Sujoy Ganguly | Head of Applied Machine Learning Research, Unity | April 25th, 2022 | Eurographics

Synthetic Data For Computer Vision: Techniques, Challenges, and Tools

Talk Outline

Talk Outline

1. Introduction

Talk Outline

1. Introduction
2. Methods to Bridge the Sim-to-Real Gap

Talk Outline

1. Introduction
2. Methods to Bridge the Sim-to-Real Gap
3. Burdens of Domain Randomization

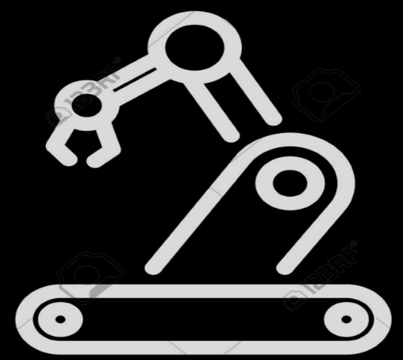
Talk Outline

1. Introduction
2. Methods to Bridge the Sim-to-Real Gap
3. Burdens of Domain Randomization
4. Benchmark Environments and Tools to Advance Research in the Sim-to-Real Gap

Labeled data is crucial to train ML Models

Autonomous Vehicles

Detect objects, lane markings, signs and traffic signals

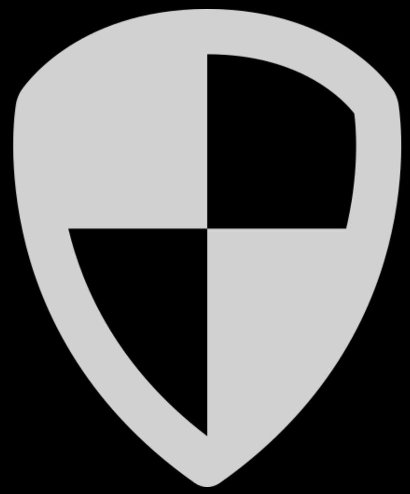


Robotics

Understand their environments, safely interact with humans and recognize products or components

Retail

Cashier-less checkouts, inventory management systems and footfall analysis



Security

Need to identify potential threats

Training with real-world data is challenging

Training with real-world data is challenging

Expensive

Unity

Training with real-world data is challenging

Expensive

Time consuming

Training with real-world data is challenging

Expensive

Time consuming

Biased and insufficient

Training with real-world data is challenging

Expensive

Time consuming

Biased and insufficient

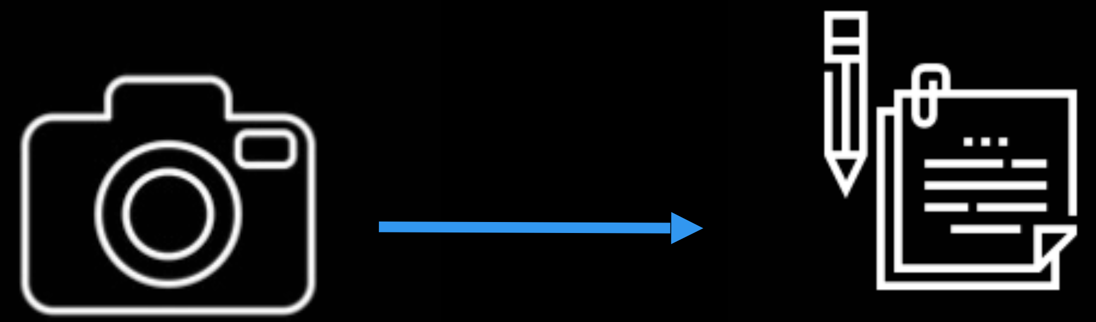
Privacy and compliance

Typical Computer Vision Workflow

Typical Computer Vision Workflow

Acquire Real
World Images

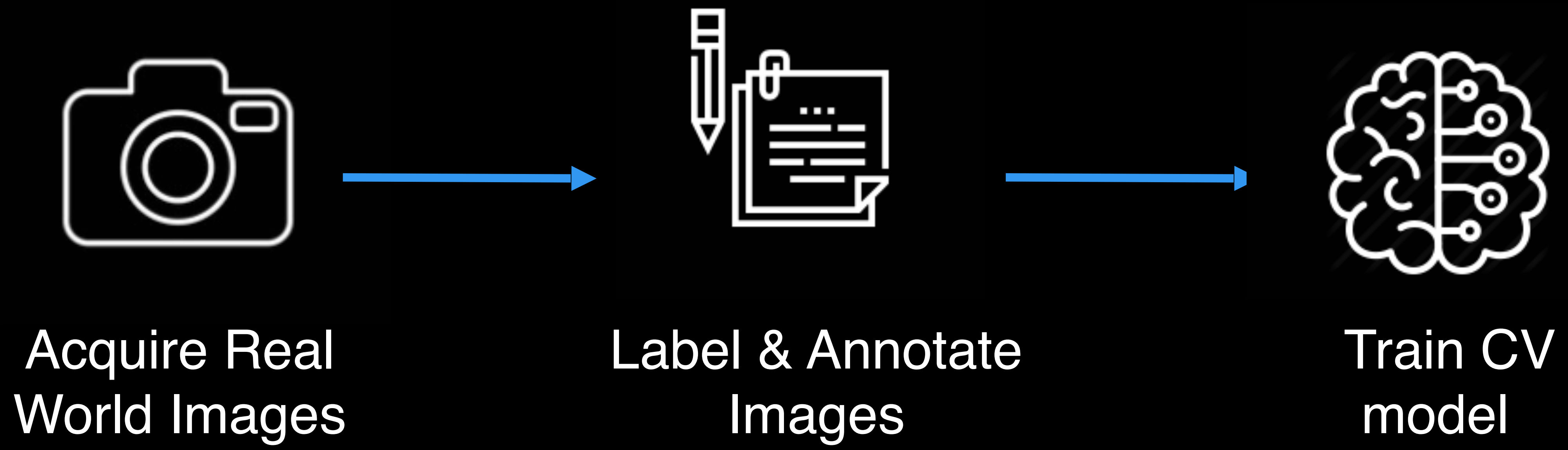
Typical Computer Vision Workflow



Acquire Real World Images

Label & Annotate Images

Typical Computer Vision Workflow



Typical Computer Vision Workflow

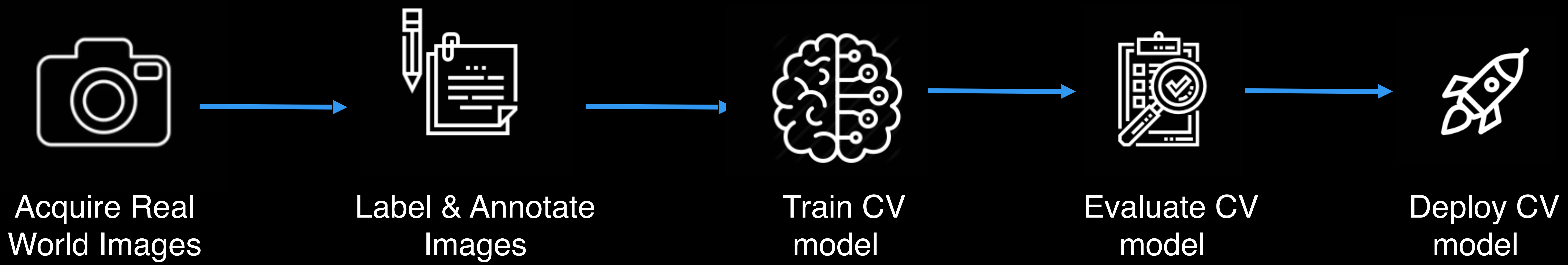
Acquire Real World Images

Label & Annotate Images

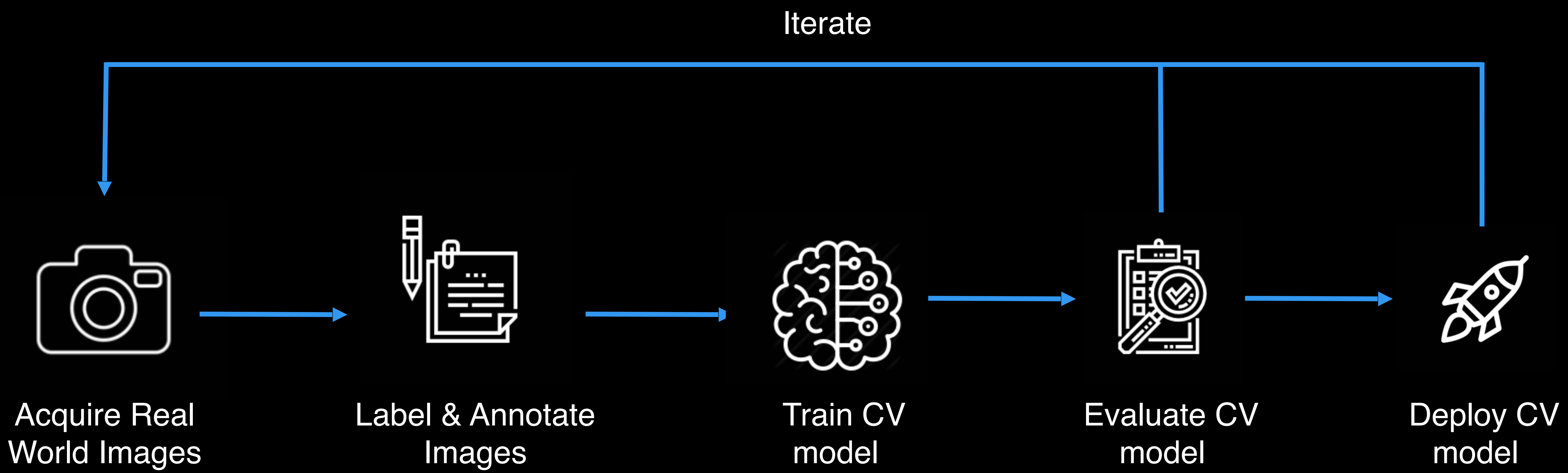
Train CV model

Evaluate CV model

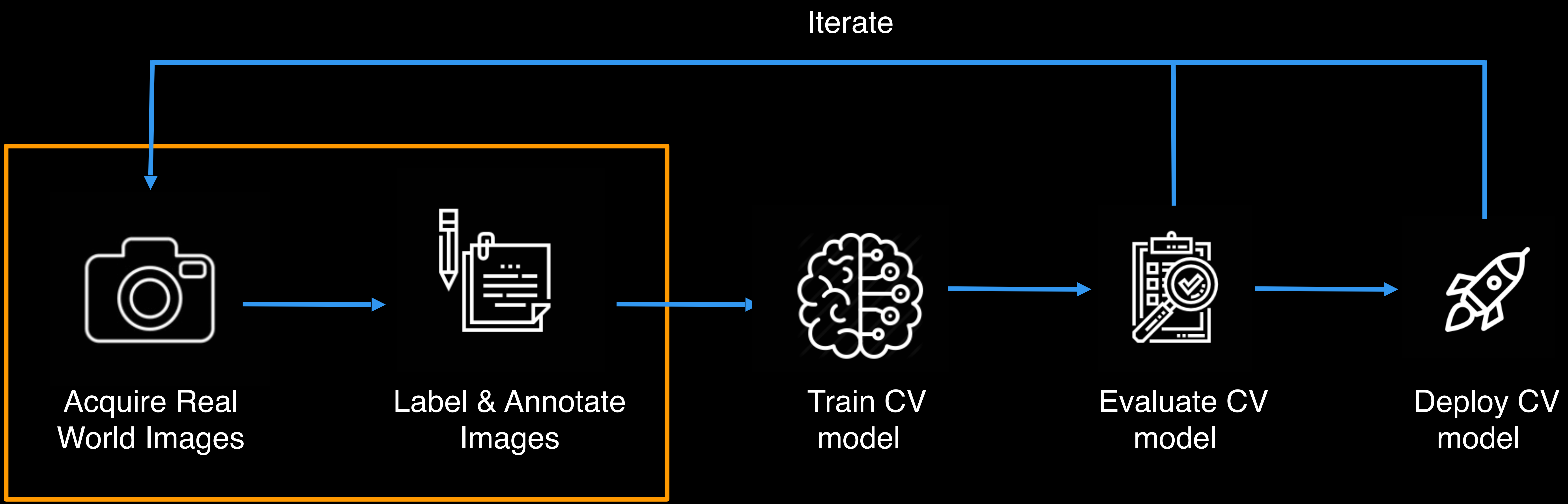
Typical Computer Vision Workflow



Typical Computer Vision Workflow



Typical Computer Vision Workflow

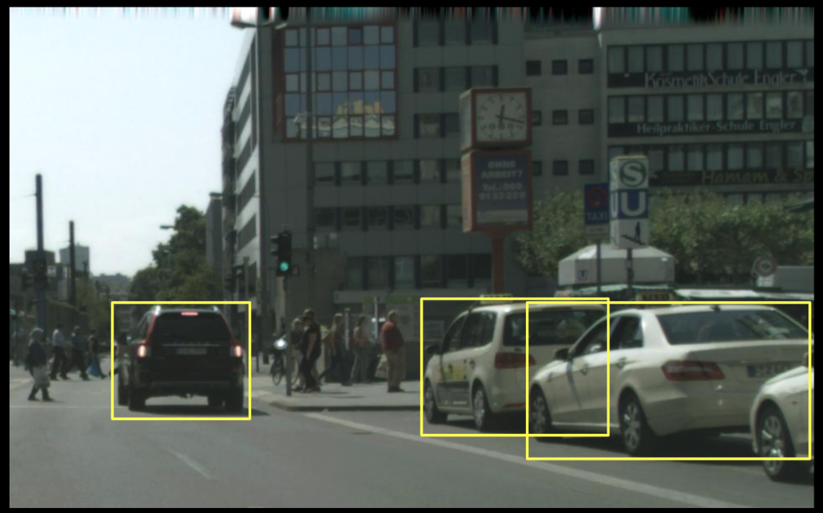


70% time is spent on data collection, labeling and annotation.

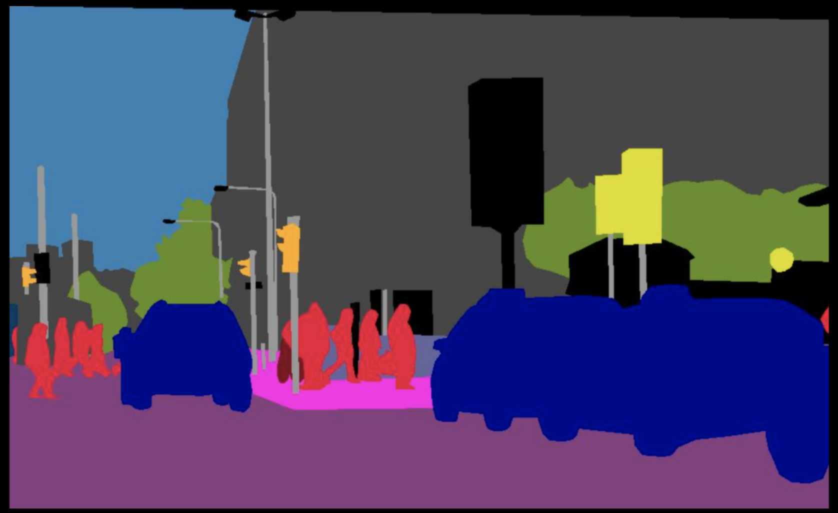
Cost of labeling increases with complexity

Input

Labels

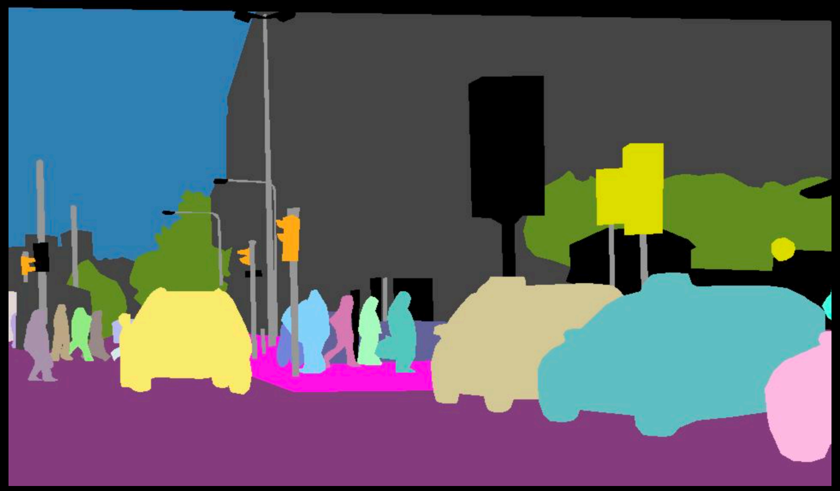


Object detection



Semantic segmentation

Instance segmentation



Panoptic segmentation

The Value of Synthetic Data

The Value of Synthetic Data

Auto-labelled

No human annotation or labelling required

The Value of Synthetic Data

Auto-labelled

No human annotation or labelling required

Privacy

Compliant with GDPR and privacy standards

The Value of Synthetic Data

Auto-labelled

No human annotation or labelling required

Privacy

Compliant with GDPR and privacy standards

Iterative

Generate variations in datasets with simple code changes

The Value of Synthetic Data

Auto-labelled

No human annotation or labelling required

Privacy

Compliant with GDPR and privacy standards

Iterative

Generate variations in datasets with simple code changes

Affordable

Small teams/startups can generate massive dataset within budget

The Value of Synthetic Data

Auto-labelled

No human annotation or labelling required

Privacy

Compliant with GDPR and privacy standards

Iterative

Generate variations in datasets with simple code changes

Affordable

Small teams/startups can generate massive dataset within budget

Representative

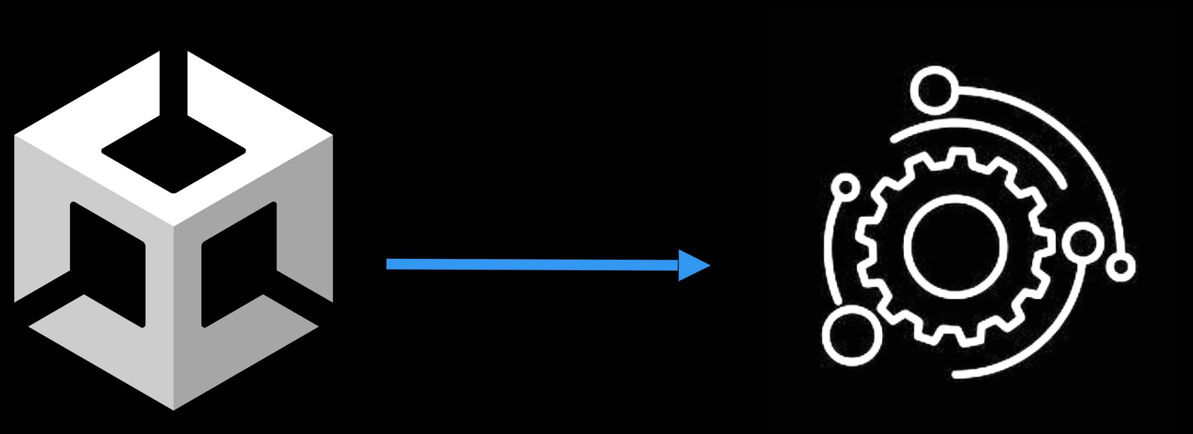
Produce training dataset that is variant and captures the real world complexity

Synthetic Data trained Computer Vision Workflow

Synthetic Data trained Computer Vision Workflow

Build Environment

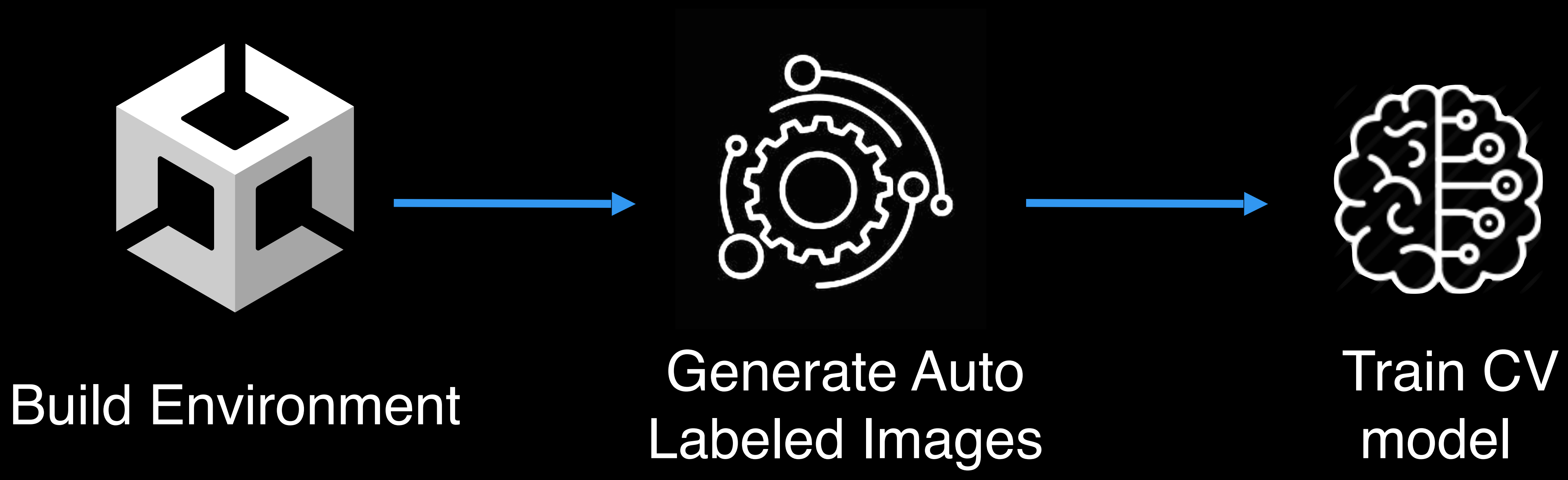
Synthetic Data trained Computer Vision Workflow



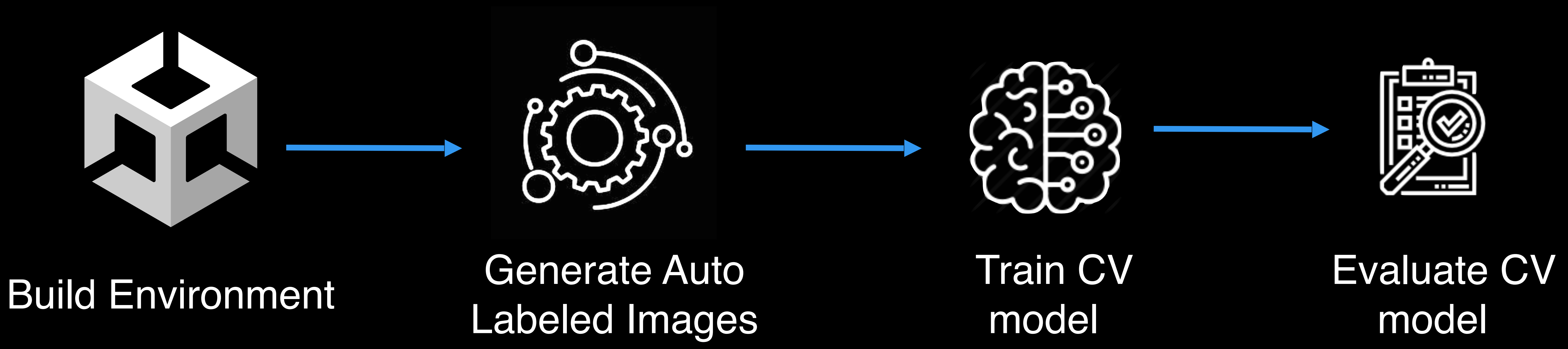
Build Environment

Generate Auto Labeled Images

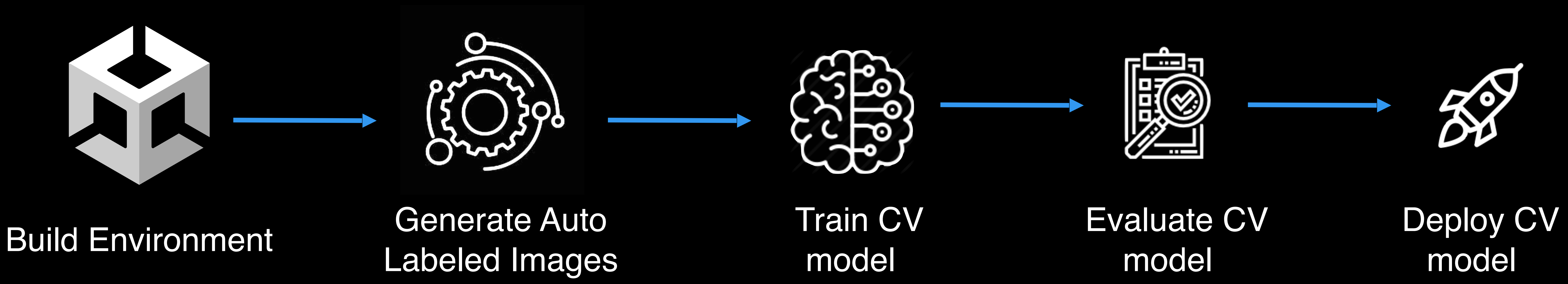
Synthetic Data trained Computer Vision Workflow



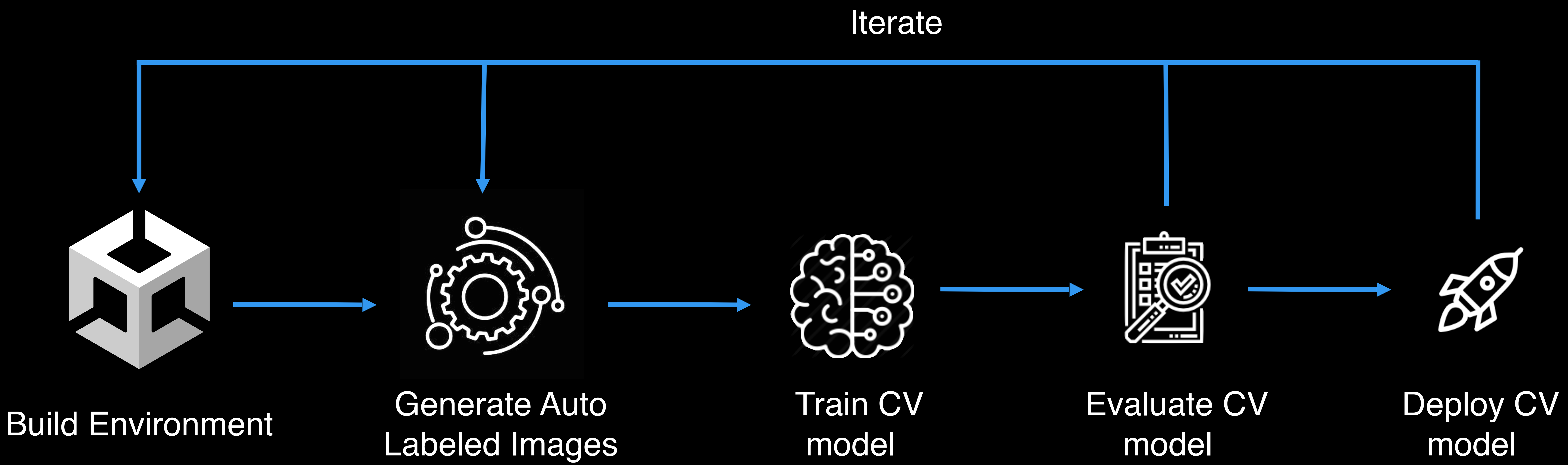
Synthetic Data trained Computer Vision Workflow



Synthetic Data trained Computer Vision Workflow



Synthetic Data trained Computer Vision Workflow



Domain Randomization and the Sim-to-Real Gap

Domain Randomization

- Create the most diverse data set that the model can learn by varying properties of the simulation[1,2].
- For Example:
  - Spatial Location and Orientations
  - Color and texture of the background
  - Lighting
  - Optical Occlusions
  - Camera position, orientation, and field of view

Domain Randomization

- Create the most diverse data set that the model can learn by varying properties of the simulation[1,2].
- For Example:
  - Spatial Location and Orientations
  - Color and texture of the background
  - Lighting
  - Optical Occlusions
  - Camera position, orientation, and field of view

Domain Randomized images with bounding box labels

[1]Tobin et al. "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World," 2017 IROS
[2]Hinterstoisser et al. "An Annotation Saved is an Annotation Earned: Using Fully Synthetic Training for Object Instance Detection, " 2019 ICCVW

Intrinsic Variations vs Extrinsic Variations

Intrinsic Variations vs Extrinsic Variations

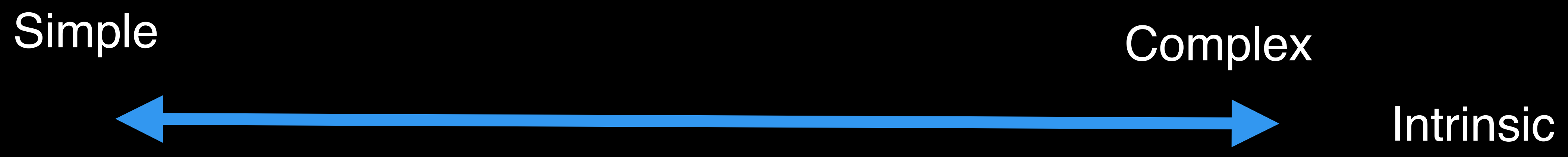
- Intrinsic: Features of the target object, e.g. shape, texture, color.

Intrinsic Variations vs Extrinsic Variations

- Intrinsic: Features of the target object, e.g. shape, texture, color.

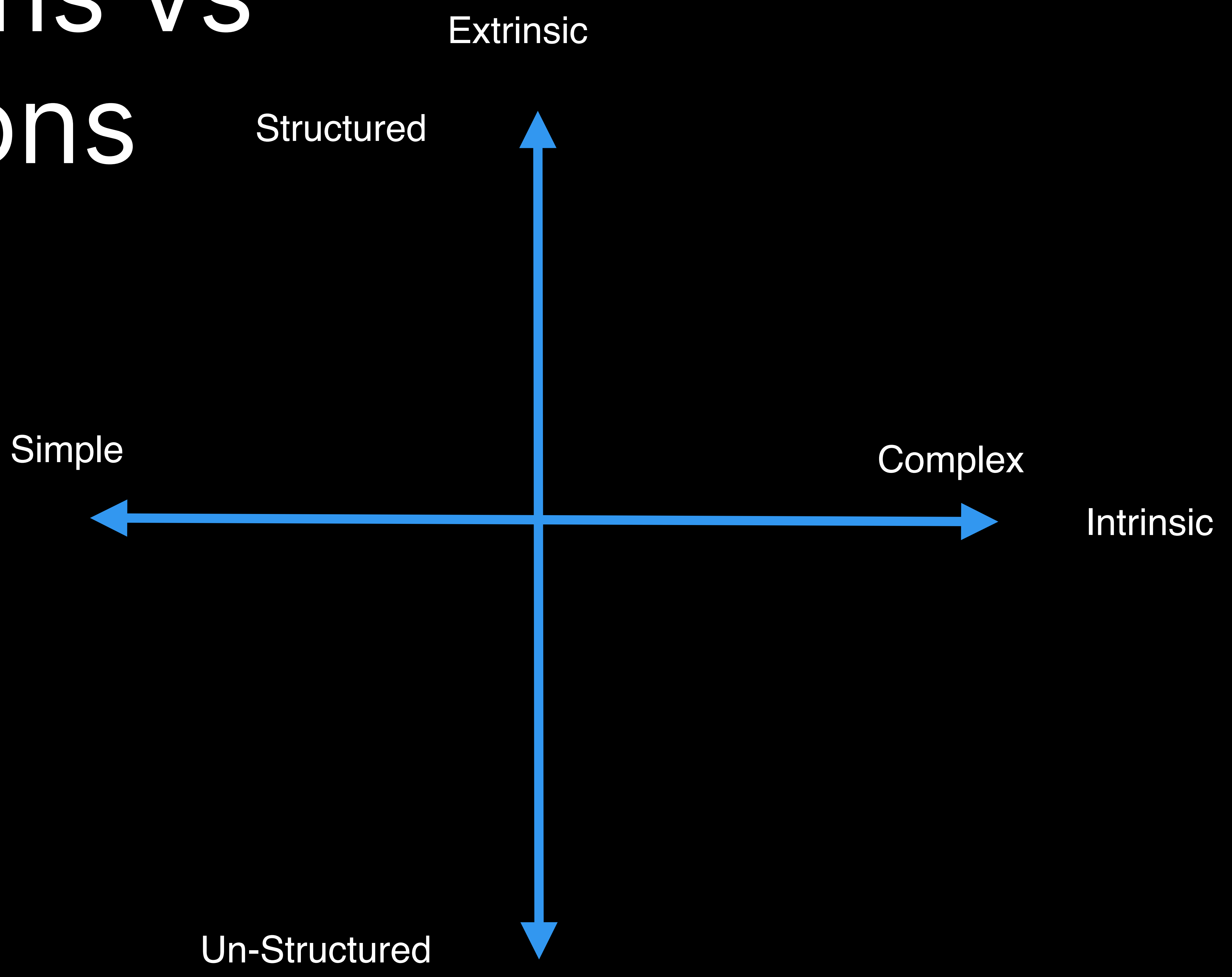
Intrinsic Variations vs Extrinsic Variations

- Intrinsic: Features of the target object, e.g. shape, texture, color.
- Extrinsic: Features of the environment, e.g. camera positions, object placement and orientation.



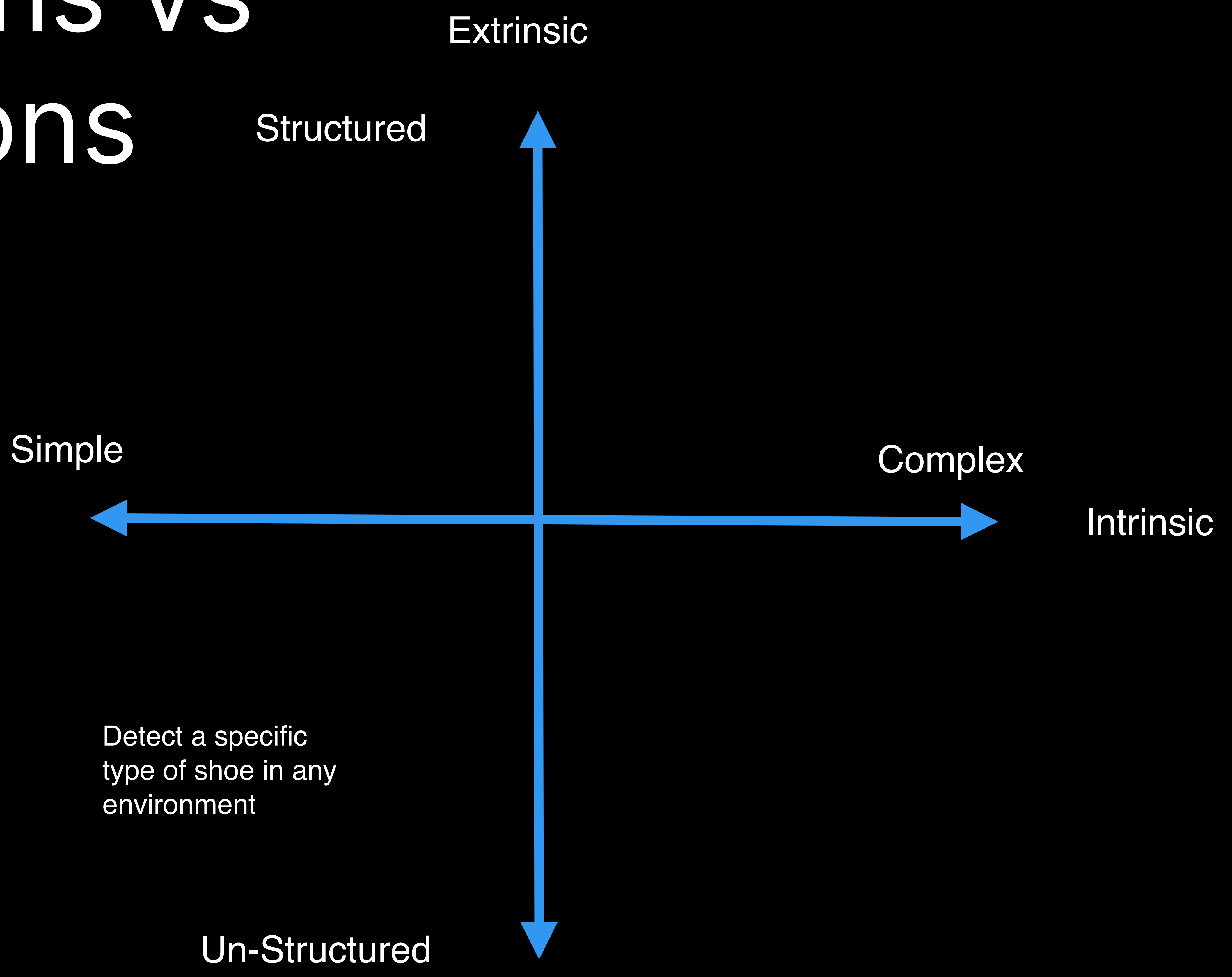
Intrinsic Variations vs Extrinsic Variations

- Intrinsic: Features of the target object, e.g. shape, texture, color.
- Extrinsic: Features of the environment, e.g. camera positions, object placement and orientation.



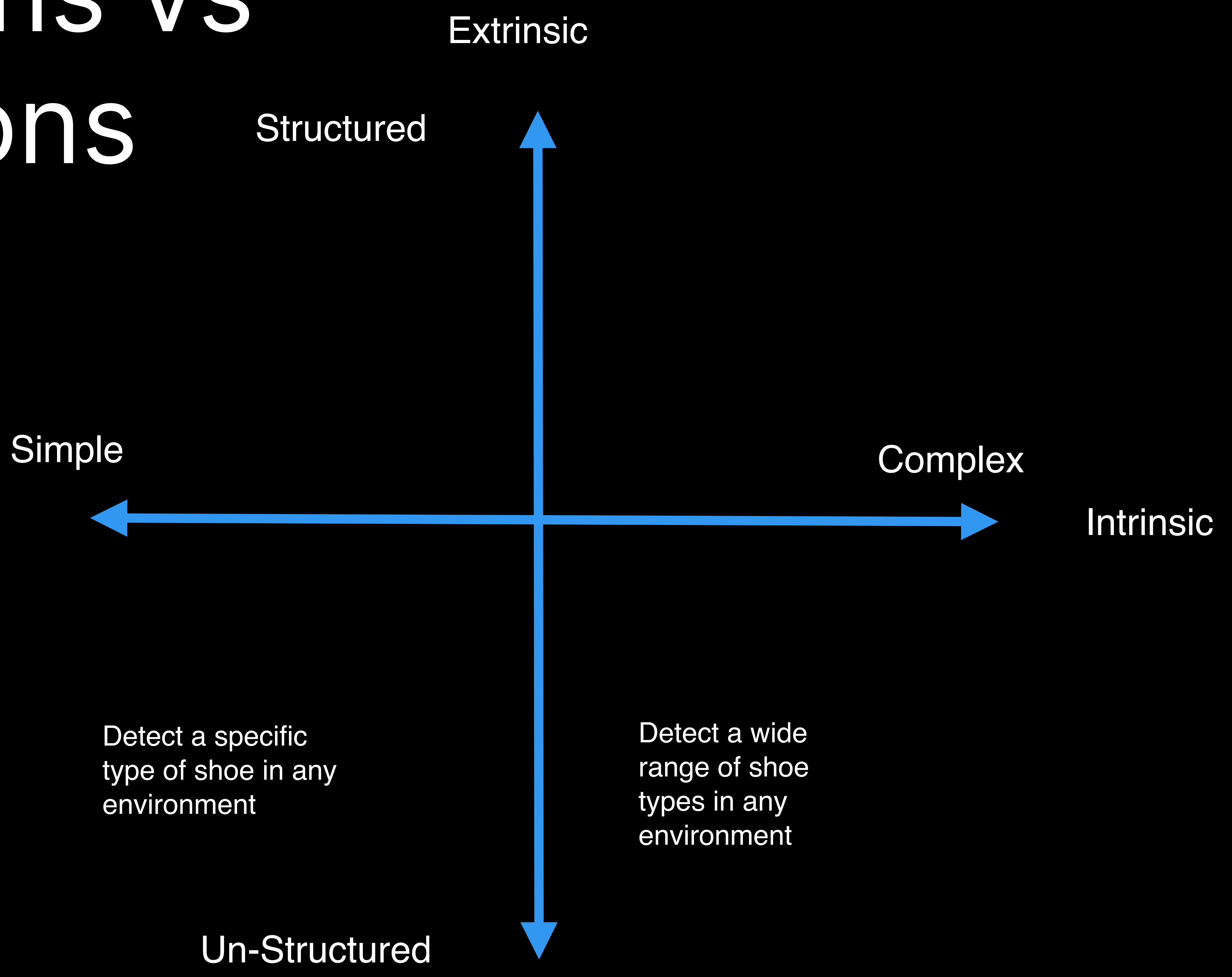
Intrinsic Variations vs Extrinsic Variations

- Intrinsic: Features of the target object, e.g. shape, texture, color.
- Extrinsic: Features of the environment, e.g. camera positions, object placement and orientation.



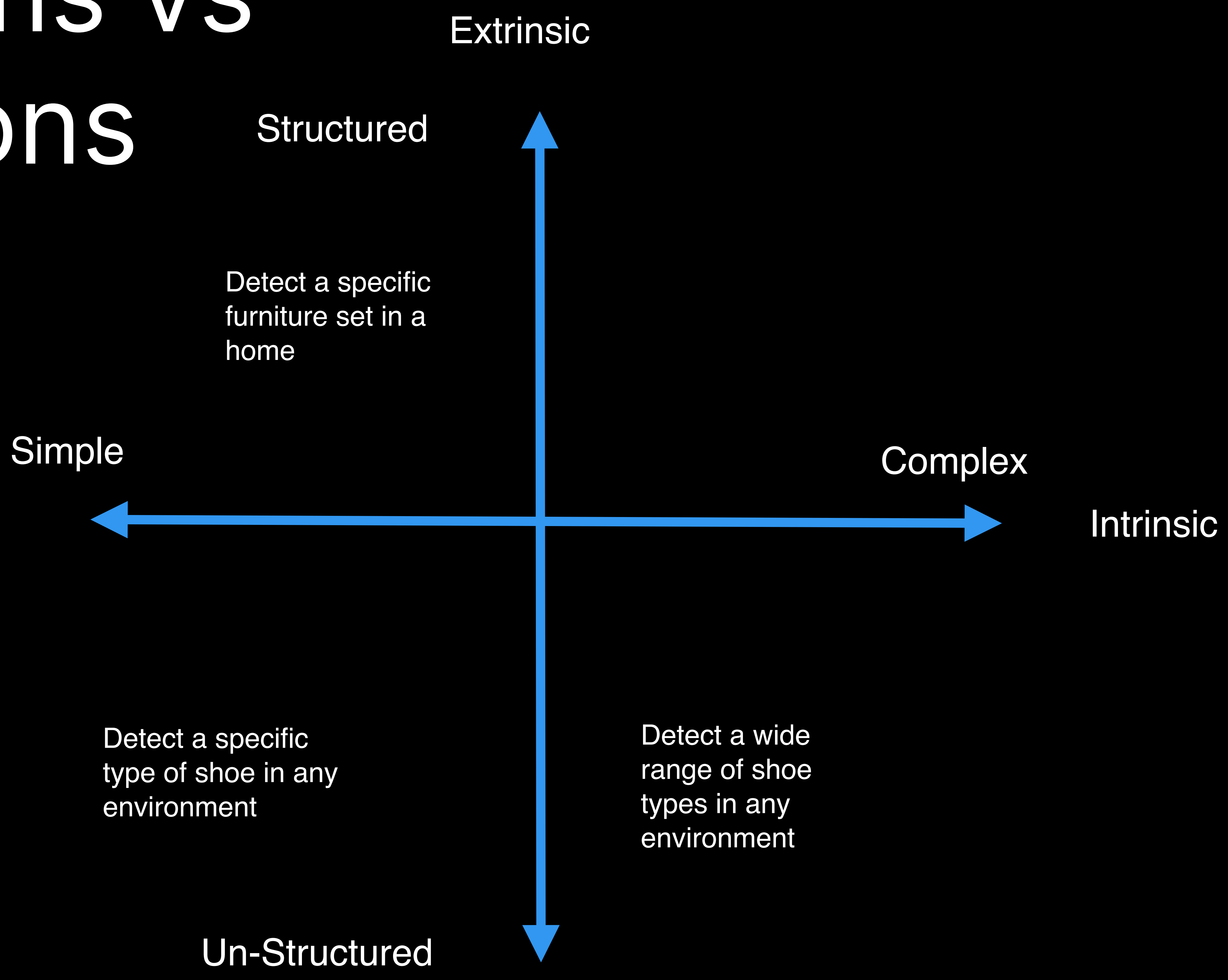
Intrinsic Variations vs Extrinsic Variations

- Intrinsic: Features of the target object, e.g. shape, texture, color.
- Extrinsic: Features of the environment, e.g. camera positions, object placement and orientation.



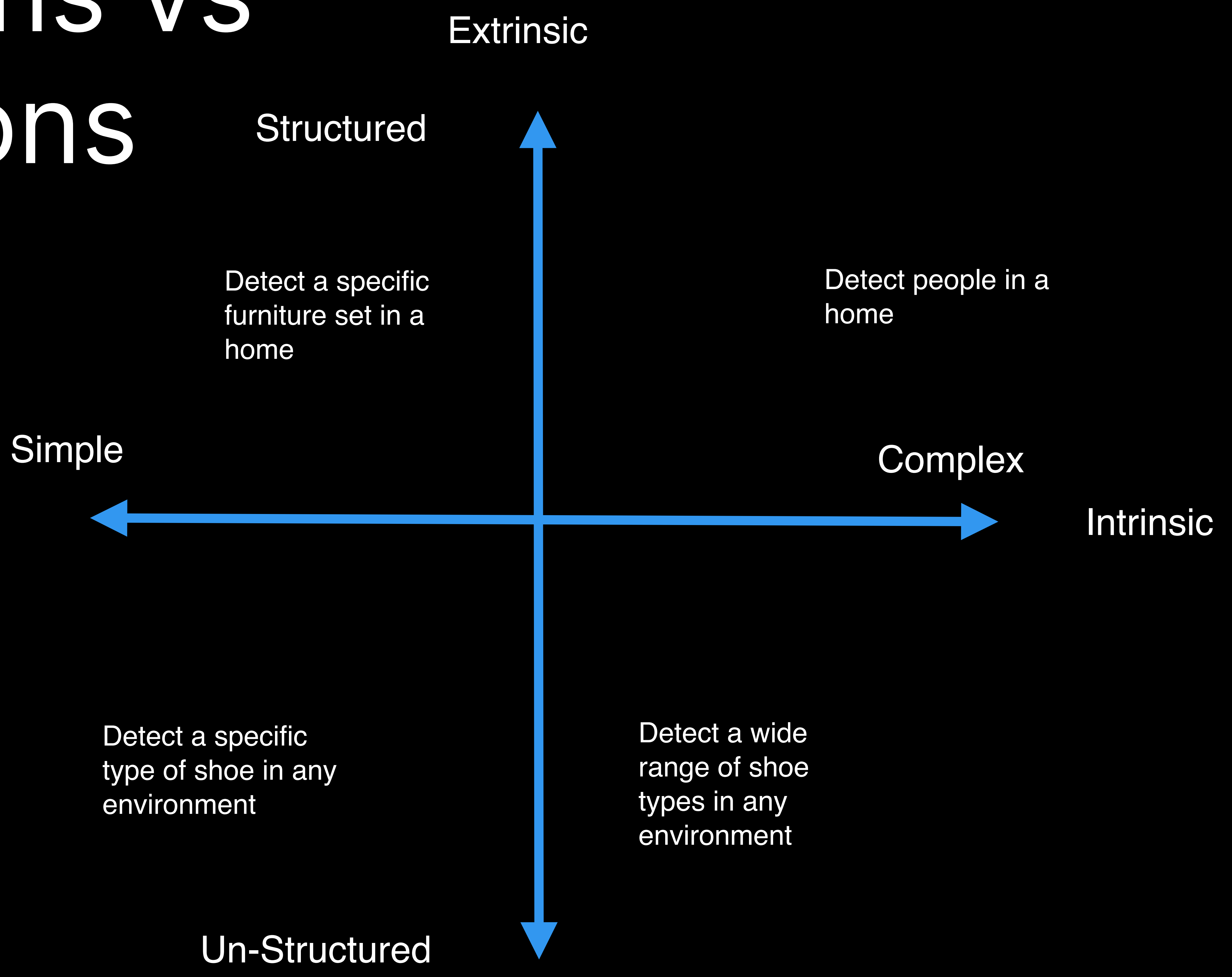
Intrinsic Variations vs Extrinsic Variations

- Intrinsic: Features of the target object, e.g. shape, texture, color.
- Extrinsic: Features of the environment, e.g. camera positions, object placement and orientation.



Intrinsic Variations vs Extrinsic Variations

- Intrinsic: Features of the target object, e.g. shape, texture, color.
- Extrinsic: Features of the environment, e.g. camera positions, object placement and orientation.



Burdens of Domain Randomization

Levels of Content for Machine Learning

Levels of Content for Machine Learning

- Visual
  - Renders correctly under a wide range of conditions, e.g., assets should be free of smoothing or hard/soft edge adverse shading (Improper smoothing producing dark or flashing artifacts on a mesh)
  - Appearance can be varied, e.g., textures and materials can be changed programmatically

Levels of Content for Machine Learning

- Visual
  - Renders correctly under a wide range of conditions, e.g., assets should be free of smoothing or hard/soft edge adverse shading (Improper smoothing producing dark or flashing artifacts on a mesh)
  - Appearance can be varied, e.g., textures and materials can be changed programmatically
- Physical
  - Assets should have accurate colliders
  - Mass and density can be varied
  - Friction, etc. can be varied

Levels of Content for Machine Learning

- Visual
  - Renders correctly under a wide range of conditions, e.g., assets should be free of smoothing or hard/soft edge adverse shading (Improper smoothing producing dark or flashing artifacts on a mesh)
  - Appearance can be varied, e.g., textures and materials can be changed programmatically
- Physical
  - Assets should have accurate colliders
  - Mass and density can be varied
  - Friction, etc. can be varied
- Kinematic
  - Objects can be rigged and animated
  - Objects in the same class can share rigs and animations
  - Animations can be varied

Levels of Content for Machine Learning

- Visual
  - Renders correctly under a wide range of conditions, e.g., assets should be free of smoothing or hard/soft edge adverse shading (Improper smoothing producing dark or flashing artifacts on a mesh)
  - Appearance can be varied, e.g., textures and materials can be changed programmatically
- Physical
  - Assets should have accurate colliders
  - Mass and density can be varied
  - Friction, etc. can be varied
- Kinematic
  - Objects can be rigged and animated
  - Objects in the same class can share rigs and animations
  - Animations can be varied
- Fully embodied content
  - Content reacts to the action of agents

Content for Computer Vision (Visual)

Content for Computer Vision (Visual)

1. Meshes must be free of all defects that will cause rendering artifacts, including:

Content for Computer Vision (Visual)

1. Meshes must be free of all defects that will cause rendering artifacts, including:
   a. Coplanar or lamina faces (Faces sharing all vertices)

Content for Computer Vision (Visual)

1. Meshes must be free of all defects that will cause rendering artifacts, including:
   a. Coplanar or lamina faces (Faces sharing all vertices)
   b. Faces with zero area (Faces having no renderable area)

Content for Computer Vision (Visual)

1. Meshes must be free of all defects that will cause rendering artifacts, including:
   a. Coplanar or lamina faces (Faces sharing all vertices)
   b. Faces with zero area (Faces having no renderable area)
   c. Non-manifold geometry (Cannot be unfolded into a continuous flat area)

Content for Computer Vision (Visual)

1. Meshes must be free of all defects that will cause rendering artifacts, including:
   a. Coplanar or lamina faces (Faces sharing all vertices)
   b. Faces with zero area (Faces having no renderable area)
   c. Non-manifold geometry (Cannot be unfolded into a continuous flat area)
   d. Faces that are self-intersecting (Faces with more than one closed contour)

Content for Computer Vision (Visual)

1. Meshes must be free of all defects that will cause rendering artifacts, including:
   a. Coplanar or lamina faces (Faces sharing all vertices)
   b. Faces with zero area (Faces having no renderable area)
   c. Non-manifold geometry (Cannot be unfolded into a continuous flat area)
   d. Faces that are self-intersecting (Faces with more than one closed contour)
   e. Free of unattached vertices

Content for Computer Vision (Visual)

1. Meshes must be free of all defects that will cause rendering artifacts, including:
    a. Coplanar or lamina faces (Faces sharing all vertices)
    b. Faces with zero area (Faces having no renderable area)
    c. Non-manifold geometry (Cannot be unfolded into a continuous flat area)
    d. Faces that are self-intersecting (Faces with more than one closed contour)
    e. Free of unattached vertices
    f. Smoothing or hard/soft edge adverse shading (Improper smoothing producing dark or flashing artifacts on a mesh)

Content for Computer Vision (Visual)

1. Meshes must be free of all defects that will cause rendering artifacts, including:
    a. Coplanar or lamina faces (Faces sharing all vertices)
    b. Faces with zero area (Faces having no renderable area)
    c. Non-manifold geometry (Cannot be unfolded into a continuous flat area)
    d. Faces that are self-intersecting (Faces with more than one closed contour)
    e. Free of unattached vertices
    f. Smoothing or hard/soft edge adverse shading (Improper smoothing producing dark or flashing artifacts on a mesh)
2. UV layout and Set

Content for Computer Vision (Visual)

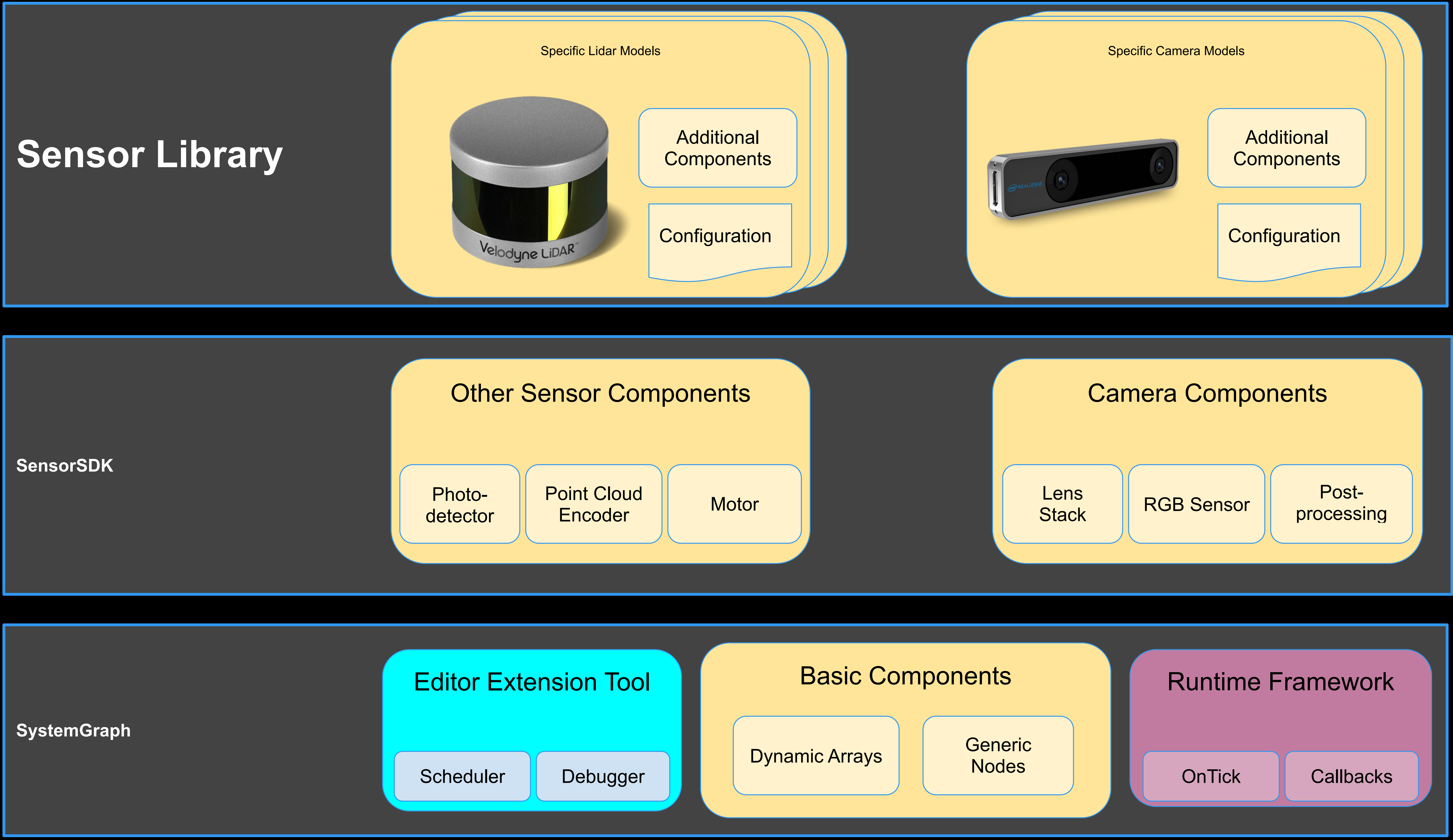
1. Meshes must be free of all defects that will cause rendering artifacts, including:
    a. Coplanar or lamina faces (Faces sharing all vertices)
    b. Faces with zero area (Faces having no renderable area)
    c. Non-manifold geometry (Cannot be unfolded into a continuous flat area)
    d. Faces that are self-intersecting (Faces with more than one closed contour)
    e. Free of unattached vertices
    f. Smoothing or hard/soft edge adverse shading (Improper smoothing producing dark or flashing artifacts on a mesh)
2. UV layout and Set
    a. Distortion-free UV coordinates using UV Set 0 allows for placement of albedo, normal, mask, anisotropy, etc.

Tools for Synthetic Data Generation

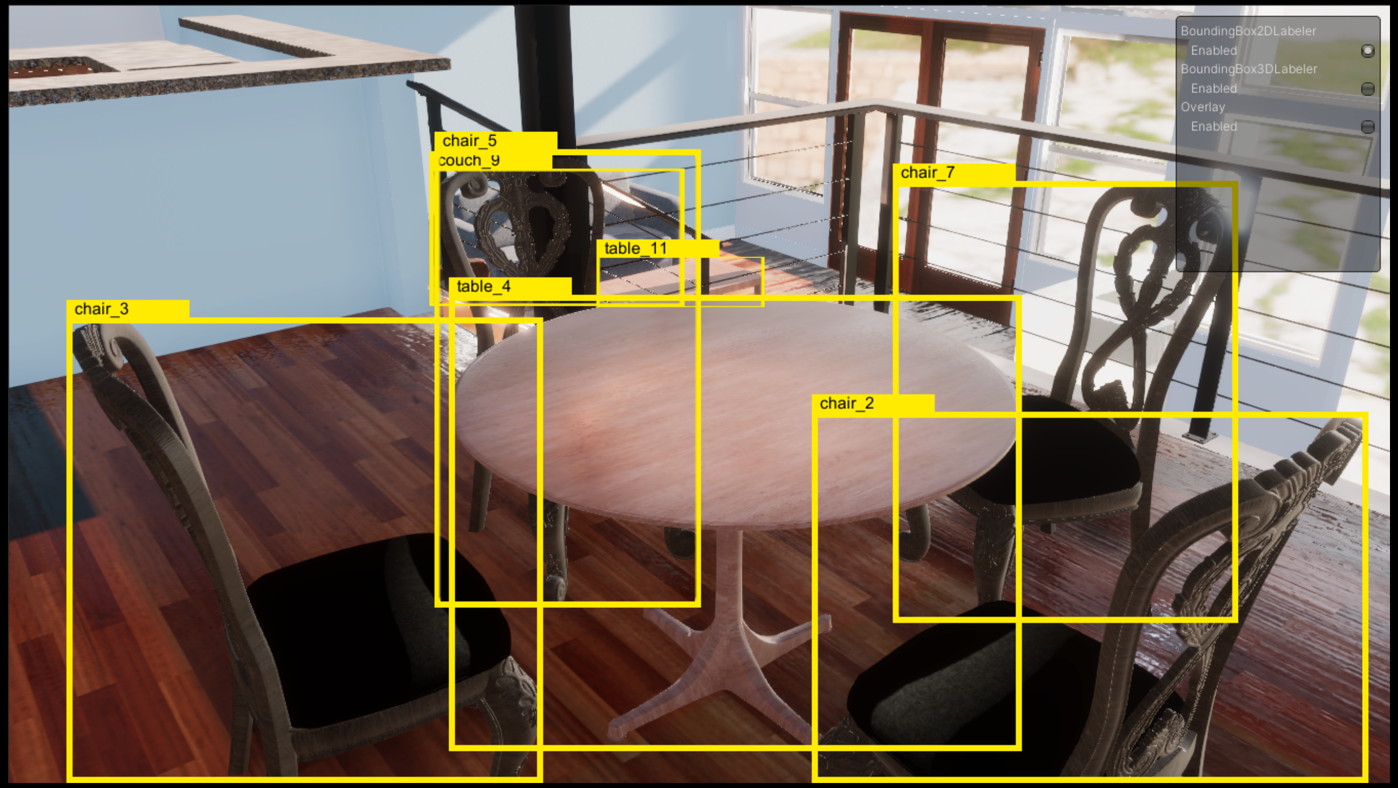
Sensors, Labelers, and Randomizers

- Sensors: Ways of capturing images to be used as input for computer vision models
- Labelers: Ways of capturing labels (Ground Truth) for those images to be used during training of computer vision models
- Randomizers: Ways of varying the scene

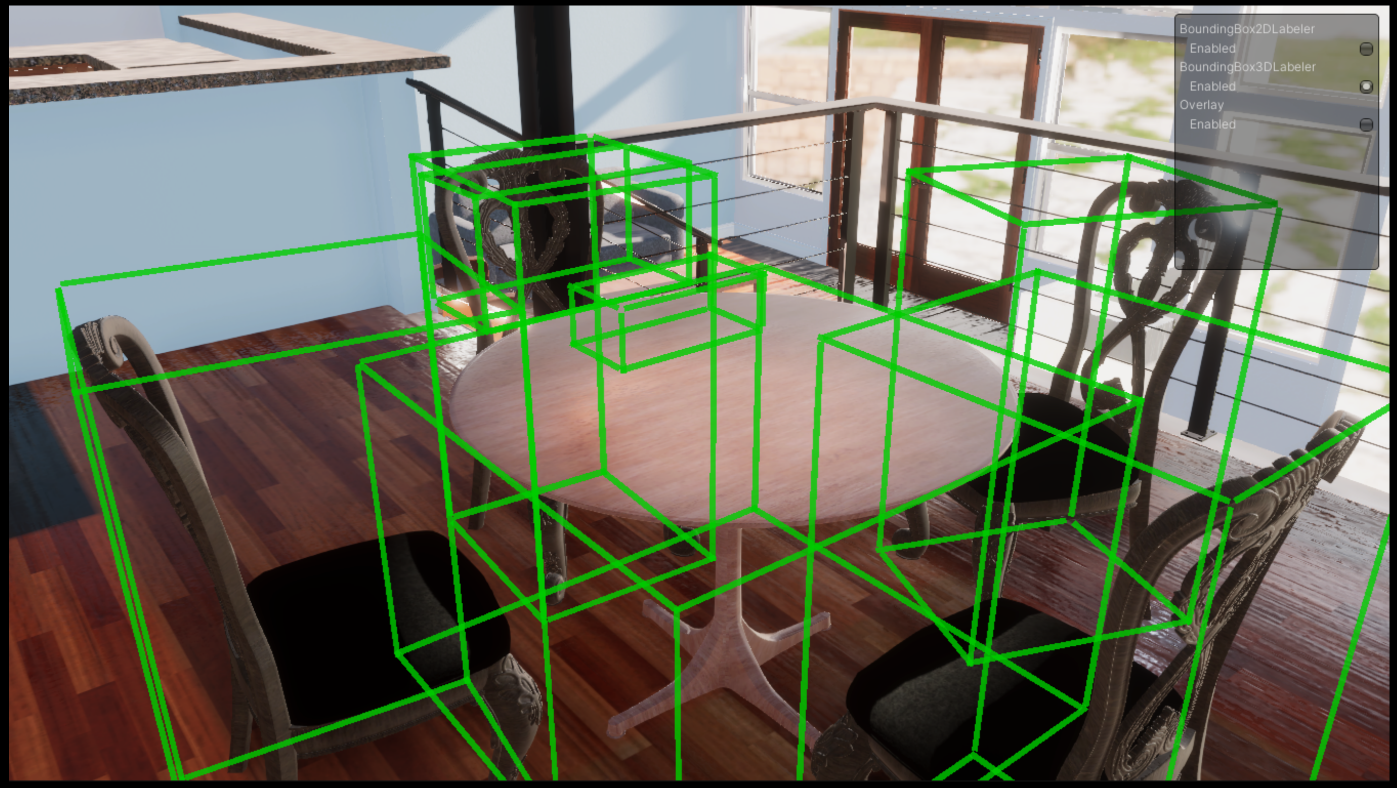
Sensor SDK


Sensor Library
Specific Lidar Models
Velodyne LiDAR
Additional Components
Configuration
Specific Camera Models
Additional Components
Configuration
SensorSDK
Other Sensor Components
Photo-detector
Point Cloud Encoder
Motor
Camera Components
Lens Stack
RGB Sensor
Post-processing
SystemGraph
Editor Extension Tool
Scheduler
Debugger
Basic Components
Dynamic Arrays
Generic Nodes
Runtime Framework
OnTick
Callbacks

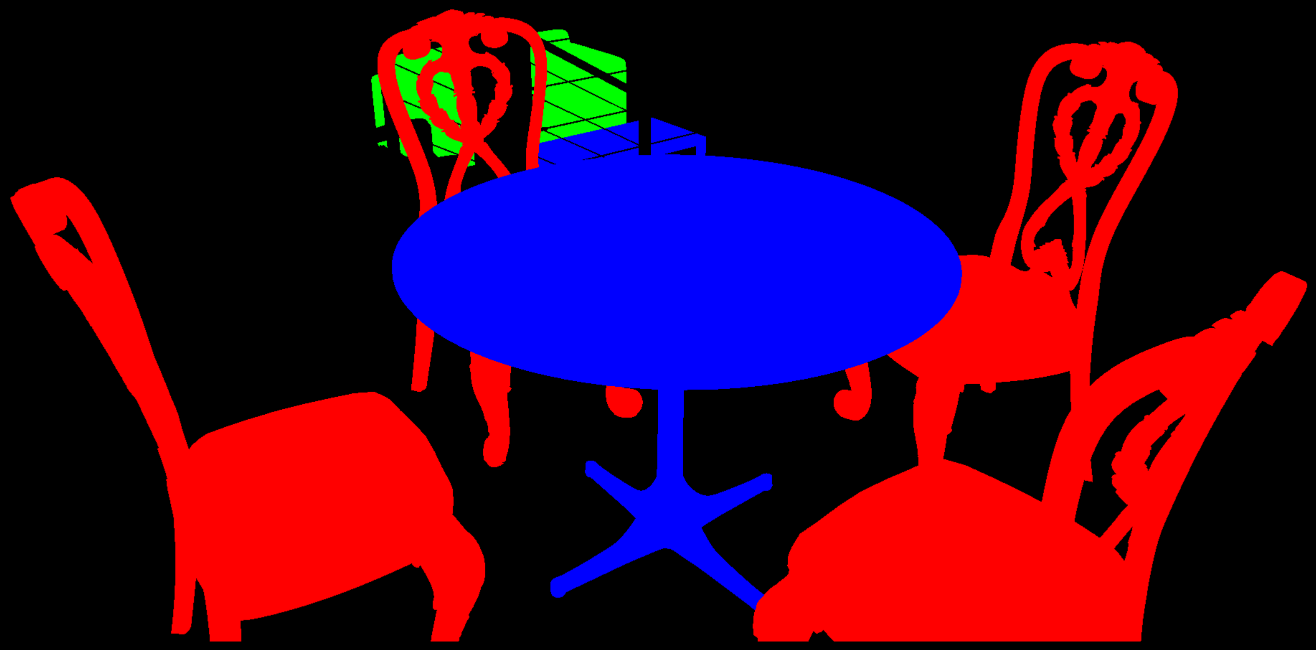
Perception SDK Labelers: Off the Shelf



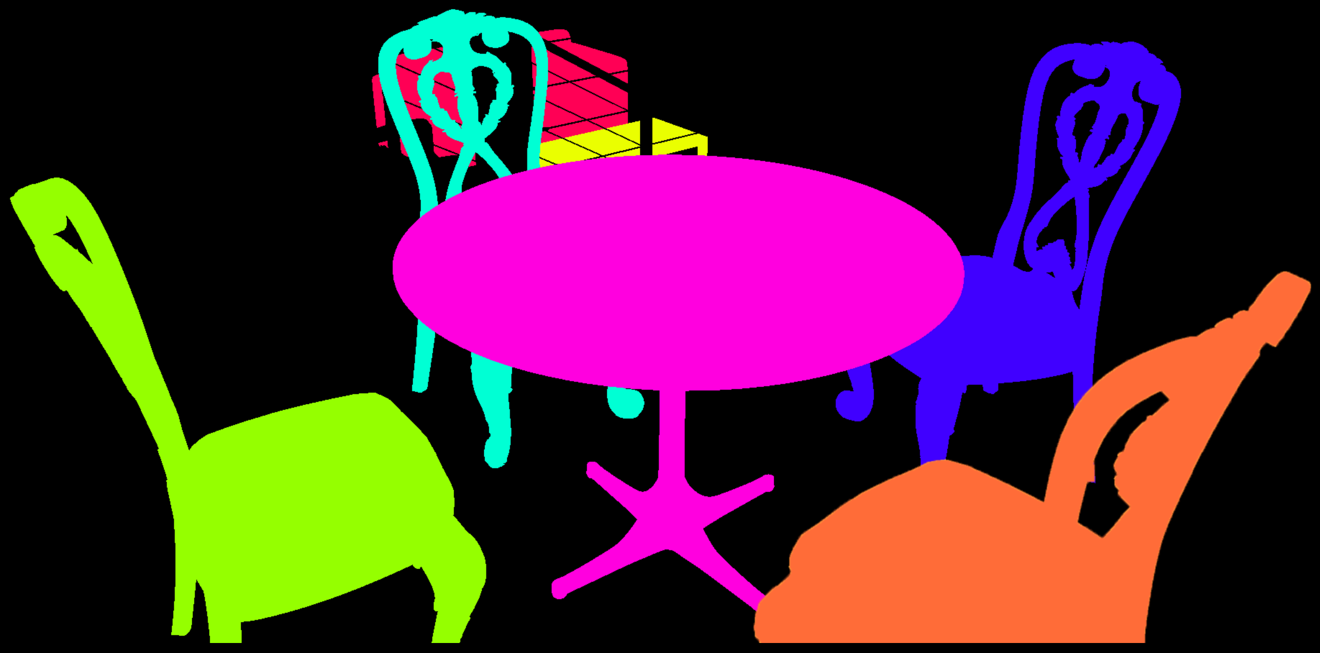
Bounding Box



3D Bounding Box



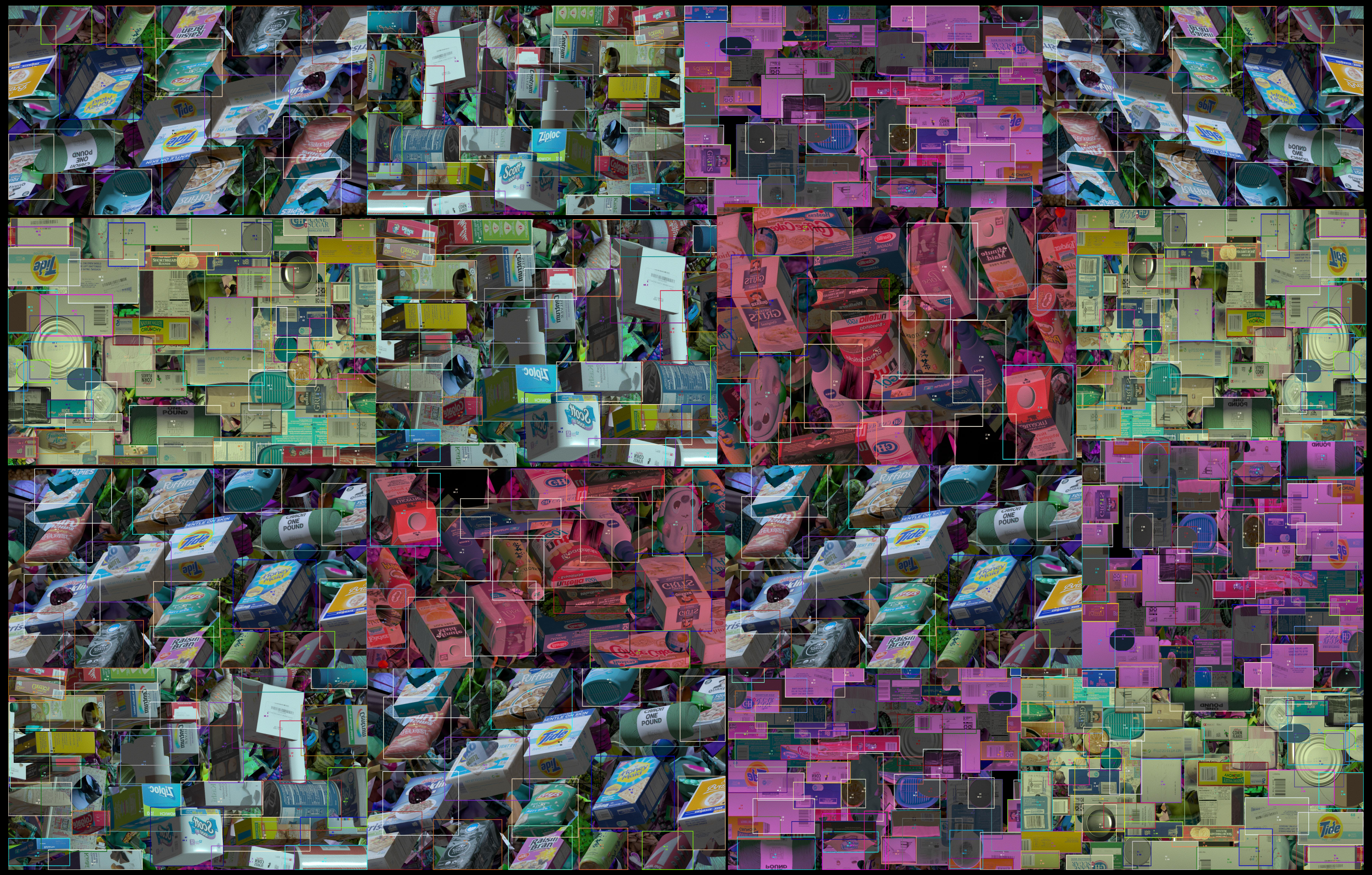
Semantic Segmentation



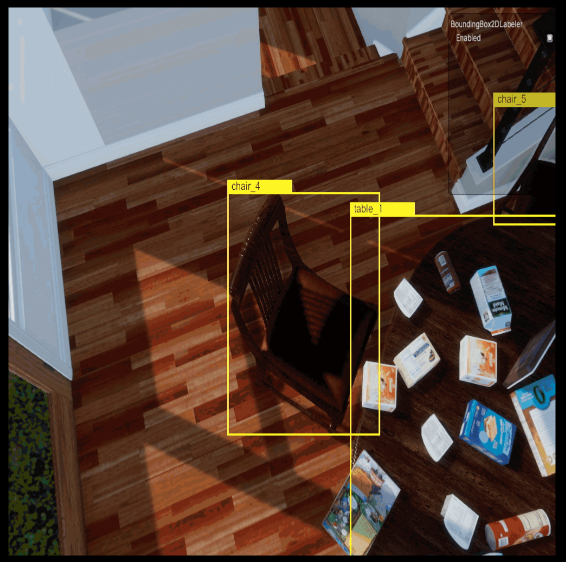
Instance Segmentation

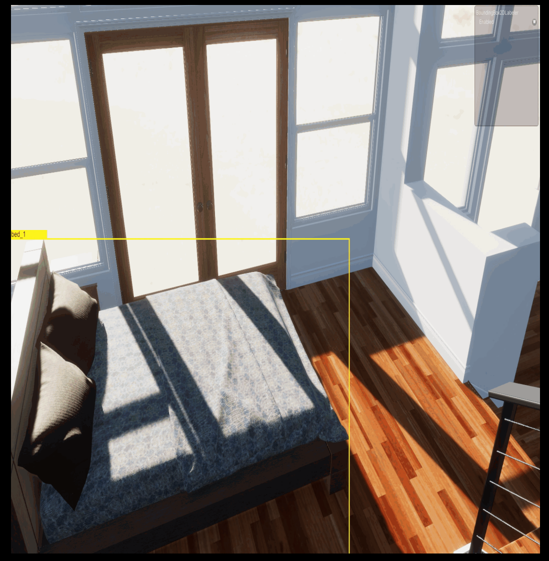
Perception SDK: Extrinsic Randomizations

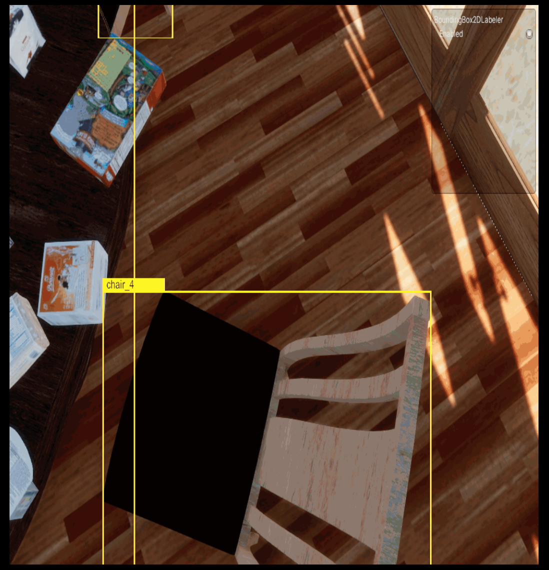
Unstructured / Semi-Structured



Structured

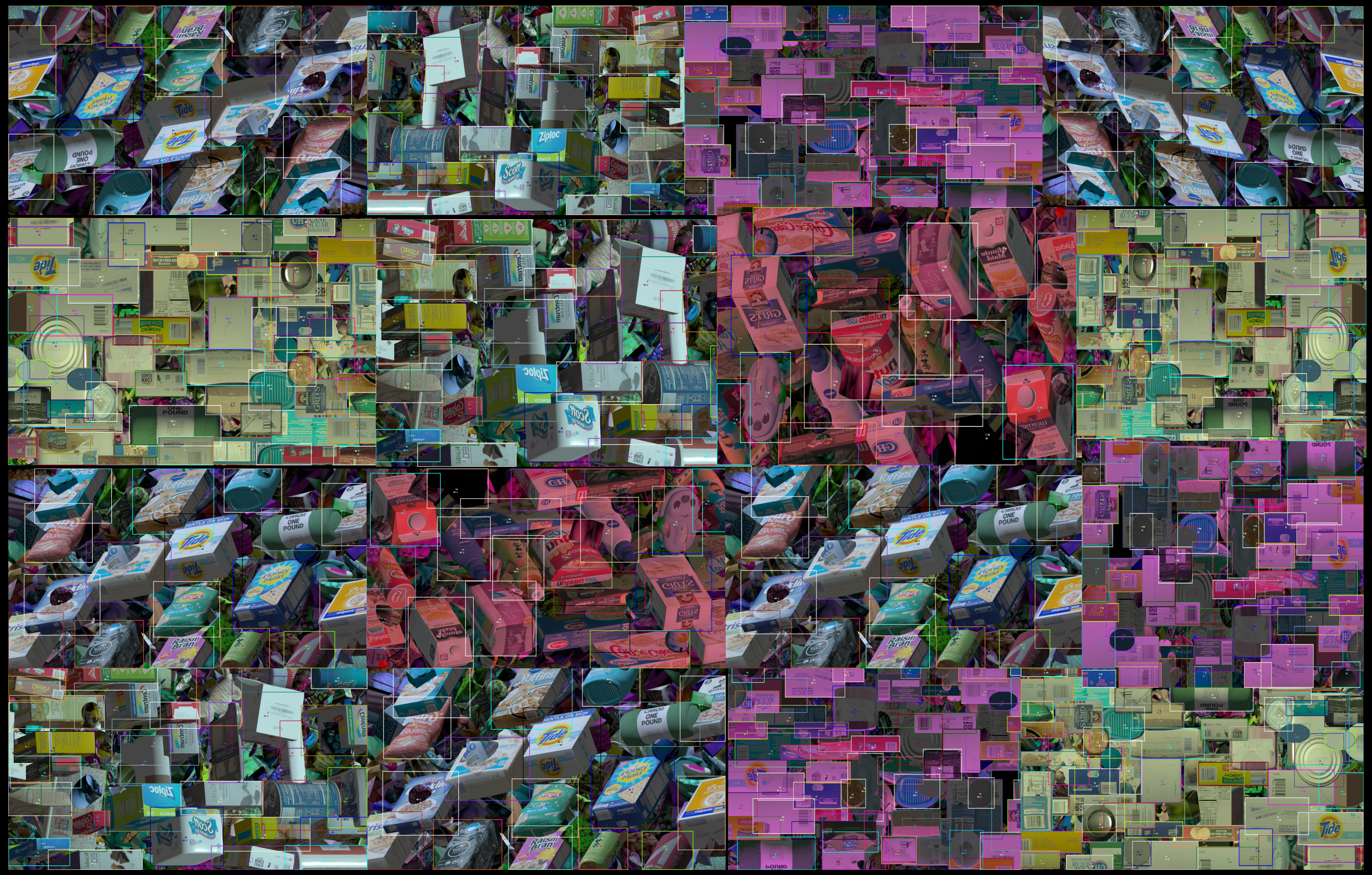




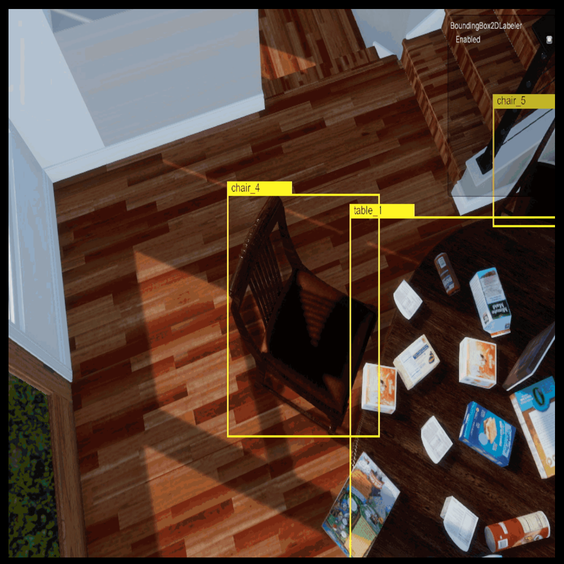


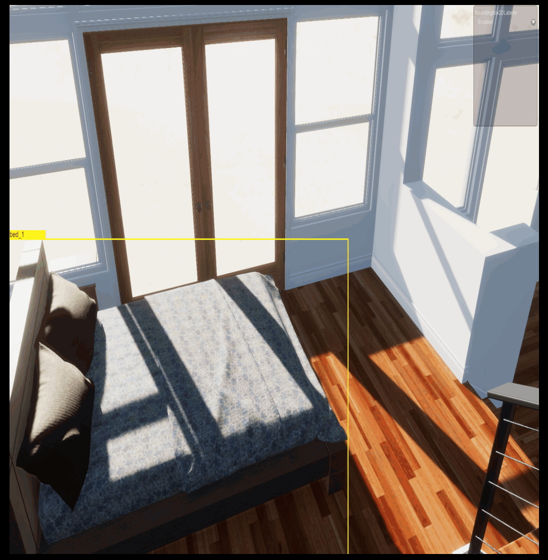
Perception SDK: Extrinsic Randomizations

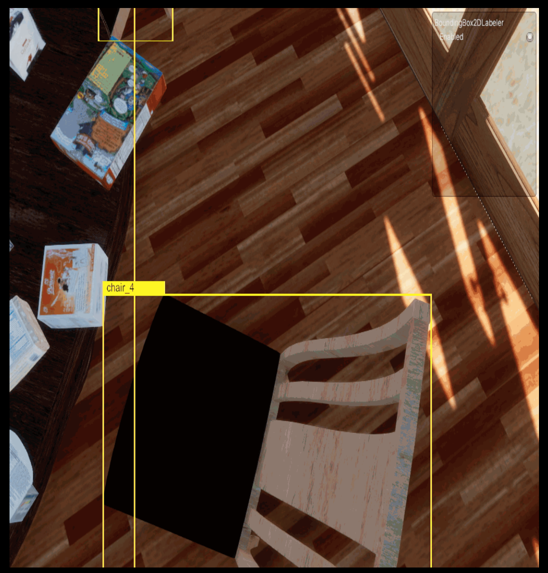
Unstructured / Semi-Structured



Structured



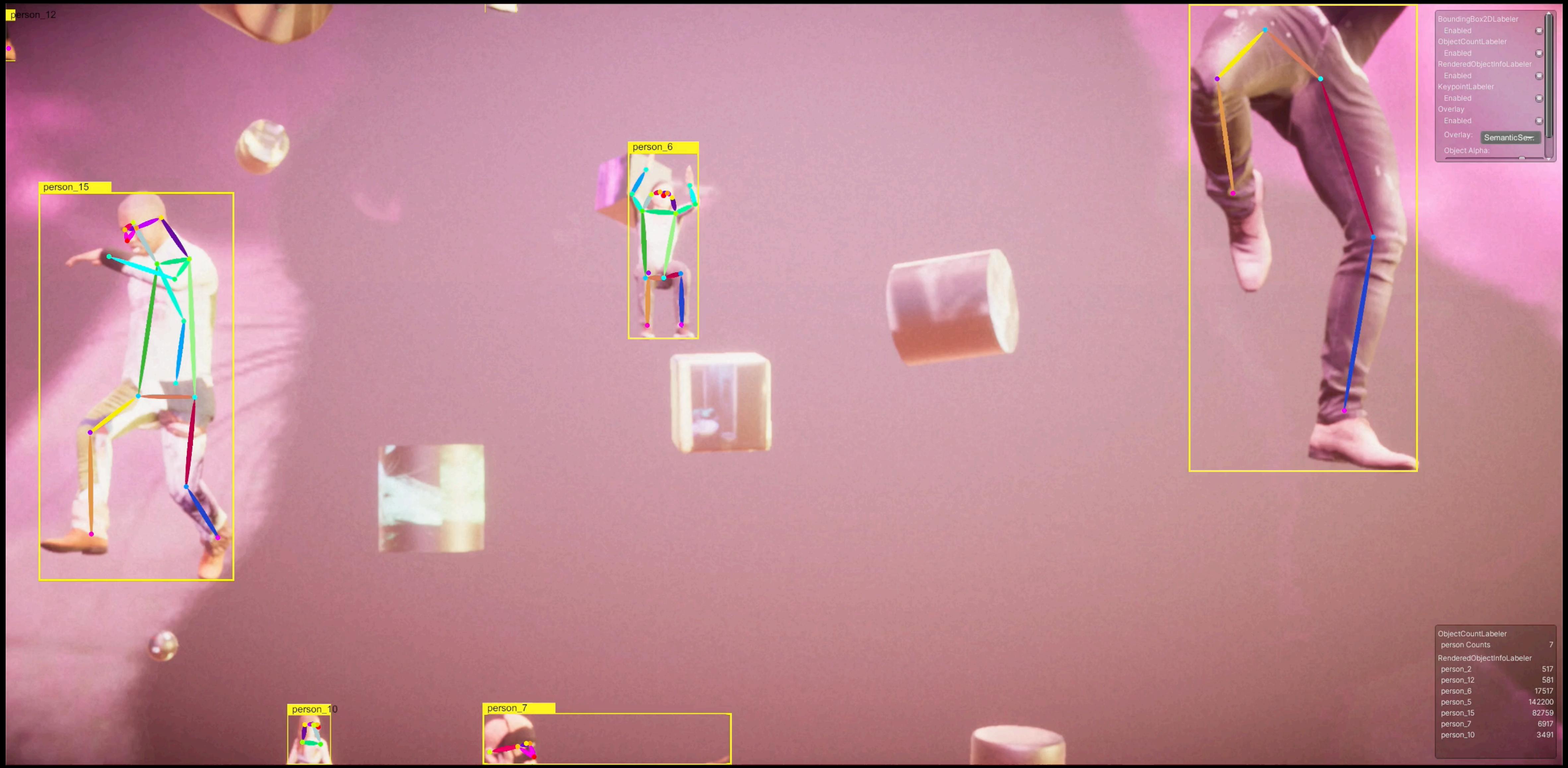




Benchmark Environment of Human Centric Computer Vision

PeopleSansPeople

- 28 Human Assets
- 39 diverse Animations sequences
- 21,952 clothing textures
- Parameterized Placement randomizer
- Parameterized Lighting and Camera System
- Occluders/Distractor objects
- RGB image capture with High Definition Render Pipeline
- Labelers:
  - Bounding Box
  - Semantic Segmentation
  - Instance Segmentation
  - Pose Labeler
    - COCO keypoints
- Packaged macOS and Linux binaries
- CLI + configs to update all parameters



PeopleSansPeople

- 28 Human Assets
- 39 diverse Animations sequences
- 21,952 clothing textures
- Parameterized Placement randomizer
- Parameterized Lighting and Camera System
- Occluders/Distractor objects
- RGB image capture with High Definition Render Pipeline
- Labelers:
  - Bounding Box
  - Semantic Segmentation
  - Instance Segmentation
  - Pose Labeler
    - COCO keypoints
- Packaged macOS and Linux binaries
- CLI + configs to update all parameters

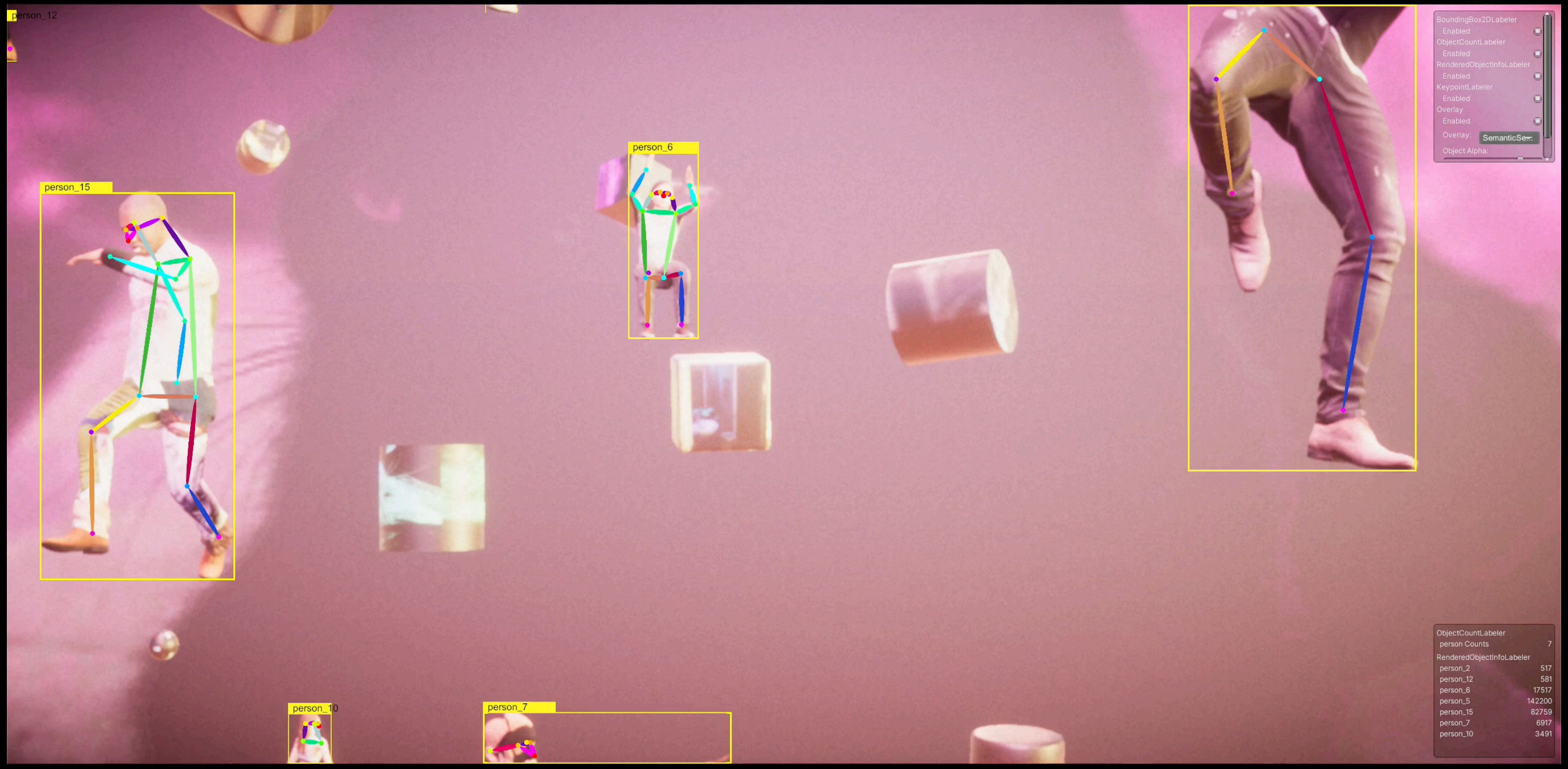
Animation/Pose Randomization

Animation/Pose Randomization

Clothing Texture (Shader-Graph) Randomizer

Clothing Texture (Shader-Graph) Randomizer

PeopleSansPeople



PeopleSansPeople

PeopleSansPeople - Exposed Parameters, Objects

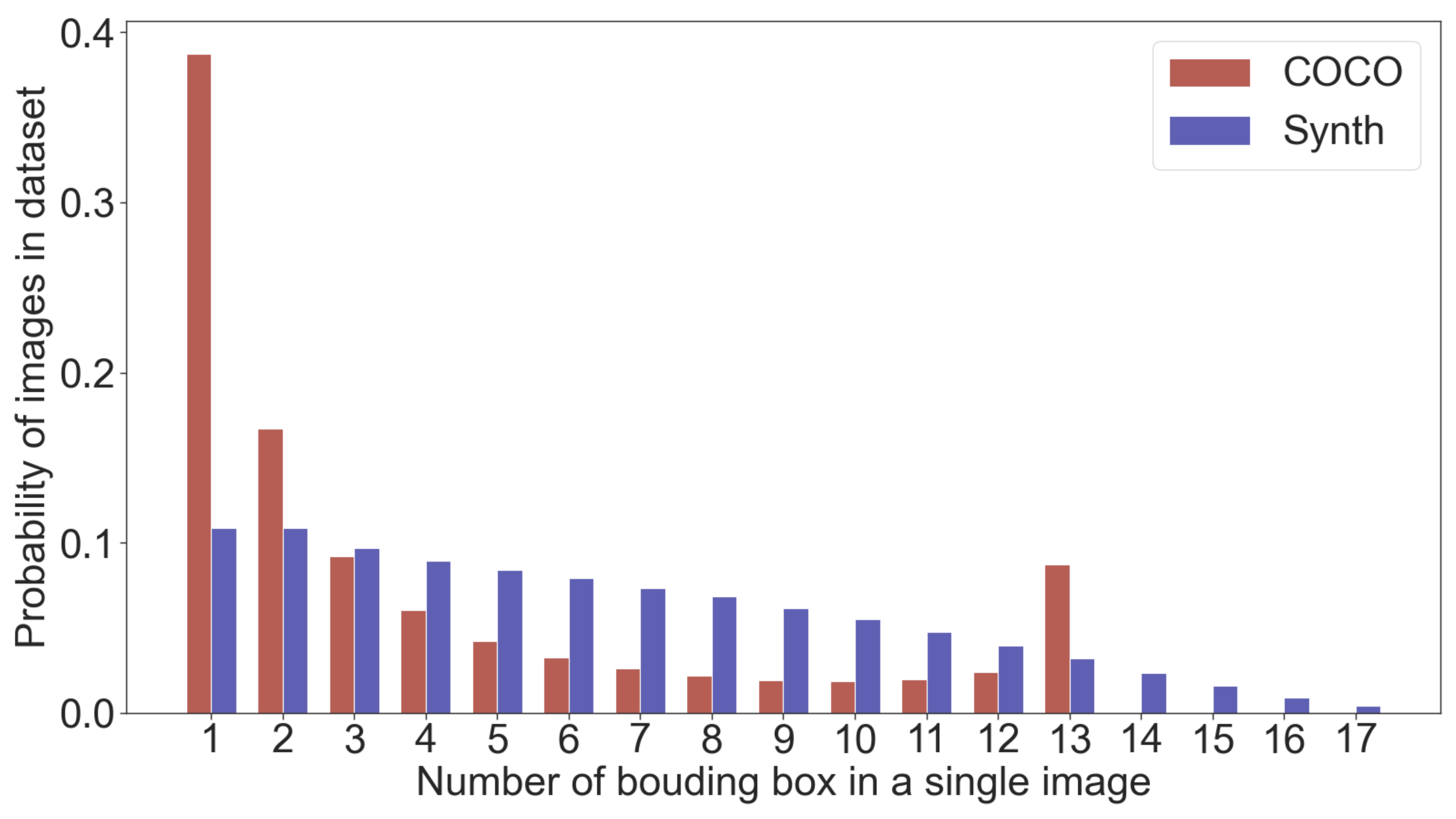
category	randomizer	parameters
3D Objects	Background/Occluder Object Placement	object placement
		separation distance
		object placement offset
	Background/Occluder Scale	
	Background/Occluder Rotation	object rotation
	Foreground Object Placement	object placement
		separation distance
		object placement offset
	Foreground Scale	object scale range
	Foreground Rotation	object rotation
	Animation	animations

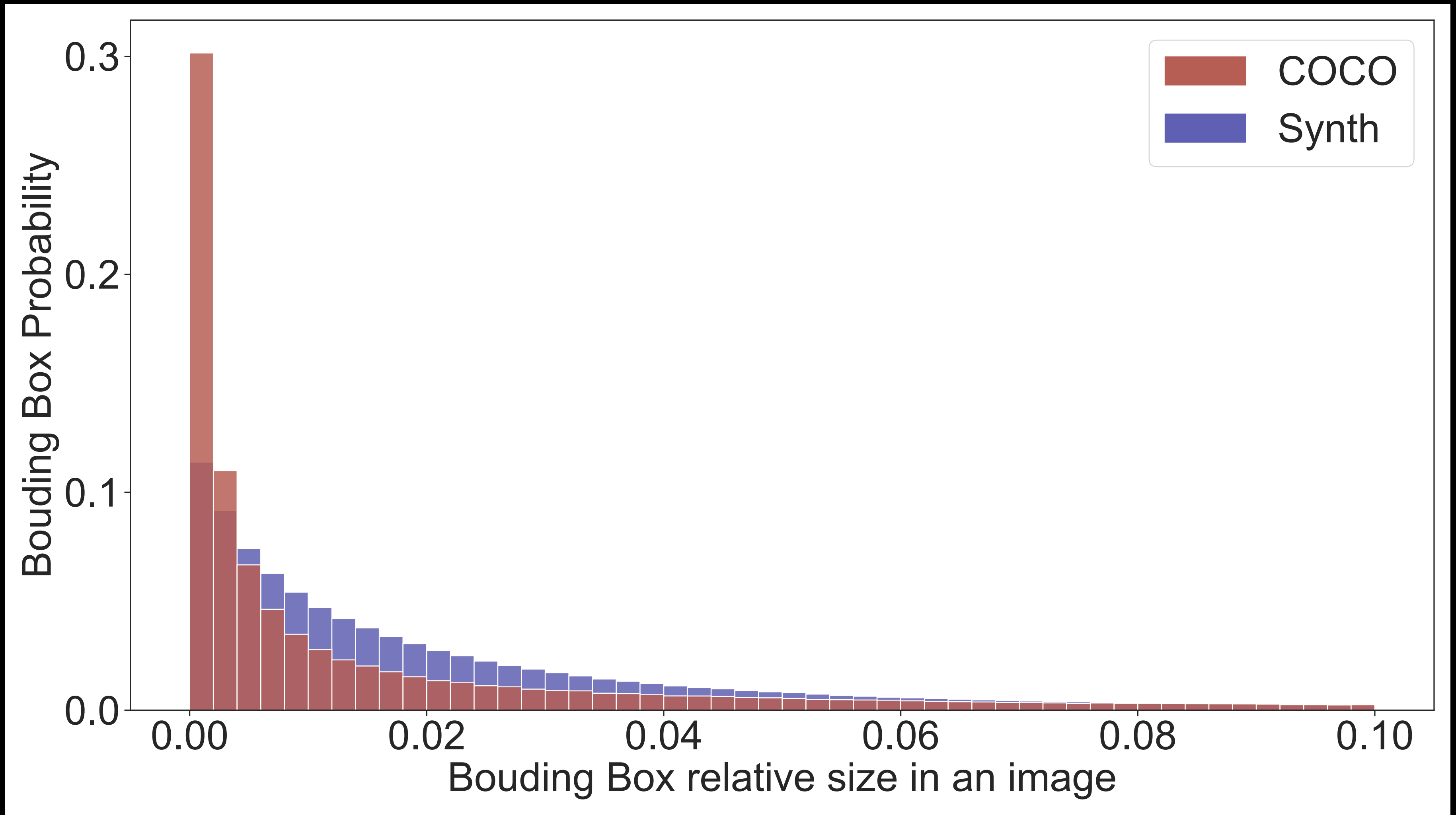
category	randomizer	parameters
Textures and Colours	Texture	textures
	Hue Offset	hue offset
	Shader Graph Texture	albedo textures
		normal textures
		mask textures
		materials
		hue top clothing
		hue bottom clothing

PeopleSansPeople - Exposed Parameters, Rendering

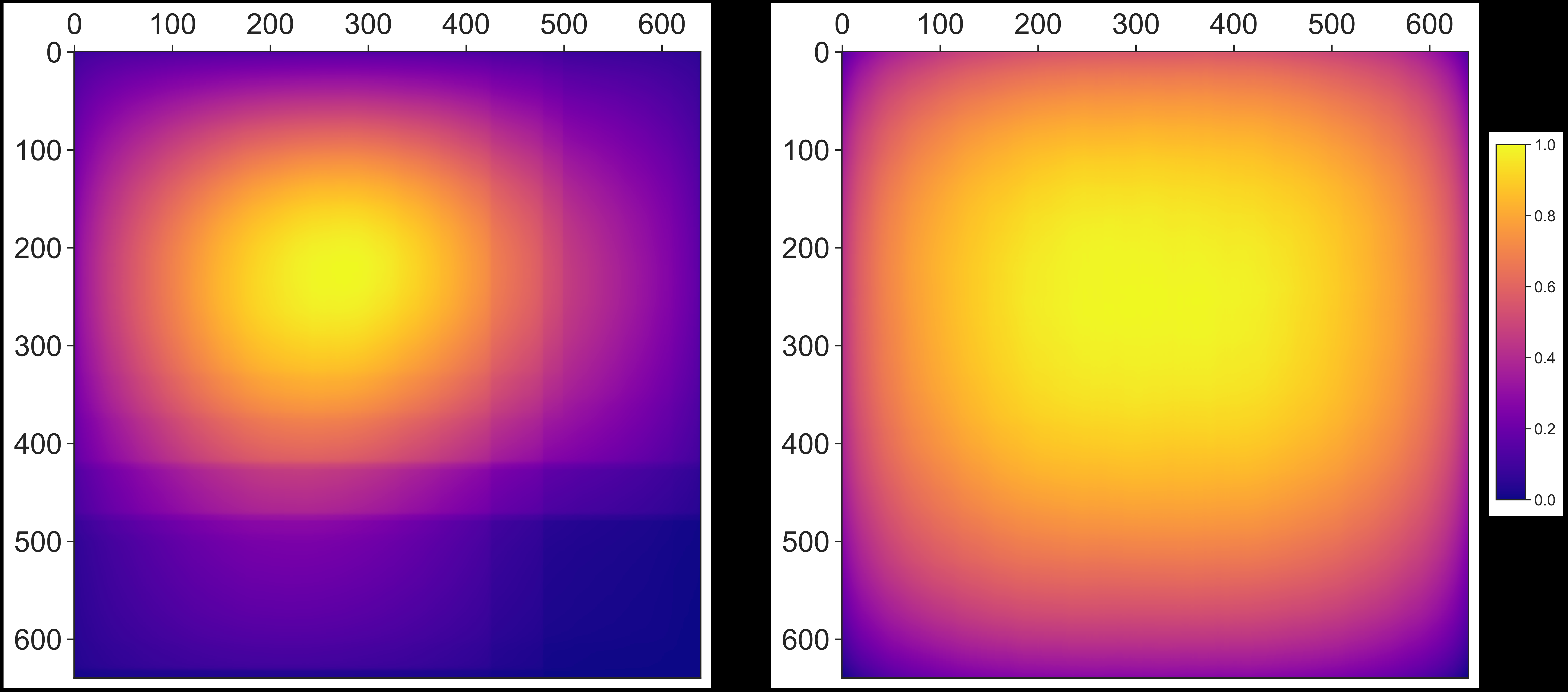
category	randomizer	parameters
Lights	Sun Angle	hour
		day of the year
		lattitude
	Light Intensity and Colour	intensity
		colour
		light switcher enabled probability
	Light Position and Rotation	position offset from initial position
		rotation offset from initial rotation
Camera	Camera	field of view
		focal length
		position offset from initial position
		rotation offset from initial rotation
Post-Processing	Post Process Volume	vignette intensity
		fixed exposure
		white balance temperature
		depth of field focus distance
		colour adjustments: contrast
		colour adjustments: saturation

Controllable Number and Size of People





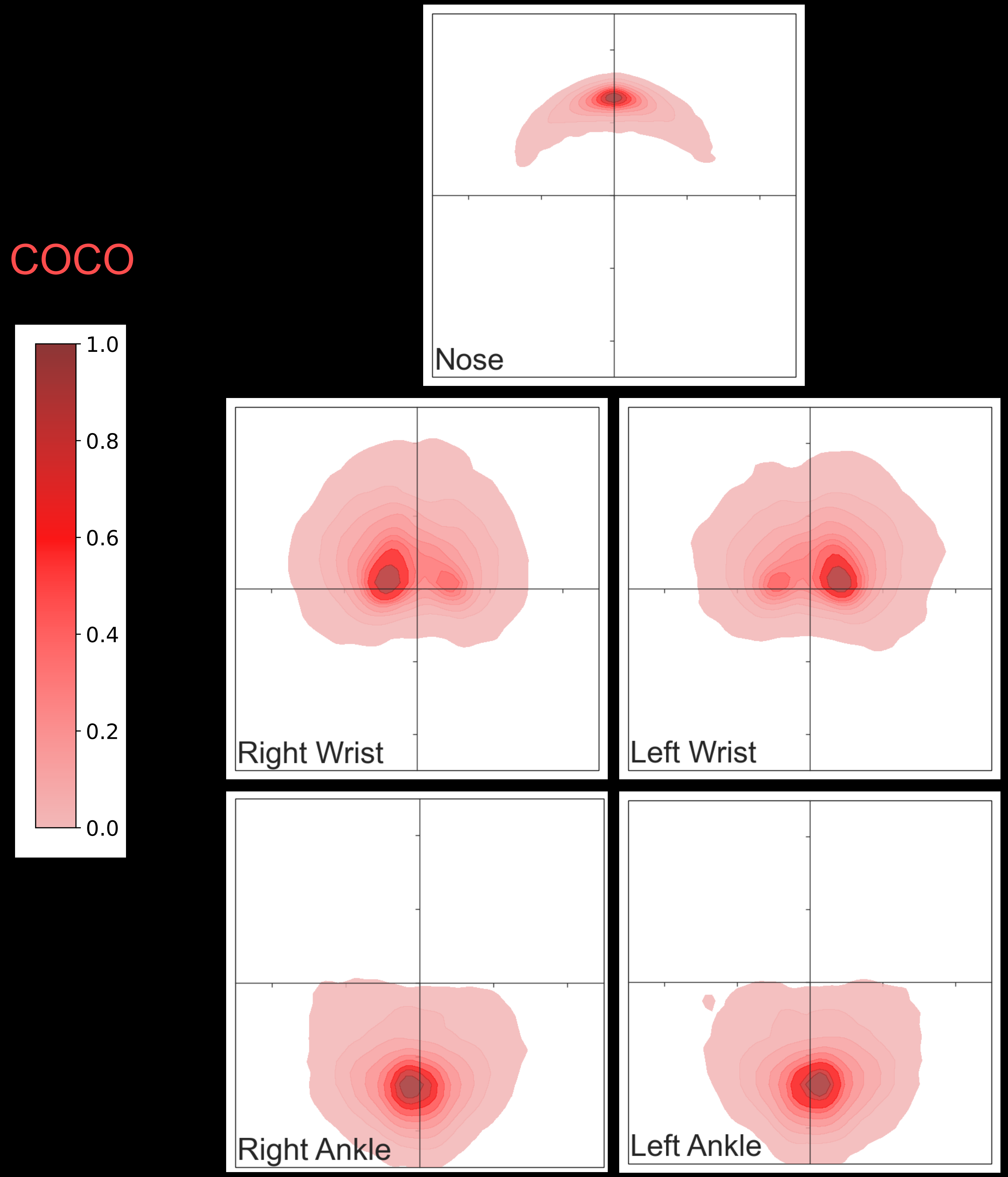
Controllable Placement of People

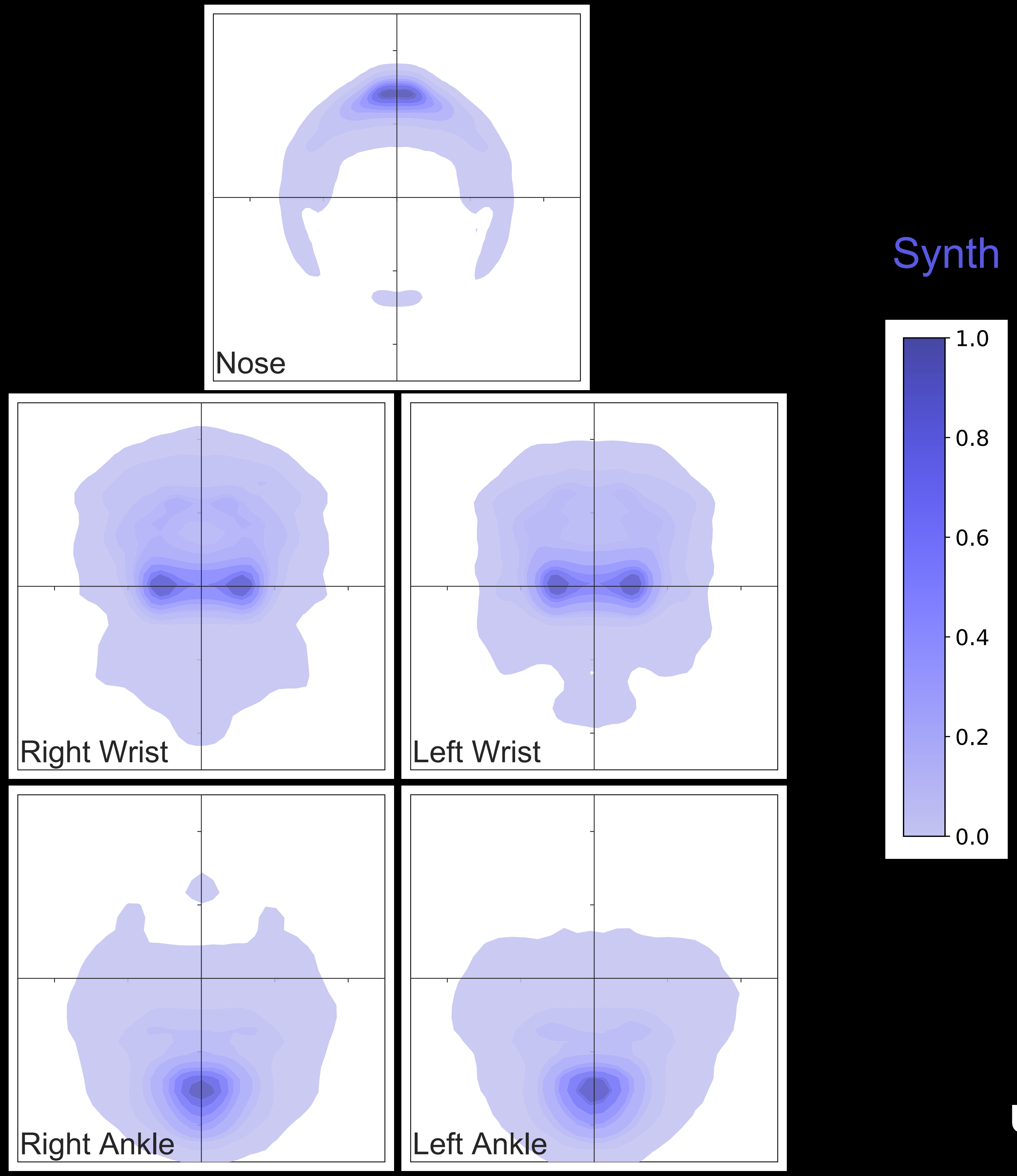


COCO

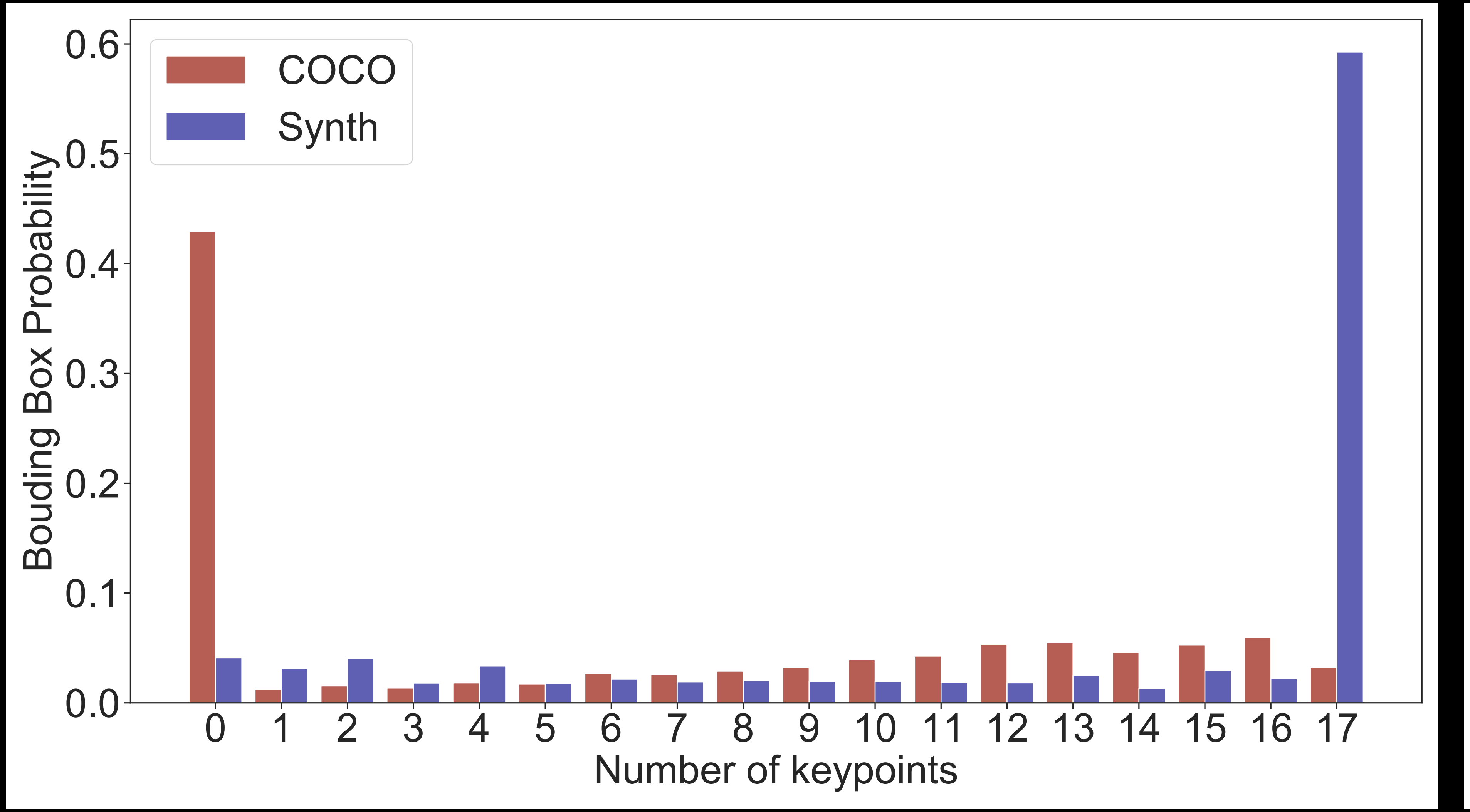
Synth

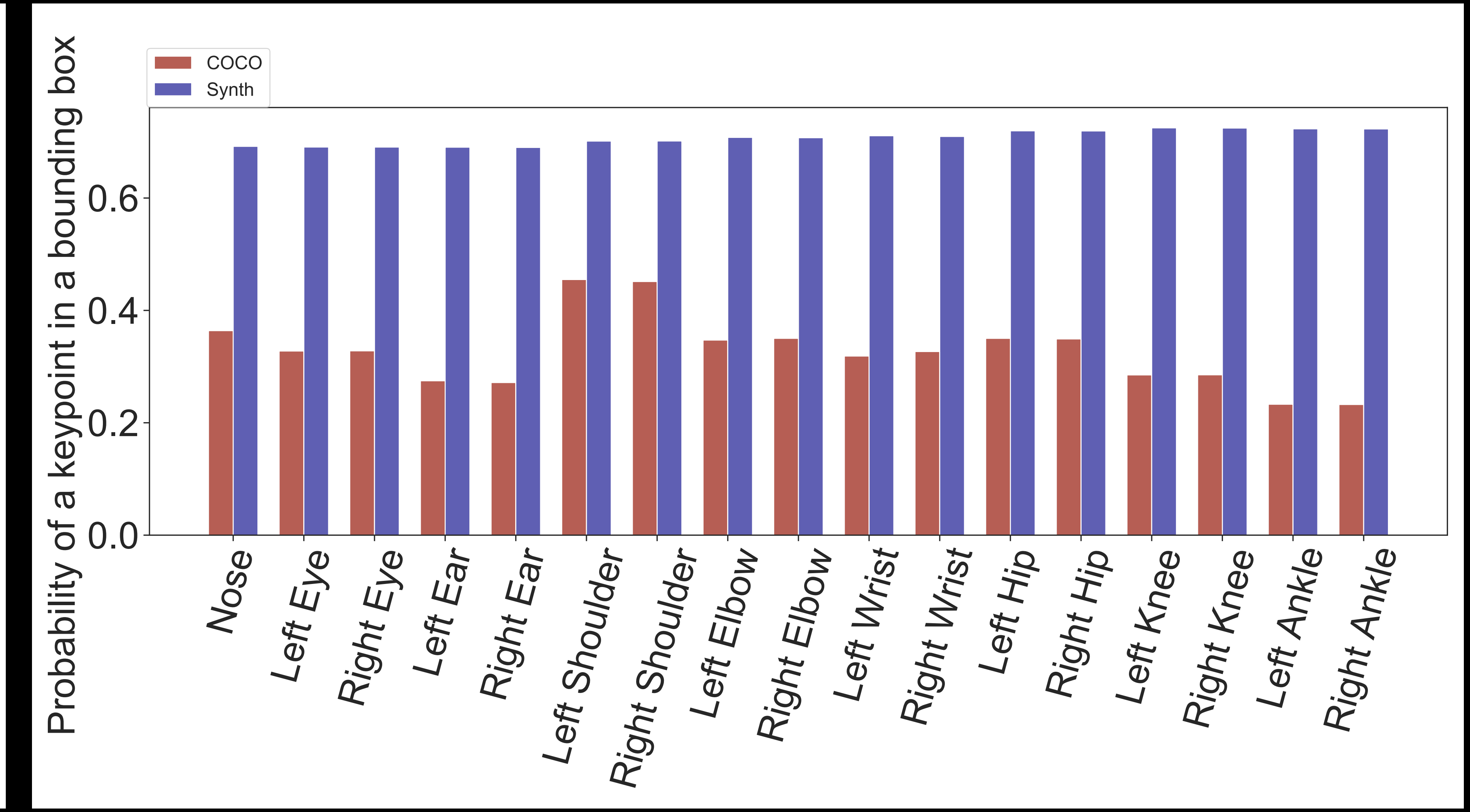
Enhanced Pose Diversity





Improved Label Consistency

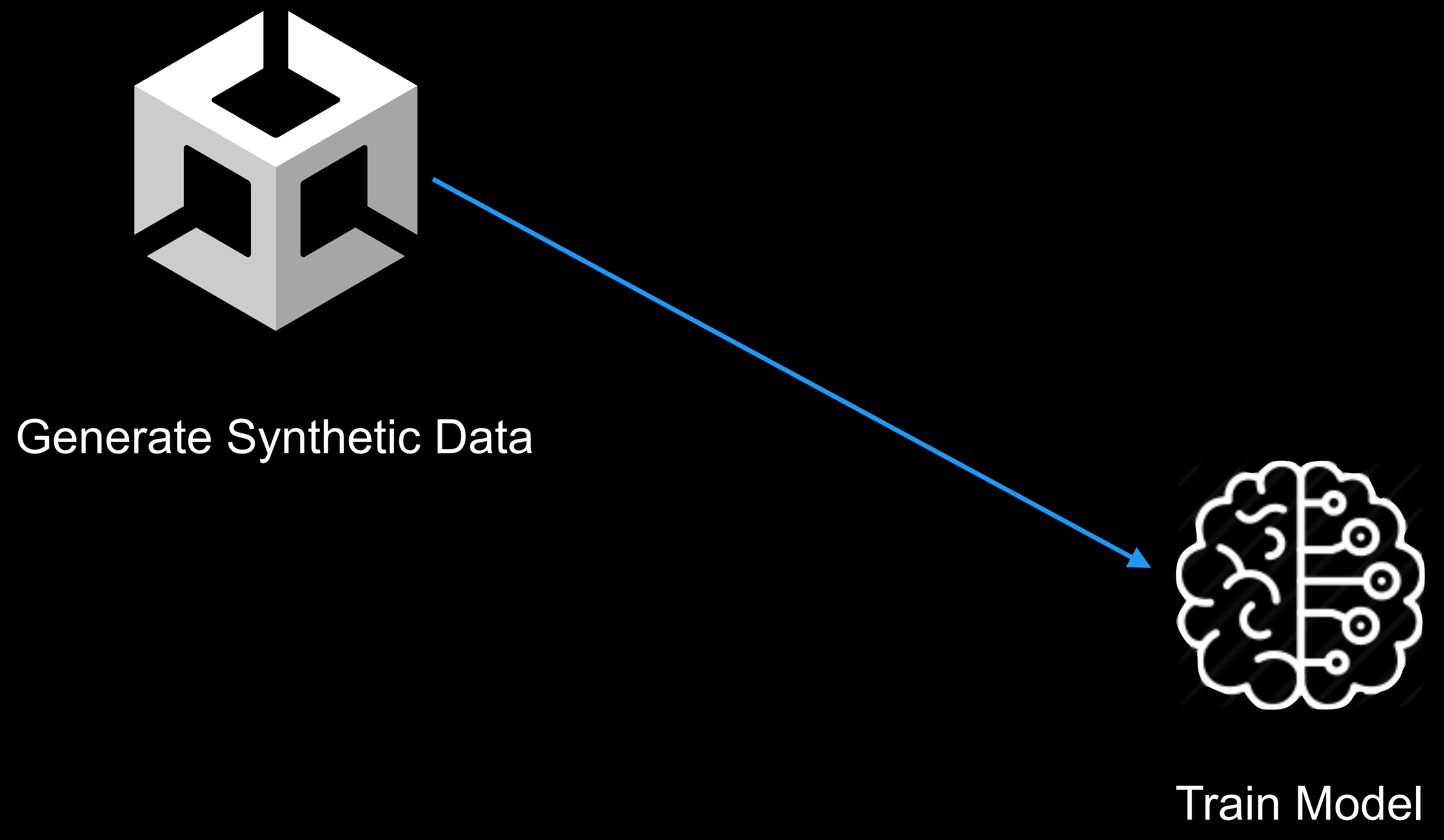




Baseline Training Method

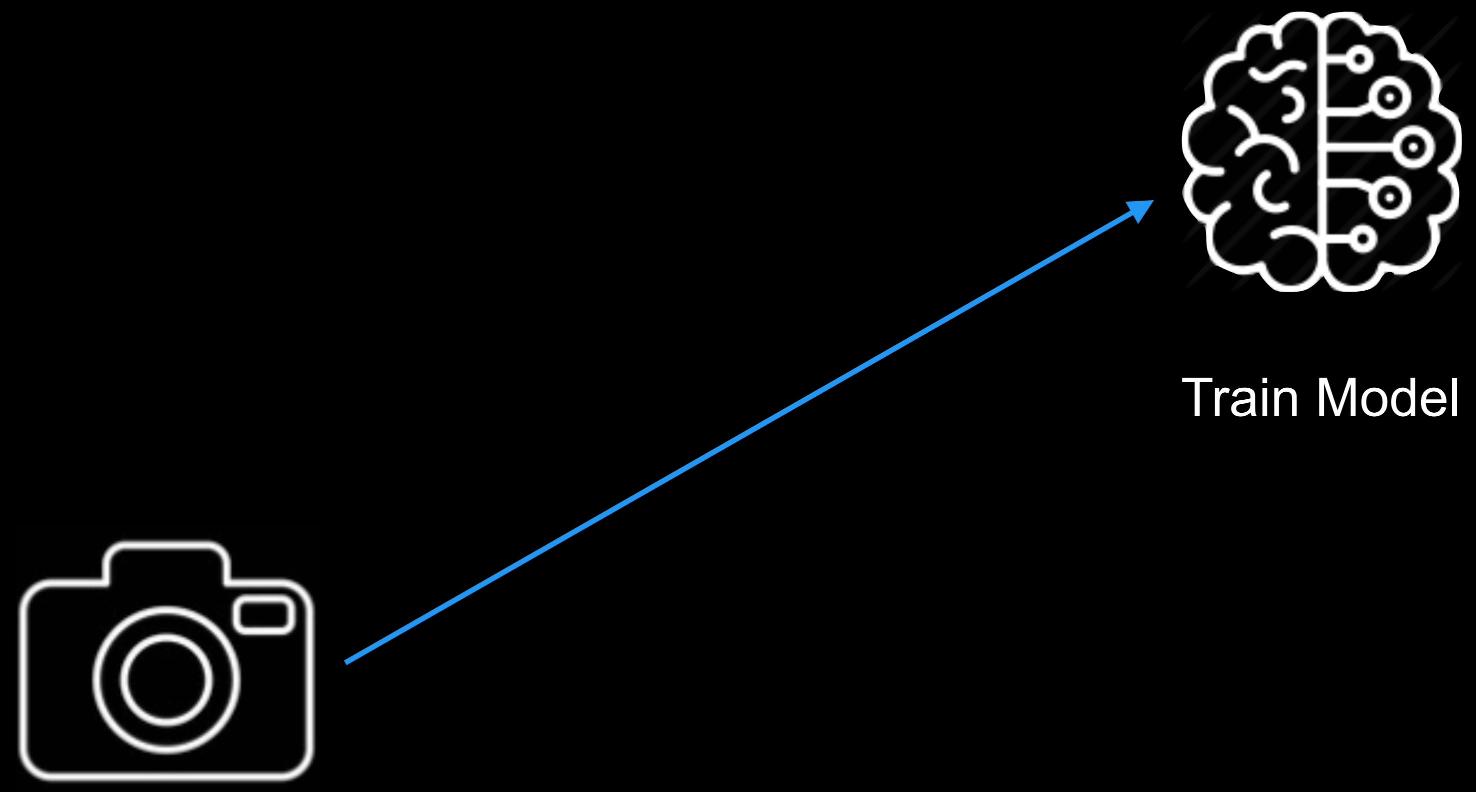
Baseline Training Method

- Generate Data from PeopleSansPeople
  - No data hyperparameter tuning
- Train model on synthetic only



Baseline Training Method

- Generate Data from PeopleSansPeople
  - No data hyperparameter tuning
- Train model on synthetic only
- Fine-tune model on target real data (COCO)
  - No weight freezing
- Evaluate on COCO test-dev2017



Improved Model Performance

Bounding Box Average Precision

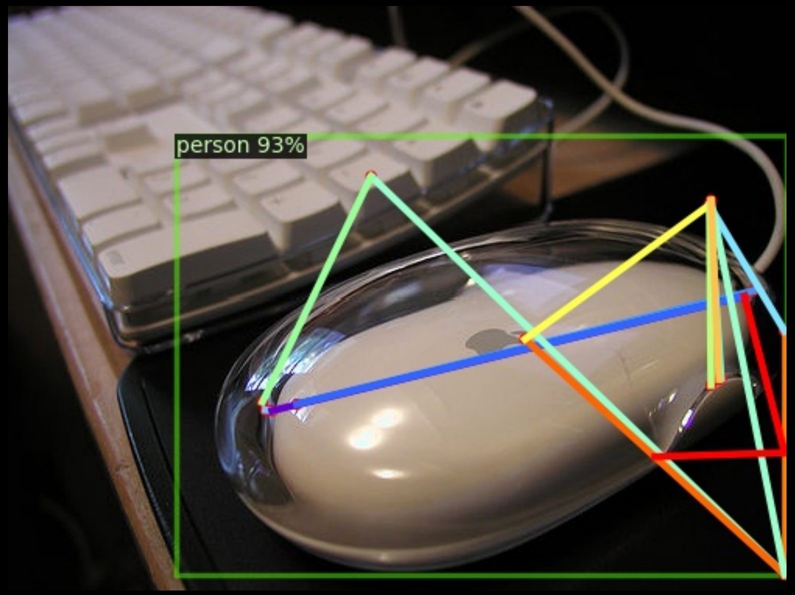
Real Data Size (COCO)	Train from scratch	ImageNet pre-training	Synthetic pre-training(490,000 frames)
641	13.82	27.61	41.24 ± 2.07
6411	37.82	42.53	48.97 ± 0.17
32057	52.15	52.75	54.93 ± 0.15
64115	56.73	56.09	57.44 ± 0.11

Keypoint Average Precision

Real Data Size (COCO)	Train from scratch	ImageNet pre-training	Synthetic pre-training(490,000 frames)
641	6.40	21.90	42.93 ± 2.80
6411	37.30	44.20	52.70 ± 0.36
32057	55.80	57.50	60.37 ± 0.48
64115	62.00	62.40	63.47 ± 0.19

Improved Model Performance - 6411 COCO images

ImageNet Pre-training



Synthetic Pre-training

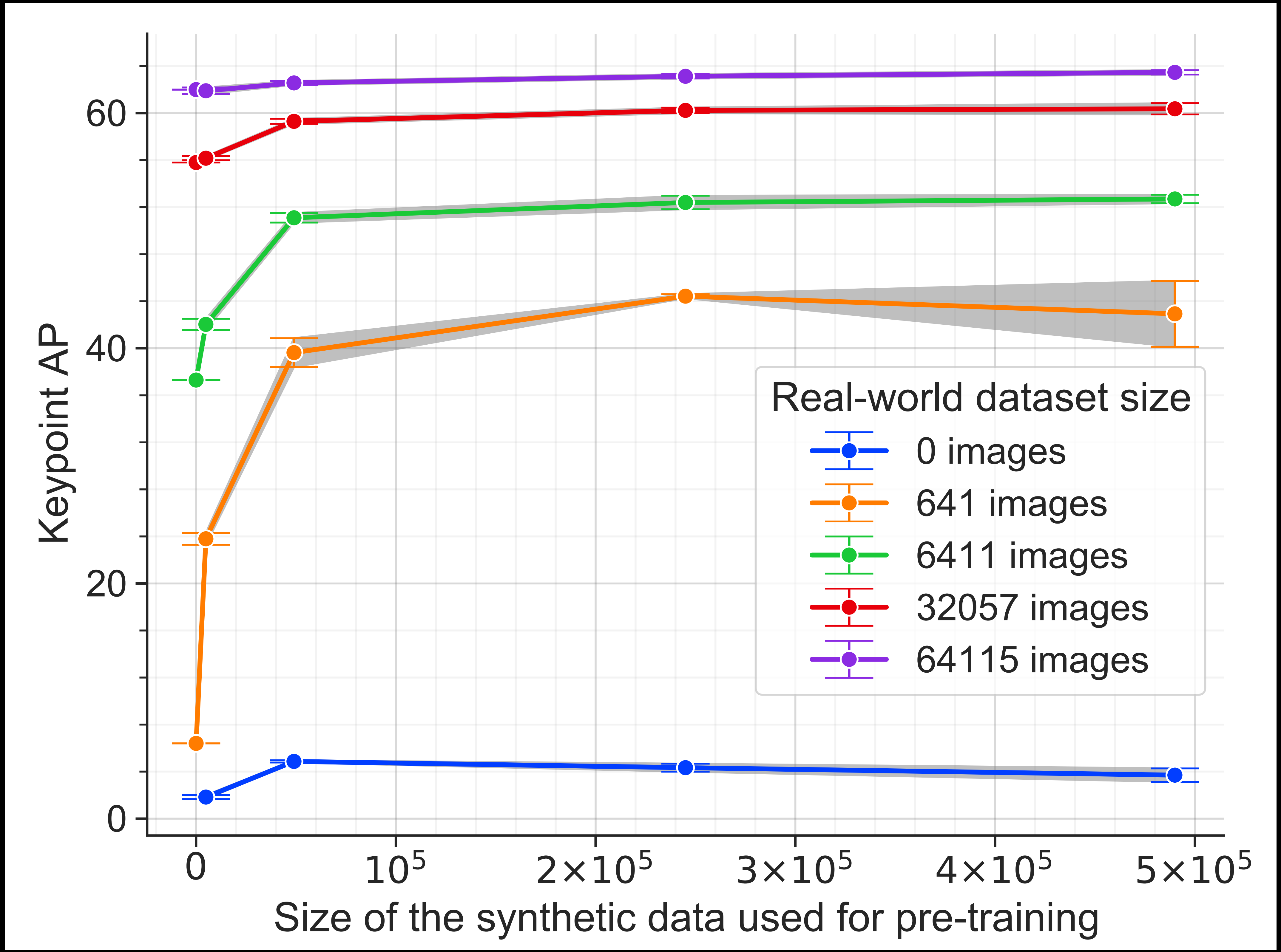
Improved Model Performance - 6411 COCO images

ImageNet Pre-training

Synthetic Pre-training

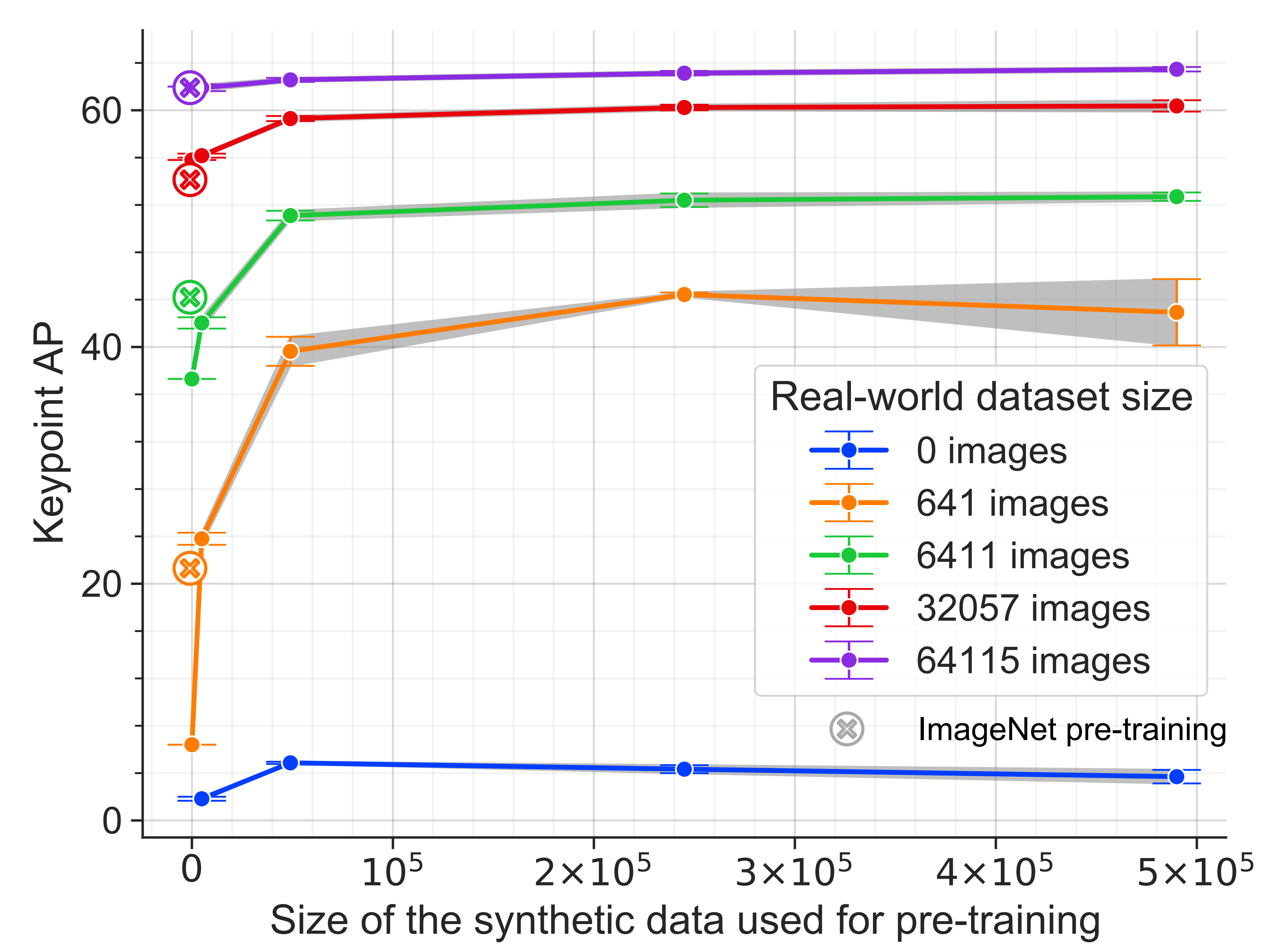
Improved Model Performance

- Pre-train Detectron2 (KeyPoint-RCNN) on synthetic data
- Fine-tuning performance improves with size of synthetic data
- Poor Zero-Shot performance with wildly randomized data



Improved Model Performance

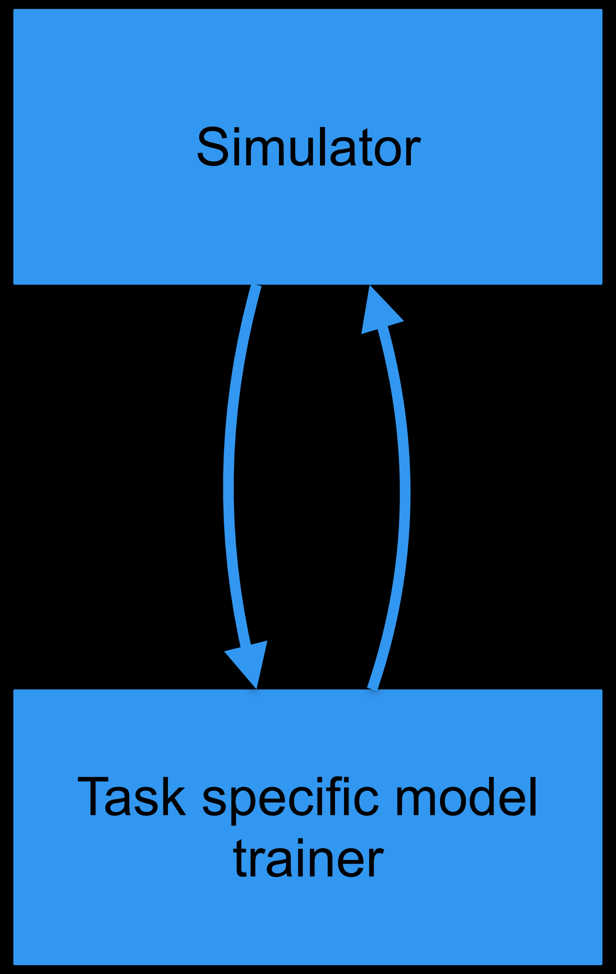
- Pre-train Detectron2 (KeyPoint-RCNN) on synthetic data
- Fine-tuning performance improves with size of synthetic data
- Poor Zero-Shot performance with wildly randomized data



Meta Learning to control the Simulation Parameters

Meta Learning to control the Simulation Parameters

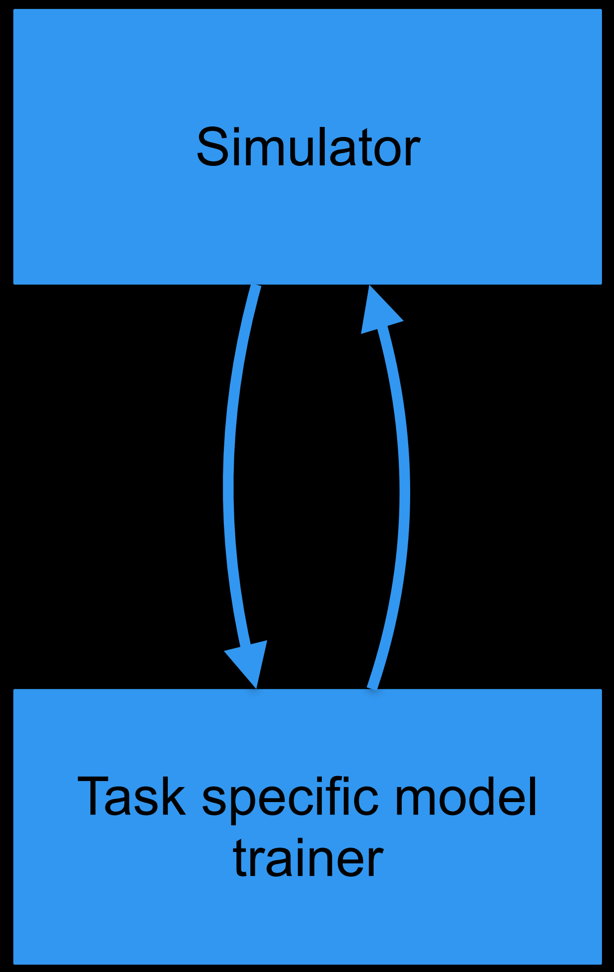
Can we learn the parameters of the simulation to optimize model performance on in the real world?



Meta Learning to control the Simulation Parameters

Can we learn the parameters of the simulation to optimize model performance on in the real world?

- Automatic[1] and Adaptive[2] and Active[3] Domain Randomization

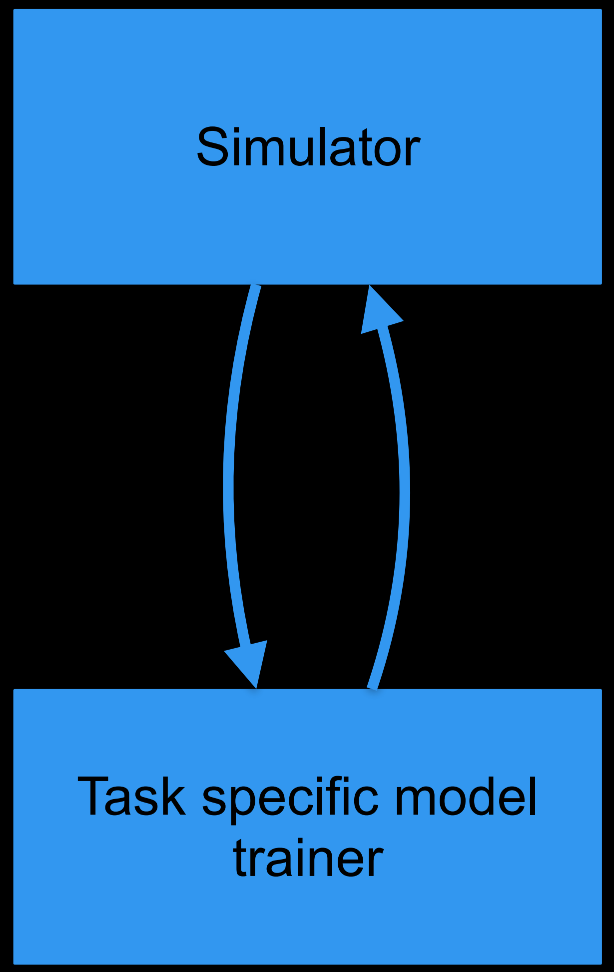


[1]Akkaya, I., et al. "Solving Rubik's cube with a robot hand." arXiv preprint arXiv:1910.07113 (2019).
[2]Ramos F., et al. “BayesSim: adaptive domain randomization via probabilistic inference for robotics simulators.” R:SS (2019)
[3]Mehta, B., et al. “Active Domain Randomization.” PMLR (2020).

Meta Learning to control the Simulation Parameters

Can we learn the parameters of the simulation to optimize model performance on in the real world?

- Automatic[1] and Adaptive[2] and Active[3] Domain Randomization
- Meta-Sim[4,5]

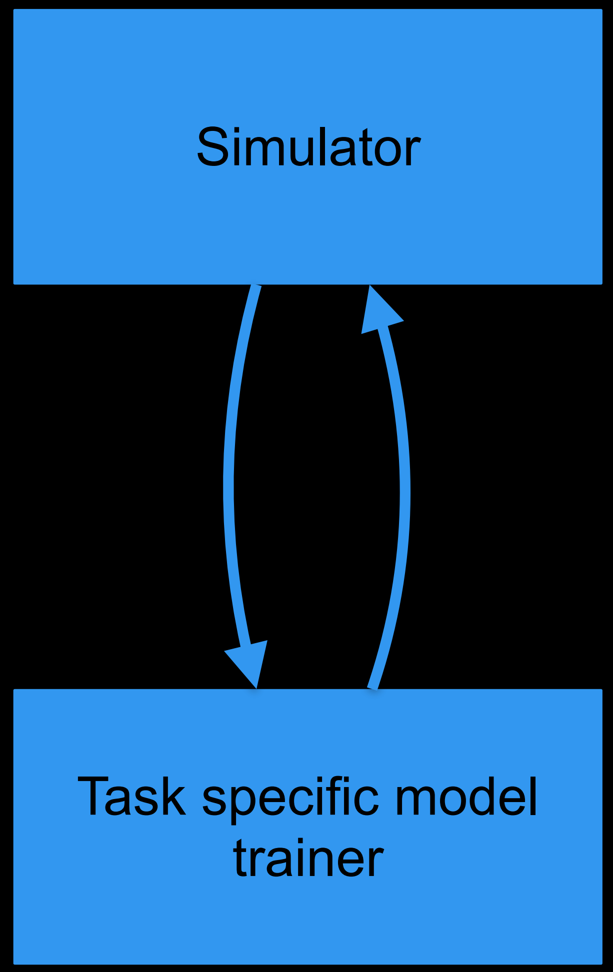


[1]Akkaya, I., et al. "Solving Rubik's cube with a robot hand." arXiv preprint arXiv:1910.07113 (2019).
[2]Ramos F., et al. “BayesSim: adaptive domain randomization via probabilistic inference for robotics simulators.” R:SS (2019)
[3]Mehta, B., et al. “Active Domain Randomization.” PMLR (2020).
[4]Kar, A., et al. Meta-Sim: “Learning to Generate Synthetic Datasets.” ICCV (2019)
[5]Dervaranjan, J., et al. “Meta-Sim2: Unsupervised Learning of Scene Structure for Synthetic Data Generation.” ECCV (2020)

Meta Learning to control the Simulation Parameters

Can we learn the parameters of the simulation to optimize model performance on in the real world?

- Automatic[1] and Adaptive[2] and Active[3] Domain Randomization
- Meta-Sim[4,5]
- Learning to Simulate[6,7]

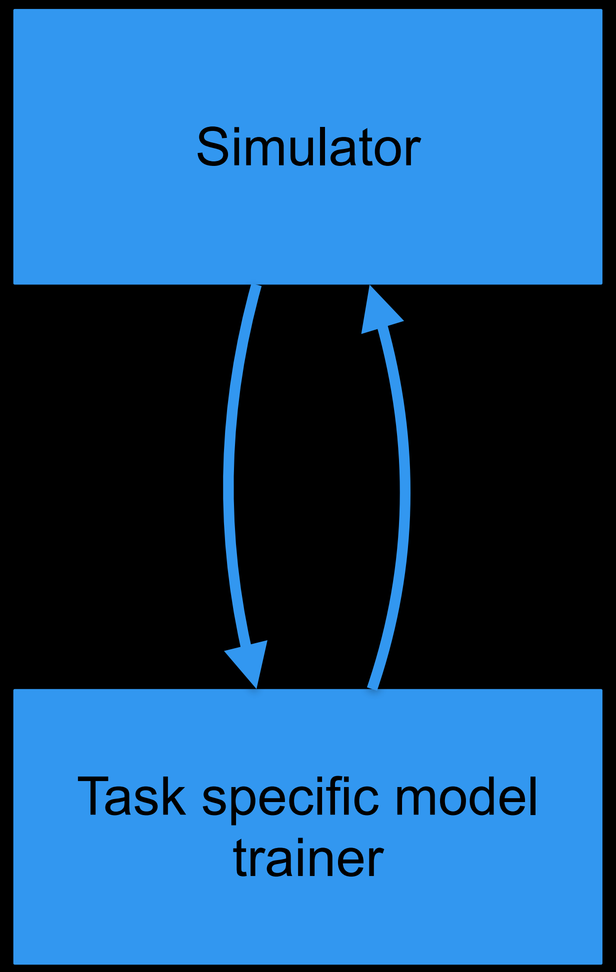


[1] Akkaya, I., et al. "Solving Rubik's cube with a robot hand." arXiv preprint arXiv:1910.07113 (2019).
[2] Ramos F., et al. "BayesSim: adaptive domain randomization via probabilistic inference for robotics simulators." R:SS (2019)
[3] Mehta, B., et al. "Active Domain Randomization." PMLR (2020).
[4] Kar, A., et al. Meta-Sim: "Learning to Generate Synthetic Datasets." ICCV (2019)
[5] Dervaranjan, J., et al. "Meta-Sim2: Unsupervised Learning of Scene Structure for Synthetic Data Generation." ECCV (2020)

Meta Learning to control the Simulation Parameters

Can we learn the parameters of the simulation to optimize model performance on in the real world?

- Automatic[1] and Adaptive[2] and Active[3] Domain Randomization
- Meta-Sim[4,5]
- Learning to Simulate[6,7]



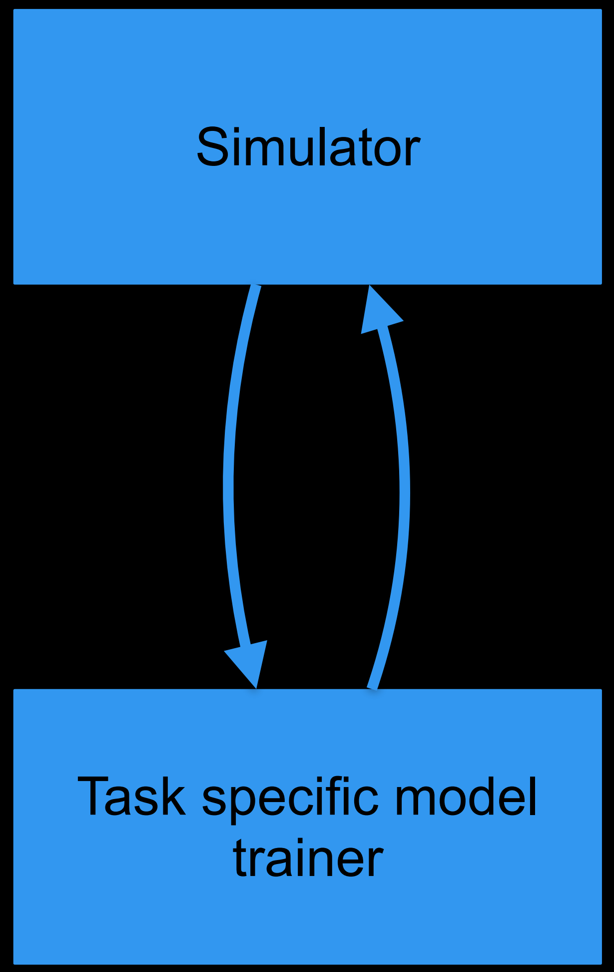
[1]Akkaya, I., et al. "Solving Rubik's cube with a robot hand." arXiv preprint arXiv:1910.07113 (2019).
[2]Ramos F., et al. “BayesSim: adaptive domain randomization via probabilistic inference for robotics simulators.” R:SS (2019)
[3]Mehta, B., et al. “Active Domain Randomization.” PMLR (2020).
[4]Kar, A., et al. Meta-Sim: “Learning to Generate Synthetic Datasets.” ICCV (2019)
[5]Dervaranjan, J., et al. “Meta-Sim2: Unsupervised Learning of Scene Structure for Synthetic Data Generation.” ECCV (2020)

Meta Learning to control the Simulation Parameters

Can we learn the parameters of the simulation to optimize model performance on in the real world?

- Automatic[1] and Adaptive[2] and Active[3] Domain Randomization
- Meta-Sim[4,5]
- Learning to Simulate[6,7]

All of these requires a way to programmatically update the simulation parameters.



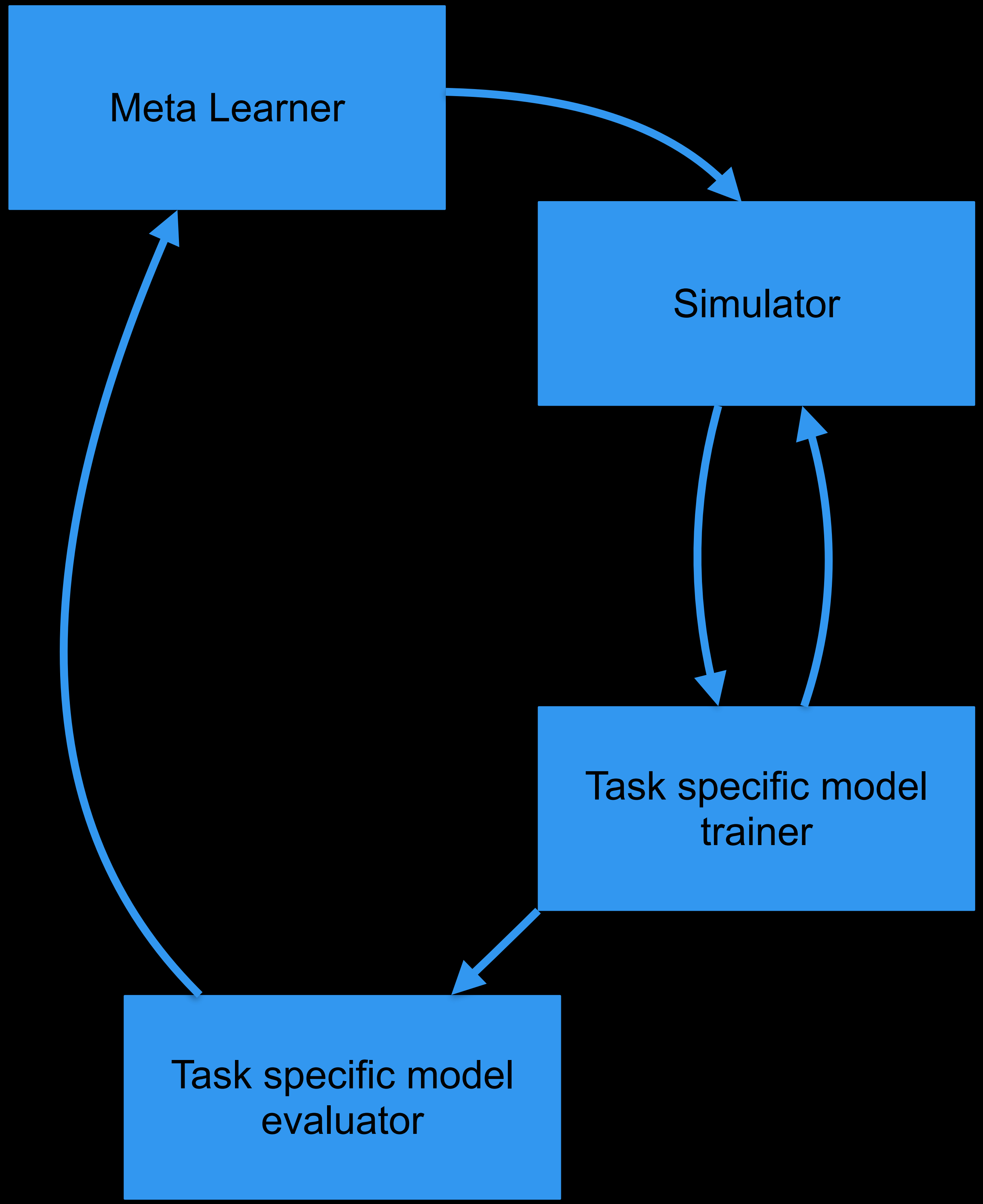
[1]Akkaya, I., et al. "Solving Rubik's cube with a robot hand." arXiv preprint arXiv:1910.07113 (2019).
[2]Ramos F., et al. “BayesSim: adaptive domain randomization via probabilistic inference for robotics simulators.” R:SS (2019)
[3]Mehta, B., et al. “Active Domain Randomization.” PMLR (2020).
[4]Kar, A., et al. Meta-Sim: “Learning to Generate Synthetic Datasets.” ICCV (2019)
[5]Dervaranjan, J., et al. “Meta-Sim2: Unsupervised Learning of Scene Structure for Synthetic Data Generation.” ECCV (2020)
[6]Ruiz, N., et al. “Learning to Simulate.” ICLR (2019)
[7]Behl H.S., et al. “AutoSimulate: (Quickly) Learning Synthetic Data Generation.” ECCV (2020).

Meta Learning to control the Simulation Parameters

Can we learn the parameters of the simulation to optimize model performance on in the real world?

- Automatic[1] and Adaptive[2] and Active[3] Domain Randomization
- Meta-Sim[4,5]
- Learning to Simulate[6,7]

All of these requires a way to programmatically update the simulation parameters.



[1]Akkaya, I., et al. "Solving Rubik's cube with a robot hand." arXiv preprint arXiv:1910.07113 (2019).
[2]Ramos F., et al. “BayesSim: adaptive domain randomization via probabilistic inference for robotics simulators.” R:SS (2019)
[3]Mehta, B., et al. “Active Domain Randomization.” PMLR (2020).
[4]Kar, A., et al. Meta-Sim: “Learning to Generate Synthetic Datasets.” ICCV (2019)
[5]Dervaranjan, J., et al. “Meta-Sim2: Unsupervised Learning of Scene Structure for Synthetic Data Generation.” ECCV (2020)
[6]Ruiz, N., et al. “Learning to Simulate.” ICLR (2019)
[7]Behl H.S., et al. “AutoSimulate: (Quickly) Learning Synthetic Data Generation.” ECCV (2020).

Structured Randomizations - Residential Interiors

- Complete project including 8 full houses, apartments, and townhomes
- Fully furnished and lit from an extensive content library
- Ready for domain randomization:
  - Split Grammar system for furniture, decor, and clutter placement
  - Procedural materials and objects to change room appearance
  - Multiple lighting scenarios and randomized daylight conditions
- All objects are physics-ready for interaction

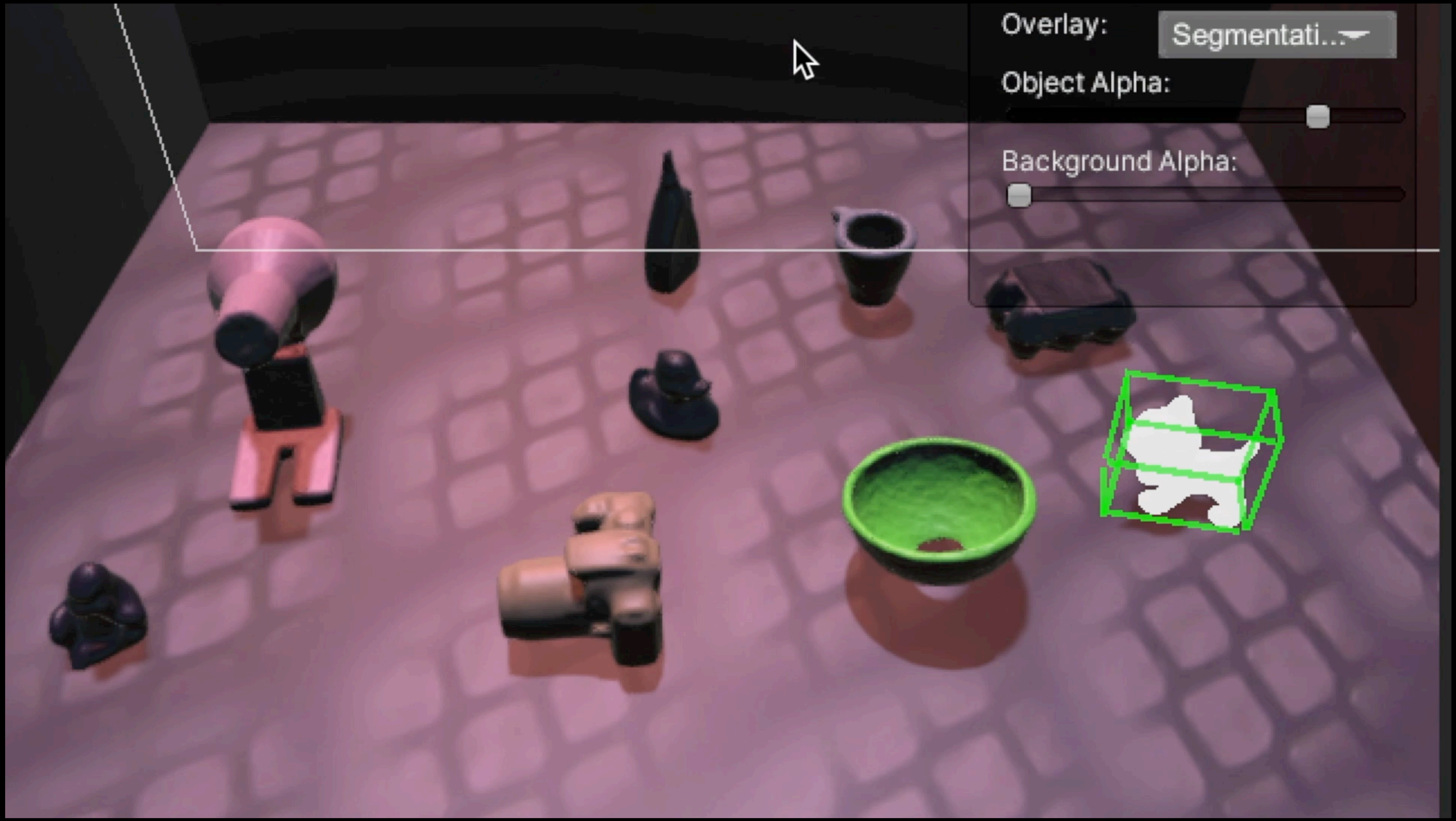

Overlay
Enabled
Overlay:
InstanceSe...
Object Alpha:
Background Alpha:


Overlay
Enabled
Overlay:
InstanceSe...
Object Alpha:
Background Alpha:


Overlay
Enabled
Overlay:
InstanceSe...
Object Alpha:
Background Alpha:

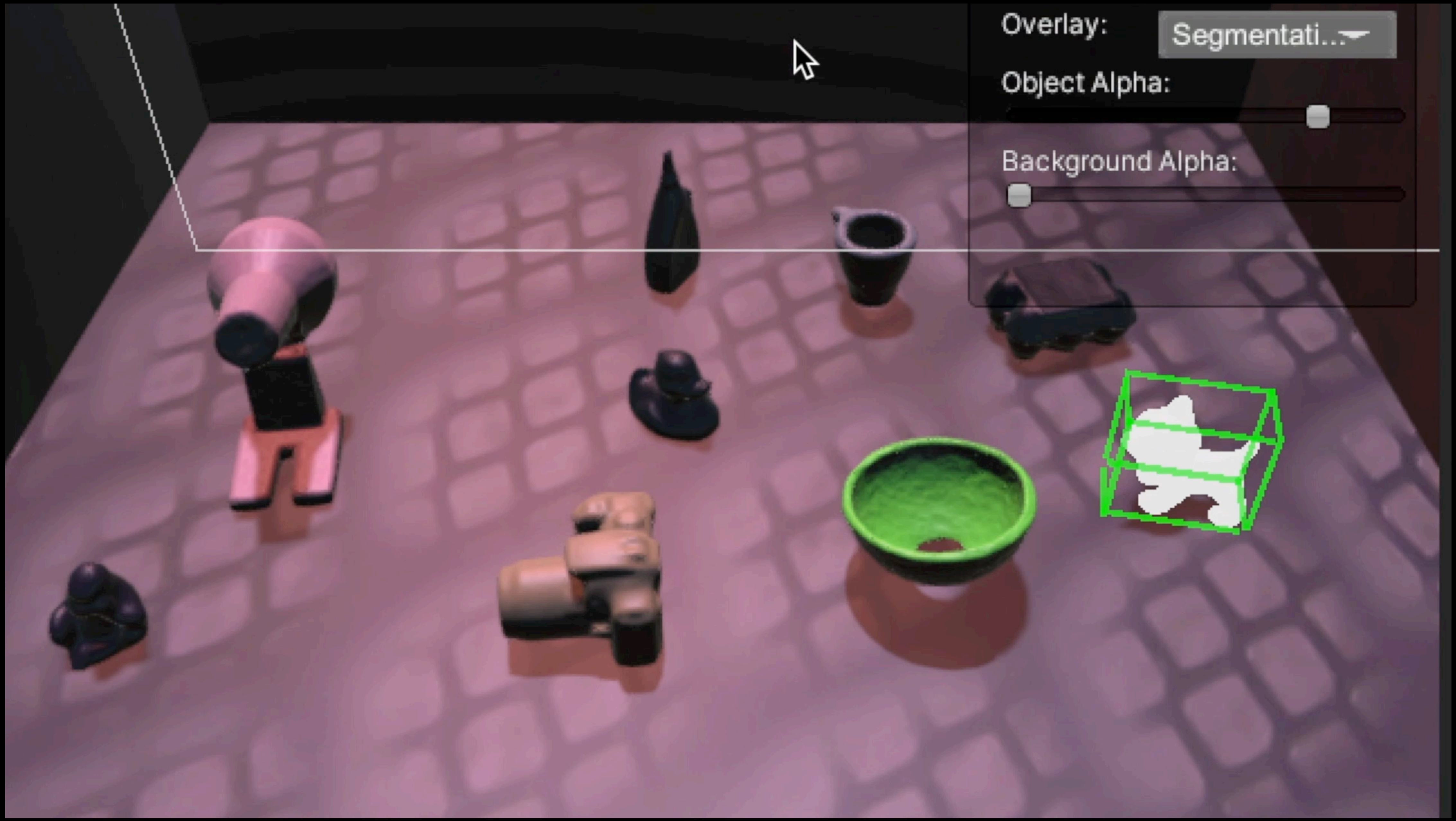
3D object pose estimation

- RGB-D image capture
- Labeling
  - 3D bounding box
  - Semantic Segmentation
- Camera Intrinsic and Extrinsic Parameters
- LineMod Assets
- Distractor Objects
- Randomizers
  - Camera
  - Object placement
  - Lighting
  - Background Textures



3D object pose estimation

- RGB-D image capture
- Labeling
    - 3D bounding box
    - Semantic Segmentation
- Camera Intrinsic and Extrinsic Parameters
- LineMod Assets
- Distractor Objects
- Randomizers
    - Camera
    - Object placement
    - Lighting
    - Background Textures



Conclusions

- Synthetic data can be the future of model training, but it is hard to make and use.
- PeopleSansPeople: a free to use synthetic data generator for human-centric computer vision research.
- 3D Object Pose Estimation environment available early next year.
- Synthetic data pre-training out performs real data pre-training.
- Can we learn optimal parameters of synthetic data generators?

Thank you

Unity®

UNITY.COM