which can be difficult to collect in many robotic applications



# Robotic Perception

## Data Collection

- Safety Concerns
- Actuation is challenging
- Expensive and Time consuming
- Unknown sensor set at beginning of development
- Corner cases generation



# Robotic Perception

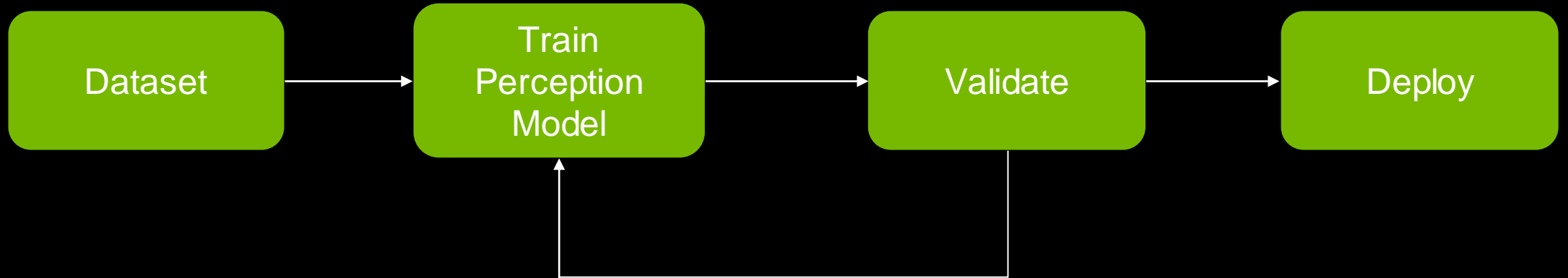
## Synthetic Data

- Virtually unlimited free labeled data can be generated
- Allows for corner/dangerous scenario generation
- Flexibility in sensor set and robot configurations during development



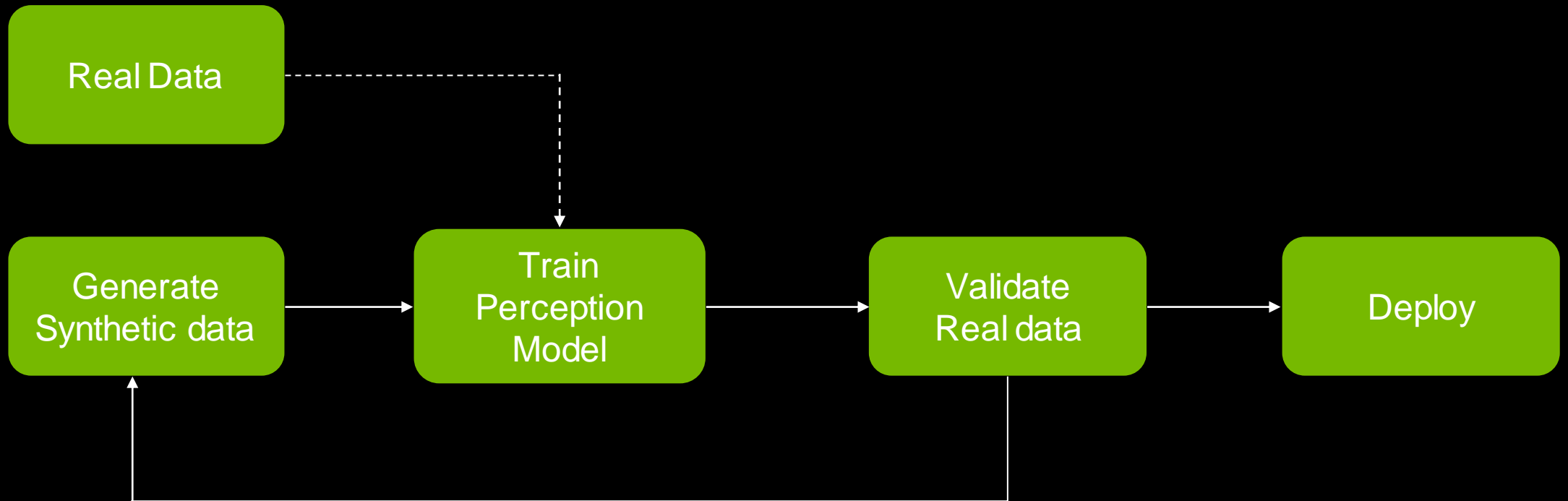
# Workflow Difference

## Normal Workflow



# Workflow Difference

## Sim Training Workflow





# Robotics Perception

## Simulated data pitfalls

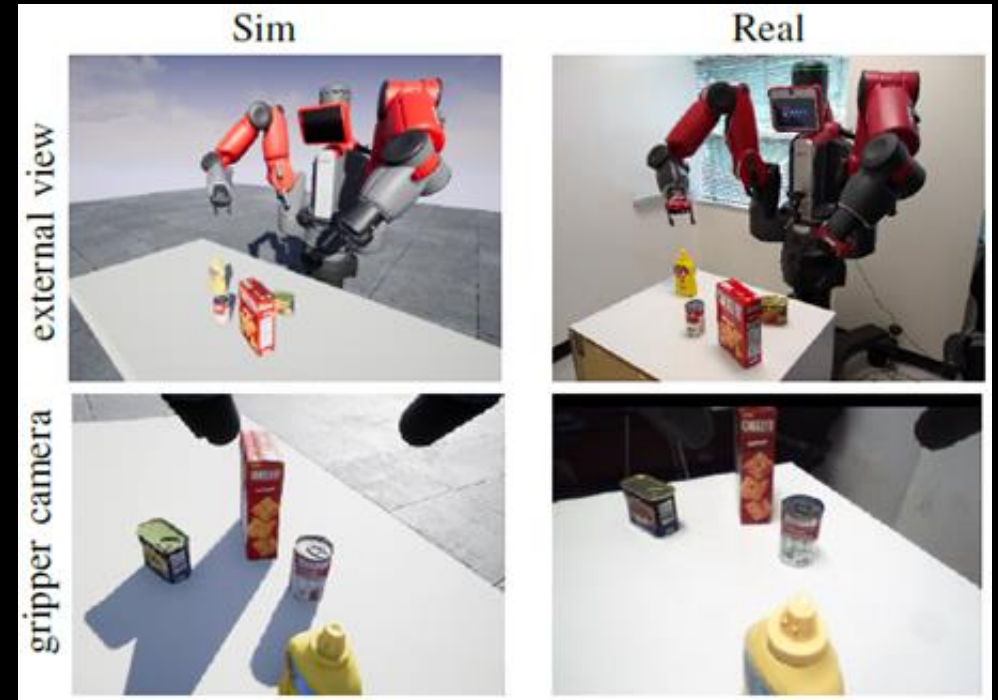
- Sim2Real gap - The domain gap between simulated data and real world data
- Models trained in simulation without proper configurations fail in the real world



# Sim2Real Gap

## Appearance Gap

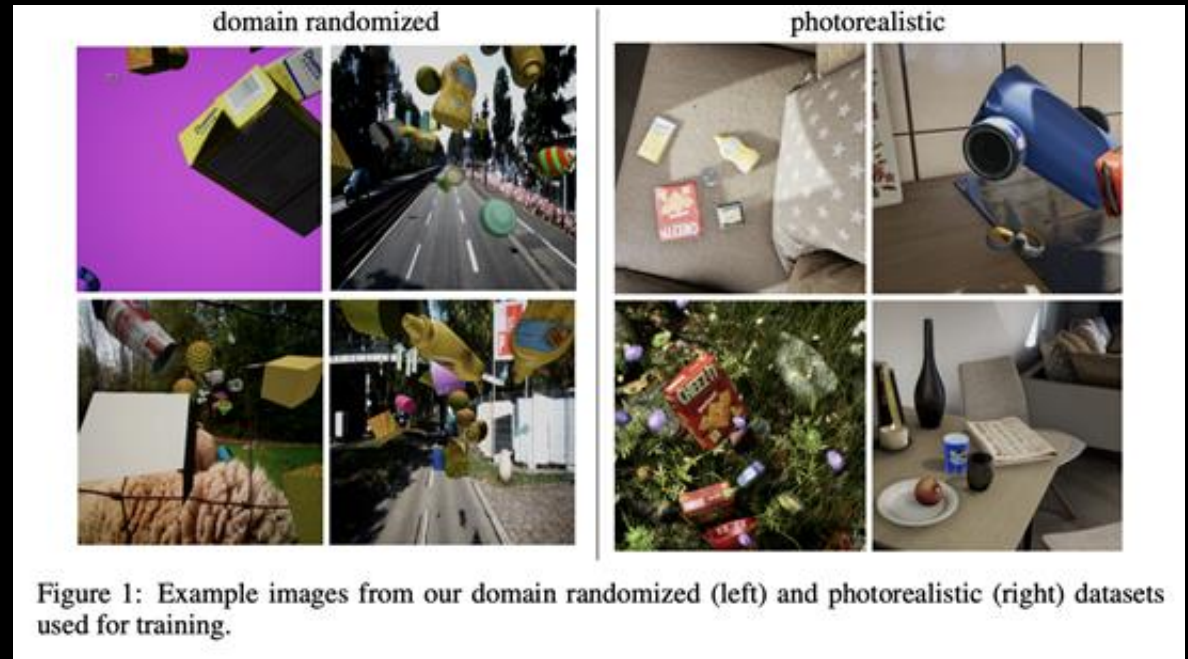
- The appearance gap is the inability to make simulated images exactly replicate what the real world looks like
- Until simulators become completely photorealistic there will always be an appearance gap
- Luckily with progression in simulators this gap is quickly closing



# Sim2Real Gap

## Content Gap

- Simulation mimics a limited set of scenes, not necessarily reflecting the diversity and distribution of objects of those captured in the real world
- The real world is composed of many different objects and structures that can be hard to recreate in simulation



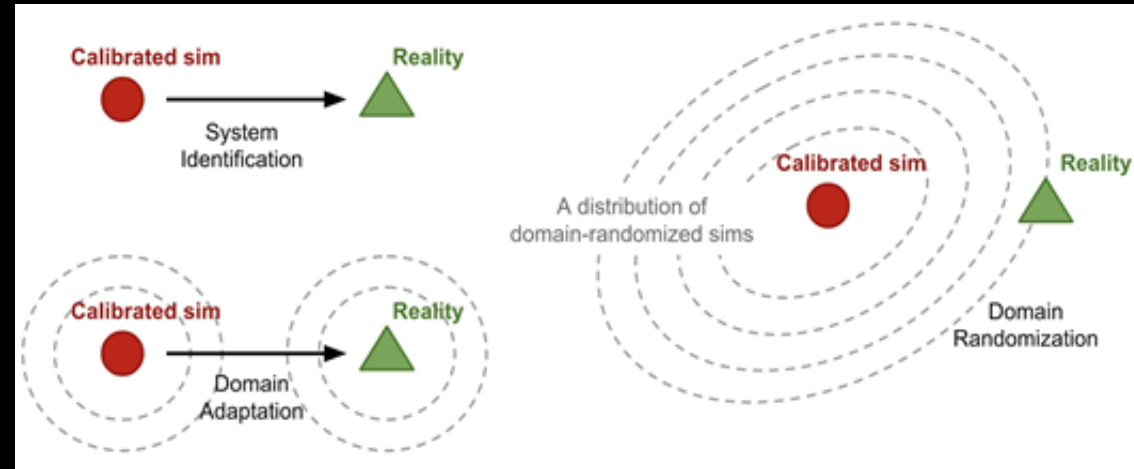
# Closing the Gap

## Techniques

Domain Adaptation - Techniques to transfer the source domain (simulation) to the target domain (real world)

Domain Randomization - Create diverse randomized simulation variations where reality is seen as just another variation by the DNN

System Identification - Create an accurate simulation that matches the properties of reality



# Closing the Gap

## OV Accurate Simulation

RTX photo-realism

Realistic Physics

Fully configurable sim properties








Domain Randomization

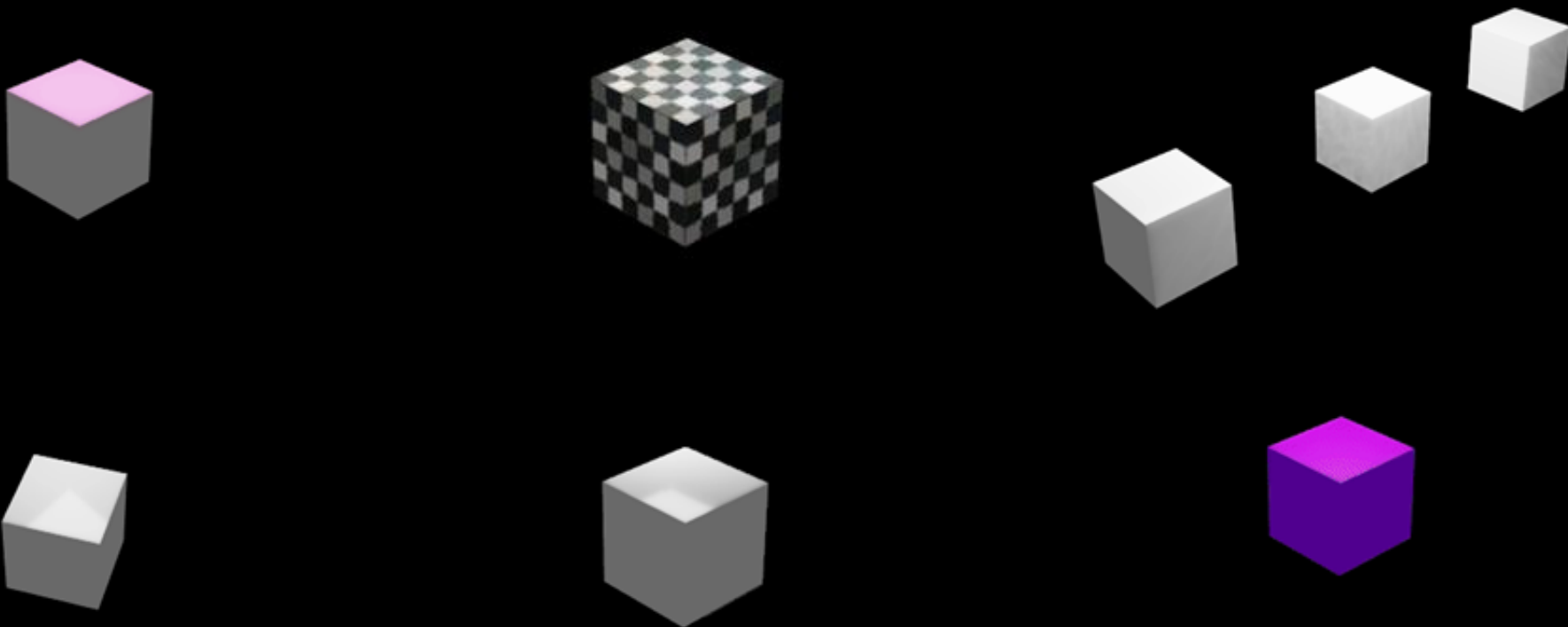


The background is a dark blue gradient. It features a complex network of thin, light green lines that crisscross the frame. Interspersed along these lines are several bright green, glowing circular dots of varying sizes. Some dots are larger and more prominent, while others are smaller and fainter. The overall effect is a sense of dynamic movement and interconnectedness, typical of a network visualization or a simulation environment.

**Tuning the  
simulation**

# A Lot of levers

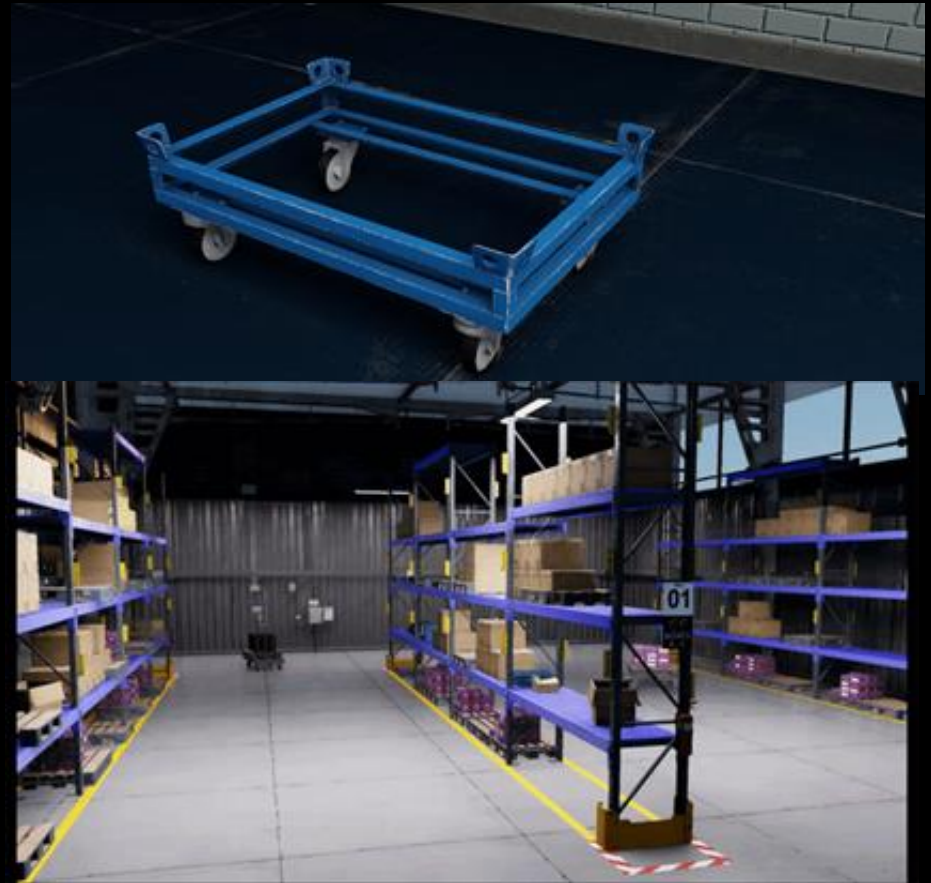
## Domain Randomization



# Problem set up

## Dolly Detection

- In order to provide examples for many of the tuning concepts the problem of trying to detect bounding boxes on dollies in the real world using a model trained only on simulated data will be used
- Model: Detectnet\_v2
- Simulator: Isaac Sim



# Simulation Tuning

## Textures

- Most CNNs are texture learners
- Need accurate textures or diverse set of domain randomized textures for the objects you would like to detect
- It is key to minimize gaps in simulation that the model can exploit
- Without accurate textures on the objects of interest the model will completely fail to generalize to the real world



Original Dolly



Dolly w/ noise

# Simulation Tuning

## Textures

- A solution that has worked well is to randomize the textures of the object of interest
  - No longer reliant on super accurate modeling
- A large diverse set of textures is necessary for success
- Without enough diversity the DNN will memorize the incorrect textures leading to degradation in performance



# Simulation Tuning

## Environment Textures

- It is also important when possible to randomize the textures of other parts of the scene
- In particular problems such as free space segmentation this is required for robust sim2real
  - Randomizing imperfections such as cracks and stains in the floor





# Simulation Tuning

## Lighting

- Randomizing lighting is one of the most important domain parameters to randomize as simple lighting conditions such as windows can confuse networks
- DNNs get confused by shadows and lighting artifacts very easily
- Ray tracing becomes very important in Sim2Real performance



# Simulation Tuning

## Lighting

- An example of this in action where just by opening and closing windows allows for much better generalization
- You can also add random lights around scene for extra randomized conditions
  - This will increase the variability between dataset generation runs



Randomly open and closing windows



Windows held constant

# Simulation Tuning

## Lighting

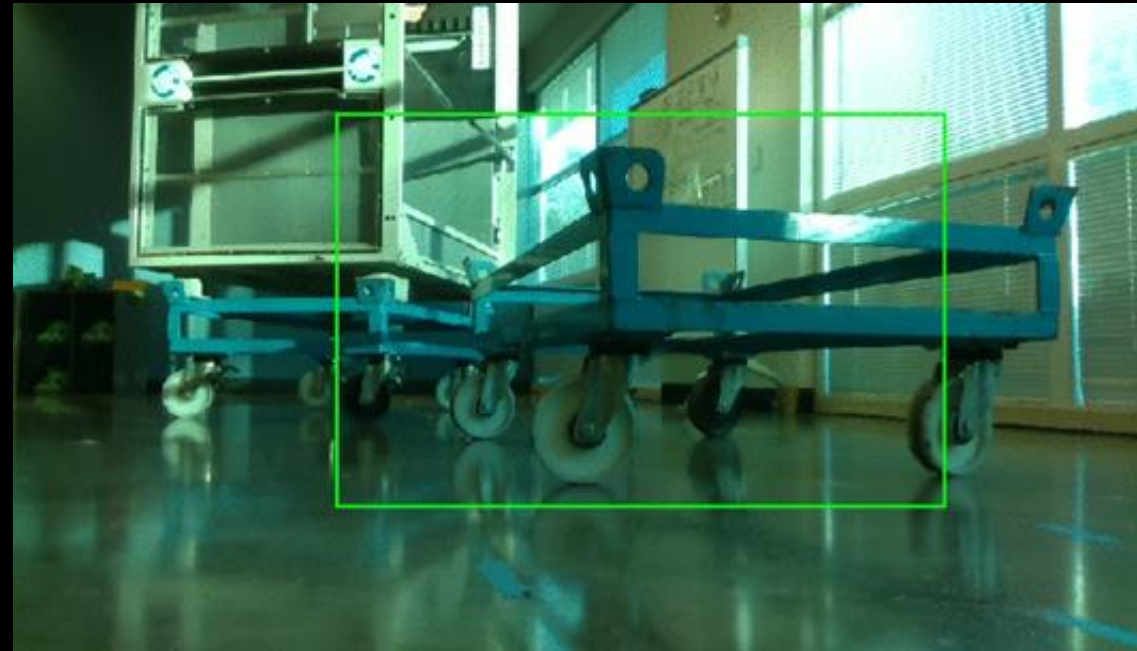
- We must remember to randomize brightness of the lighting within a scene
- If lights are held constant the algorithms will fail in low light or over exposed situation
- On the right we can see a low light scene that a model fails on as domain randomized lights were not lowered enough



# Simulation Tuning

## Lighting

- The model on the right is heavily confused by saturation and color change
- Randomizing saturation and light colors brings some of the largest improvement in sim2real performance with very low risk



# Simulation Tuning

## Lighting

- A common problem for simulation trained models is the ability to handle reflections
- Without having Ray tracing enabled and reflective materials in your scene the perception systems will miss-detect reflections





# Common Pitfall

## Object of Interest Perspective

- It is imperative to capture data points with the objects of interests in a diverse set of perspectives
- Neural networks struggle to infer the structure of objects like humans
- This includes distance as well where far away objects become more difficult to detect



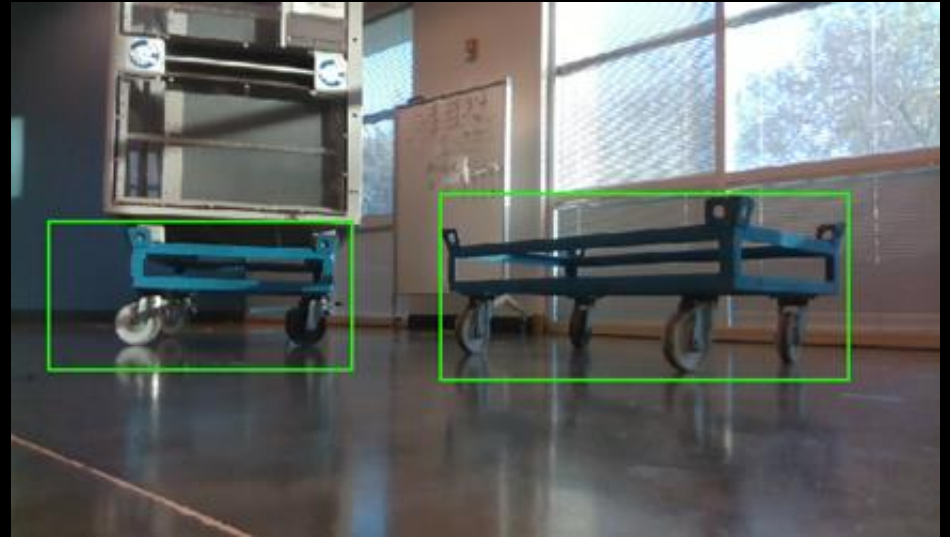


# Common Pitfall

## Object of Interest Perspective



Trained without Dolly rotation

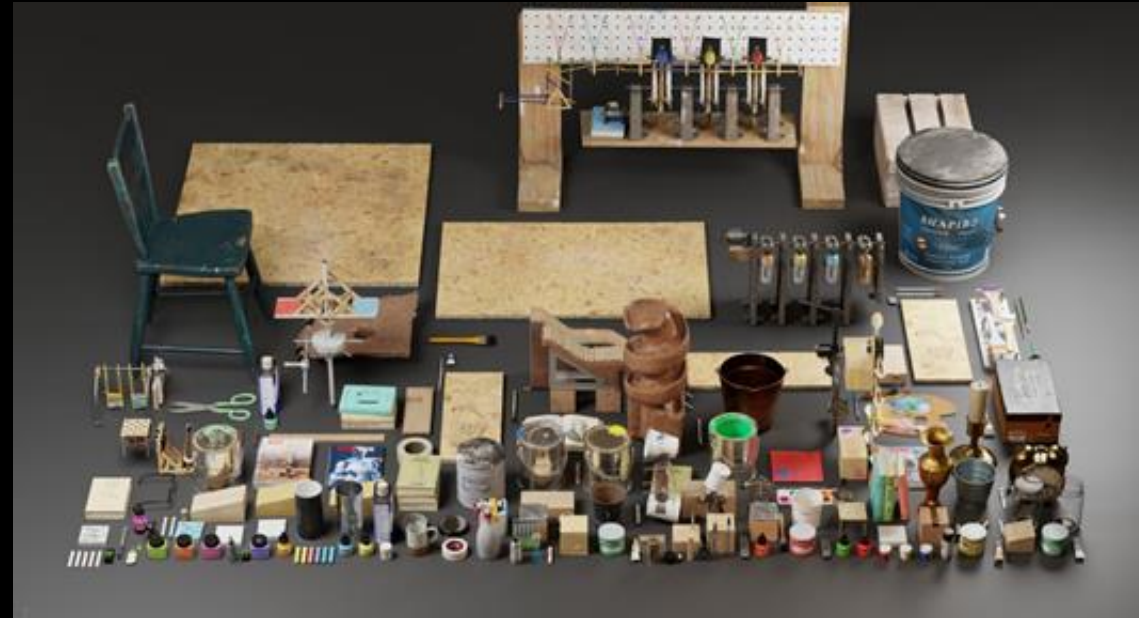


Trained with Dolly rotation

# Simulation Tuning

## Diversity of Content

- The more diverse and representative the content of the scene the better the model will generalize to the real world
- Real data naturally has high variance and a lot of diversity
- This is especially helpful with false positives



# Simulation Tuning

## Diversity of Content

- In experiments the DNN often would mistakenly detect Kaya robots in the scene as dollies
  - By adding kaya robots into training scenario the problem was resolved
- It is infeasible to model the entire world but the more diverse the content the more robust the model
  - Problem is easier under constrained domains



# Simulation Tuning

## Structural Content Gap

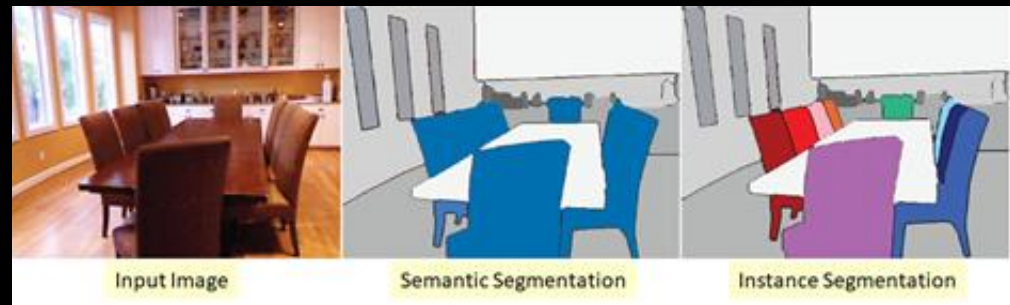
- Addressing the content gap in term of structure is important for hard to detect objects such as small and occluded objects
  - Networks often rely on the context around an object for detection
- In the past random Distractors (see right) were used but this eliminates the ability of the NN to use context around objects
  - The solution is to randomize in a structured way



# Simulation Tuning

## Structural Content Gap

- Maintaining contextual information when randomizing is important
  - Keeping randomized objects and distractors within realistic bounds allows the model to learn correlations that help detection
- In the bottom right example the green occluded chair can be identified more easily as it has the table to give context to the scene
- Later we will see more advanced approaches using structured domain randomization

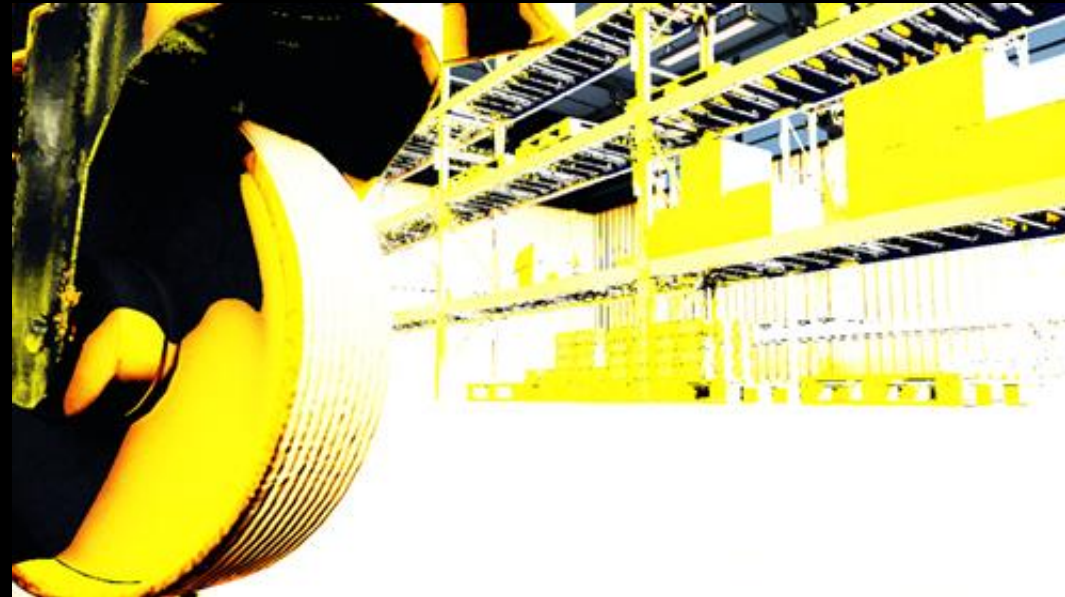




# Simulation Tuning

Is there too much variation?

- With few exceptions it is beneficial to keep the simulation parameters within the bounds of reality
  - Object of interest textures can be an exception
- If a human would struggle with annotation, then the model probably will as well
  - Lighting can be difficult to tune in this regard as there is a wide range unknown lighting scenarios





# Simulation Tuning

## Controlling Data Distribution

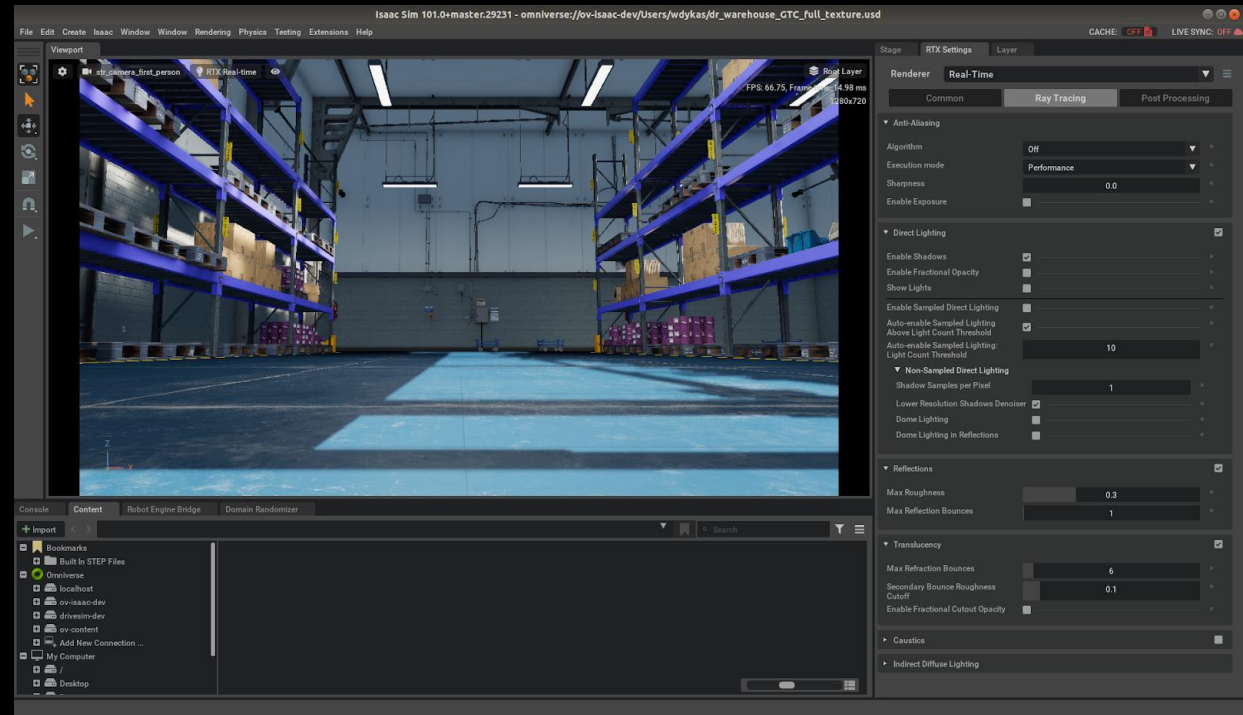
- An advantage of having control over data generation is the ability to control the data distribution
- By varying the distribution of objects in a scene you can make the algorithm more robust
  - A broad distribution of objects of interest will make the model more robust
  - Varying the distribution also means making sure other synthetic parameters are not randomized in a way the model can take advantage of



# Simulation Tuning

## Resolution Tradeoff

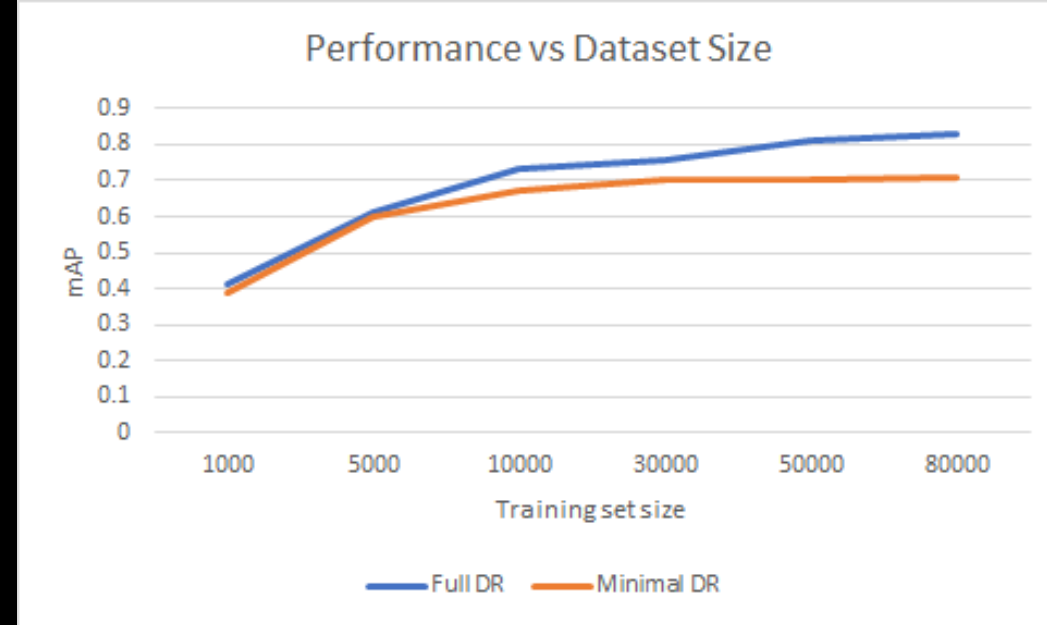
- The plateau of performance gain from increasing resolution fidelity has not been hit by even the highest quality simulators
- However increasing rendering quality dramatically increases the time to produce data
- When compute constrained you can have render setting lowered when configuring DR parameters then increase fidelity at later tuning stages
  - This includes resolution, ray tracing sampling, and ray tracing modes



# Simulation Tuning

## Scaling Dataset Generation

- Synthetic data performance improvement quickly plateaus without careful consideration
- Diversity and Photorealism is the key if you want to scale the dataset size
  - If data points being generated are not novel there will be nothing new for the model to learn
- On the right the dataset with minimal DR stop improving



# Mixing data

## FineTuning with Real Data

- If you have a small set of labeled data that can be held out from validation it can be extremely beneficial to fine tune the final model on real data
- It has been proven in many scenarios that you can get superior performance by using a mix of real and synthetic data rather than just real data
  - A model trained on synthetic data provides a better base to fine tune on then other models trained on large generic datasets such as imagenet

	Synthetic Only	Real Only	Synthetic + Real
mAP	.85	.93	.95

# Validation

## Tuning for downstream task

- A robust and carefully designed validation set makes tuning more reliable
- Validation on real data
  - Validating the model on real data in most cases is required before full deployment as it is hard to predict generalization to the real world
- Validation on synthetic data
  - Generating difficult scenarios for a validation set allows for more accurate tuning
  - Any model can accurately predict the easy case but edge cases is where you find a usable model

# Validation

## Synthetic Scenarios

- Having a set of difficult synthetic scenarios will allow for better tuning
- Want to cover a broad range of difficult problems and failure cases
  - Occluded object
  - Low light situations
  - Highly saturated camera
  - Problems specific to your task
- When failure cases are found in deployment recreating them in simulation allows for consistent iteration





# Sim2Real Gap

## Key Takeaways

- Simulation offers a solution to the data scarcity problem in robotic perception
- Often tuning the simulator can become more of an art than a science
- Diversity that mimics reality is the key to Sim2Real transfer at scale
- Perception models trained in simulation can offer a great prior to be fine tuned on for maximum performance



The background of the slide is a dark, almost black, field. It is populated with numerous thin, light green lines that crisscross the frame in various directions. Interspersed among these lines are several small, bright green circular dots or nodes. Some of these dots are slightly larger and more prominent than others. The overall effect is one of a complex, interconnected network or a digital space, possibly representing data flow or a futuristic theme.

# Advanced Topics and the Future

# Advanced Topics

## Adaptive and Learnable Simulators

Guided Domain Randomization

Domain Adaptation

Structured Domain Randomization

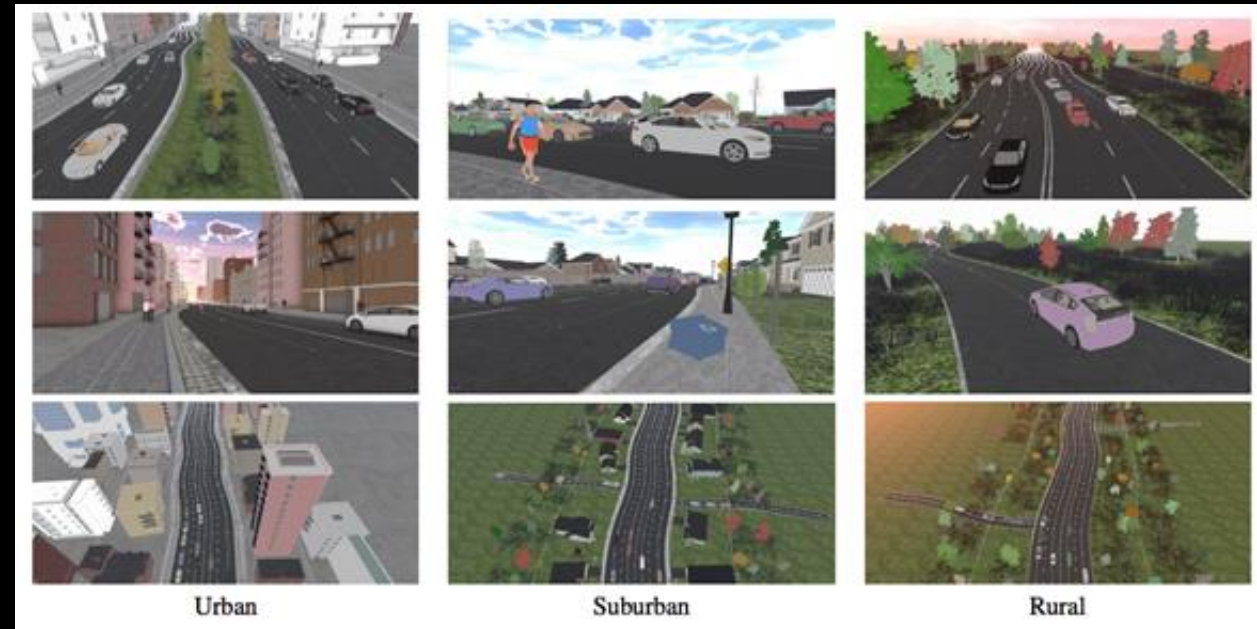
Metasim 1 / 2



# Advanced Topics

## Structured Domain Randomization

- Rather than placing distractors randomly, they are probabilistically placed based on the target problem
- Allows for the network to take better advantage of context
  - Especially beneficial for detection of small and occluded objects



# Advanced Topics

## MetaSim

- The dataset generator is now parameterized by a neural network
- The neural network learns to adjust components of scene graphs to minimize the difference between simulation and the real world
- Allows for optimization of a downstream task automatically if a real dataset is available



# The Future

## Omniverse & Isaac Sim

- Nvidia is continually improving Omniverse and Isaac Sim to bring better rendering and performance
- New domain randomizations tools coming out for more advanced data generation
- Continually adding support for new sensors and sensor models





# Thank You

- Feel free to email me at [wdykas@nvidia.com](mailto:wdykas@nvidia.com) with any questions or comments
- Check out: **Sim2Real in Isaac Sim** for more information on specific tools in Isaac Sim
- For more information on Structured domain randomization and synthetic data generation for self driving cars checkout **NVIDIA Omniverse + DriveSim for Synthetic Data Generation**

# References

Structured Domain Randomization - <https://arxiv.org/abs/1810.10093>

Metasim - <https://arxiv.org/abs/1904.11621>

Metasim 2 - <https://arxiv.org/abs/2008.09092>

Deception net - <https://arxiv.org/pdf/1904.02750.pdf>

Slide 12 Graphic - <https://lilianweng.github.io/lil-log/2019/05/05/domain-randomization.html>

