covariant

HOW TO GUIDE

How to Run an AI Benchmark for Robotic Warehouse Picking

Contents

AI Benchmark Basics

AI Robotics has emerged as a powerful supply chain solution. Robots that can learn and adapt can unlock a previously impossible level of performance and reliability — for the first time enabling the automation of warehouse picking.

AI Benchmark Basics

The AI breakthrough that enables this capability represents a new challenge for supply chain leaders to evaluate.

You must understand if the AI software in question can learn tens of thousands of products and adapt to the dynamic nature of your operations – while continually achieving top performance objectives.

To learn more about the the need for performance based assessments, check out the Covariant Blog.

That means you need to evaluate the AI software itself – separate from the robotic hardware it powers. An objective performance based assessment is the only reliable way to assess the core capabilities of AI software.

This AI Benchmark Guide provides a framework to run a three day performance based assessment focused on evaluating the AI software behind these new robotic systems. The AI benchmark process is broken down into three parts:

1. Design
2. Execution
3. Analysis

1. Design

The AI benchmark starts with constructing a challenge that tests the most relevant aspects of AI software, while taking into consideration the unique context of your own operations.

There are two primary considerations that test the AI software: the objects to include and the scenes of how those objects are arranged.

The benchmark should be designed to assess AI performance for your most difficult situations. Think about your own operations as you read the following examples describing what objects and scenes are the most challenging for AI software.

Photo examples
Examples of object selection and scene design can be found in the Appendix starting on page 18.

Object Selection

The main goal of object selection is to pick objects representative of your product mix that will challenge the AI software.

Autonomy
Autonomy is defined by a robotic system achieving levels equivalent to manual performance, 99%+ success rates. Learn more on the Covariant Blog.

Key considerations

- Select at least five objects but usually no more than 25.
- To focus on the AI software capability, pick objects under 3 kilograms.

Photo examples
Object selection examples can be found in the Appendix starting on page 19.

Select a sample of "fast movers" – objects with high volume in your operations. But autonomy depends on the ability to handle your full product mix. Make sure you have a good representation.

All objects must be challenging for the AI software to pick. Characteristics like size, packaging type, color, and shape all influence how challenging an object is to pick. AI vision systems find patterns to help distinguish objects and determine how to pick. This becomes challenging when differences are difficult to distinguish or when it's unclear where to grasp the object, for example:

1. Optics: Reflective, transparent, or translucent materials
2. Texture: Objects with low contrast or repetitive patterns
3. Shape: Complex or non-uniform shapes
4. Flexible Packaging: Deformable packaging that lacks rigidity
5. Size: Small or thin

Examples of challenging objects

Additional photos
More object selection examples can be found in the Appendix starting on page 19.

Optics

Texture

Shape

Flexible Packaging

Size

Scene Design

The main goal of scene design is to test the AI software's ability to recognize your objects in a variety of unstructured scenes. It's important for the AI software to not just understand the object itself, but also learn to successfully pick an object as it dynamically appears in bins.

Design bin layout as close to your actual operation as possible. Think about how your decanting operators place objects into a bin and what those bins look like when they are both full and nearly empty, for example:

1. Chaotic Bin: Single SKU or multiple objects packed chaotically
2. Orderly Packed Bin: Packed orderly, but with no space in between objects
3. Compartmentalized: Objects placed in divided bins within the bin
4. Object Placement: Distributed in obscured positions within the bin when nearly empty – standing along the side, stacked on top of each other, or in the corner

Photo examples
Scene design examples can be found in the Appendix starting on page 24.

Examples of challenging scenes

Additional photos
More scene design examples can be found in the Appendix starting on page 24.

Chaotic Bin

Orderly Packed Bin

Compartmentalized Bin

Object Placement

Common Mistakes

Three mistakes to avoid with the design of an AI benchmark:

01 — Assessing hardware at the same time as AI software

AI software is a long-term, platform decision. Your choice of hardware - robotic arm, end effector, design of the station, etc. - will change across deployments and over time.

02 — Failing to test the long tail

In our experience it's typical to see fast movers among the test objects - they are the easiest to identify. Less common is to do an equally good assessment of the slow movers to identify representative objects. Testing the long tail of your product mix is just as important to achieving autonomy.

03 — Collecting insufficient data

Aim for enough data to make the results statistically significant. A good rule of thumb is 200 picks per object and at least 1,000 picks overall.

2. Execution

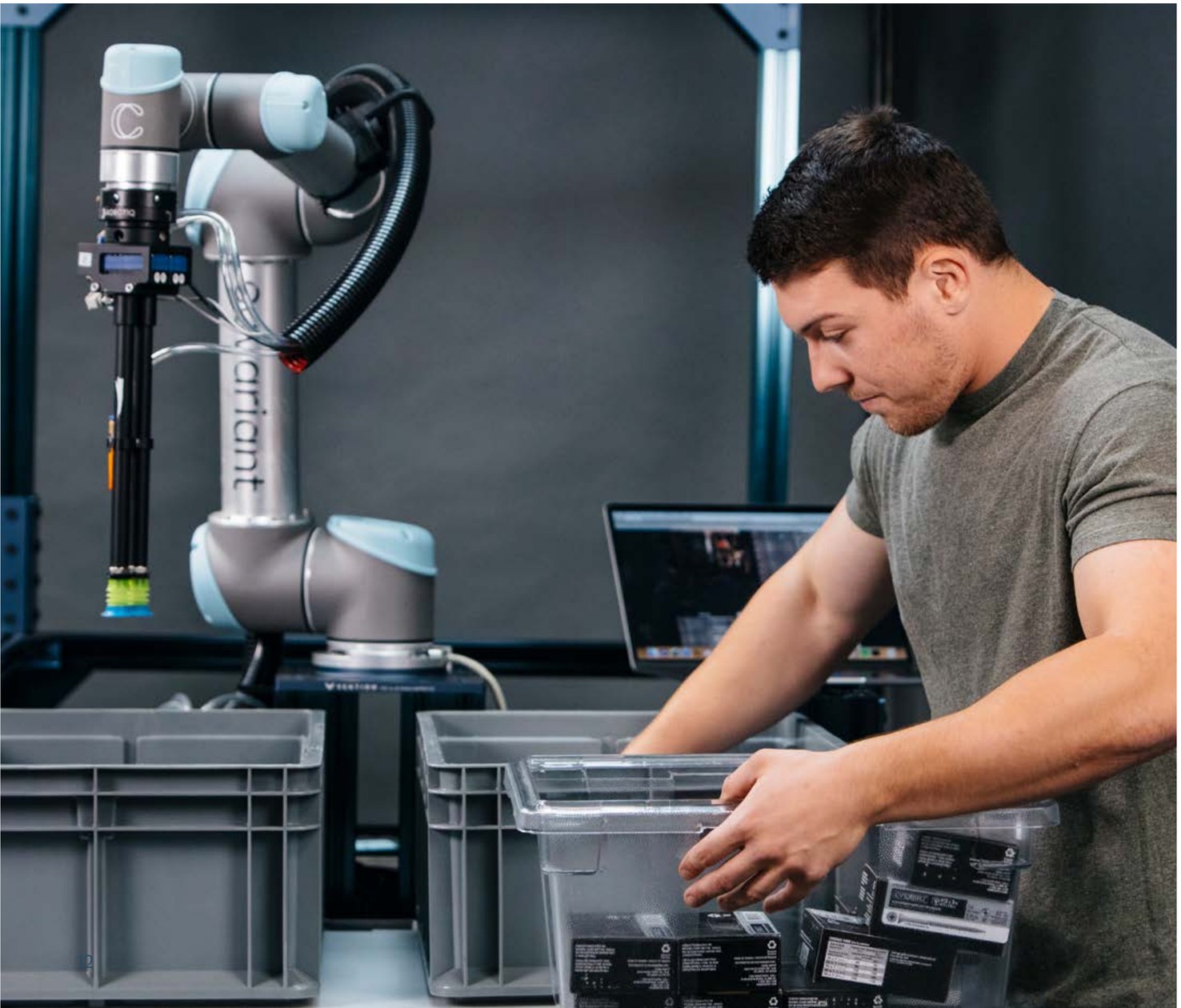
You've designed a great benchmark challenge intended to measure AI software's ability to autonomously perform in dynamic and challenging situations.

This next phase will guarantee that each of the selected AI software providers are setup to execute an objective AI test in the context of your own operations.

To learn more about these core AI capabilities, check out the Covariant Blog.

Properly outlining the conditions of the execution phase will help ensure that the assessment is focused on core AI capabilities – including out of the box performance, learning speed, and the ultimate learning potential of the system.

Execution



The first phase of the challenge starts by testing for **out of the box** native competency. To do so, instruct the AI software providers to open the box and run the defined scenes while taking an uncut video of the entire process. This should be the first time their systems are seeing your specific test objects.

The second phase explores the AI software's ability to learn. In this phase, AI software providers should **wait 72 hours** after the initial test.

– To determine the speed at which the system learns, instruct the providers to run the **exact same scenes** a second time.

– To analyze the learning potential and get a clear view of overall system performance, take at least a **30-minute uncut video** of the system continually picking a variety of your objects.

Setting expectations and providing clear directions are critical to executing a successful AI benchmark.

– In advance of the challenge, ensure all participants understand the goals (strategic considerations and quantifiable objectives) and the object parameters (but not the specific items that will be tested).

– When the time comes, provide the challenge objectives, a document with procedural instructions (along with pictures of intended scenes), and all test objects in a sealed box to the participants.

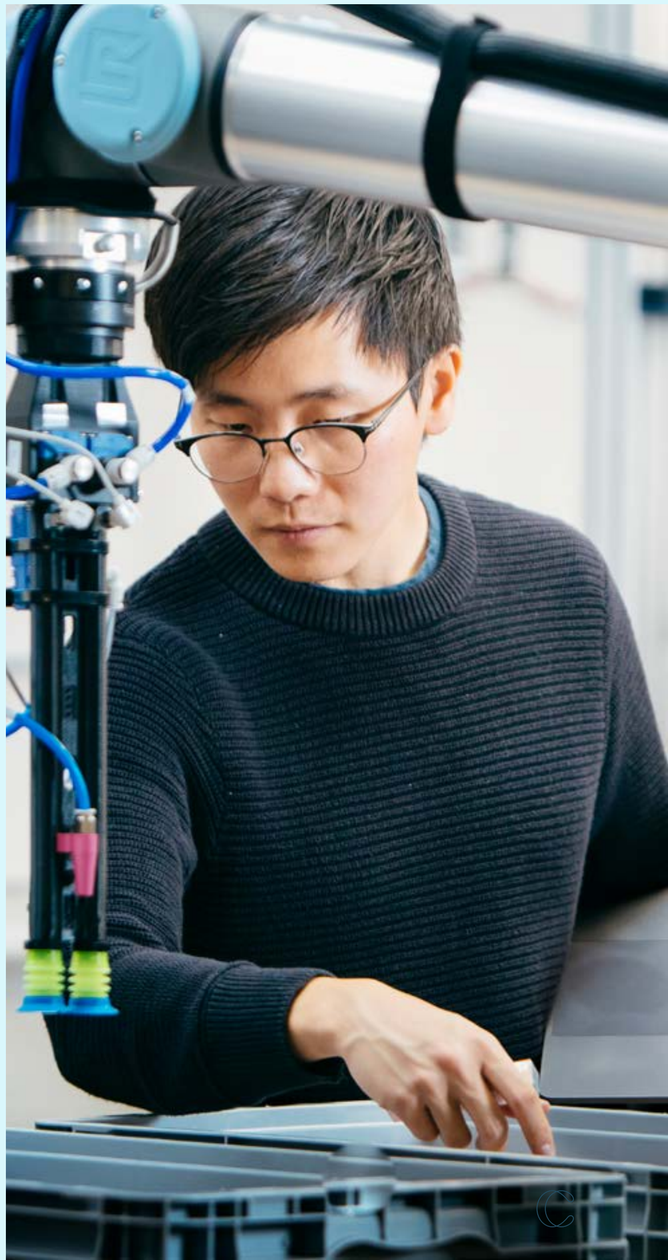
3. Analysis

How do you evaluate the results from each AI provider?

First of all, it's important for the AI software providers to submit both the video footage and their analysis of potential opportunities for improvement.

Similar to how we assess results in our day-to-day lives, how you get there is just as important as the outcome itself. The same is true of AI software. Therefore, run a quantitative assessment with a qualitative judgement of how the system performed.

Remember, your analysis should be focused on evaluating critical AI capabilities – including out of the box performance, learning speed, and the ultimate learning potential of the system.



Analysis

Quantitative Analysis

	Rationale	Calculation
Pick Success Rate (PSR)	In an AI benchmark, PSR abstracts away mechatronic and robotic considerations in the better known Picks Per Hour (PPH) (PPH is a metric associated with your overall systems integration performance. In this AI benchmark, focus on metrics directly related to the AI software like PSR)	# Successful picks / # Total picks
Intervention Rate (IR) Interventions include: - items dropped outside of bin - robot collision - resolve system deadlock (i.e. no grasp point, repeating failed grasps)	IR is a critical measurement of autonomy, the ability of a system to operate at or above manual process performance levels.	# Picks requiring operator interventions / # Total picks
Production Error Rate (PER) Production errors include: - place multiple picks to target bin - damage to items	Production errors typically have an operational cost orders of magnitude greater than the pick itself. PER captures that dimension.	# Production errors / # Total picks

Analysis

Qualitative Analysis

While quantitative metrics will provide a data-driven view for performance over a set of tests, ultimately you seek to understand performance at the scale of your total operations.

To understand the scalability of the results, we recommend a qualitative assessment of how the AI software performs.

There are two main concepts to consider: grasp quality, how effectively the system determines where to pick an item, and system quality, how effectively the system recovers from atypical situations.

Grasp Quality

Does the robot prefer grasps:

- on un-occluded objects
- at the object center
- on flat region within an object
- that utilize more suction cups

System Quality

Does the system have a strategy to:

- recover from failure autonomously
- break deadlock (i.e. don't repeat the same failed grasp, perturbation method, etc.)

As you watch the videos, imagine you were operating this robot remotely by looking through a state of the art Virtual Reality (VR) headset. Ask yourself, "Is the AI software making the same decisions I would?" Try to look for patterns across both system quality and grasp quality when answering this question.

Conclusion

AI software enables robots to learn and adapt to the world around them — delivering a previously impossible level of autonomy in dynamic environments. AI Robotics is redefining how we make, move, and store the products our communities and businesses rely on.

This is a breakthrough AI advancement that has also emerged as a transformative supply chain solution.

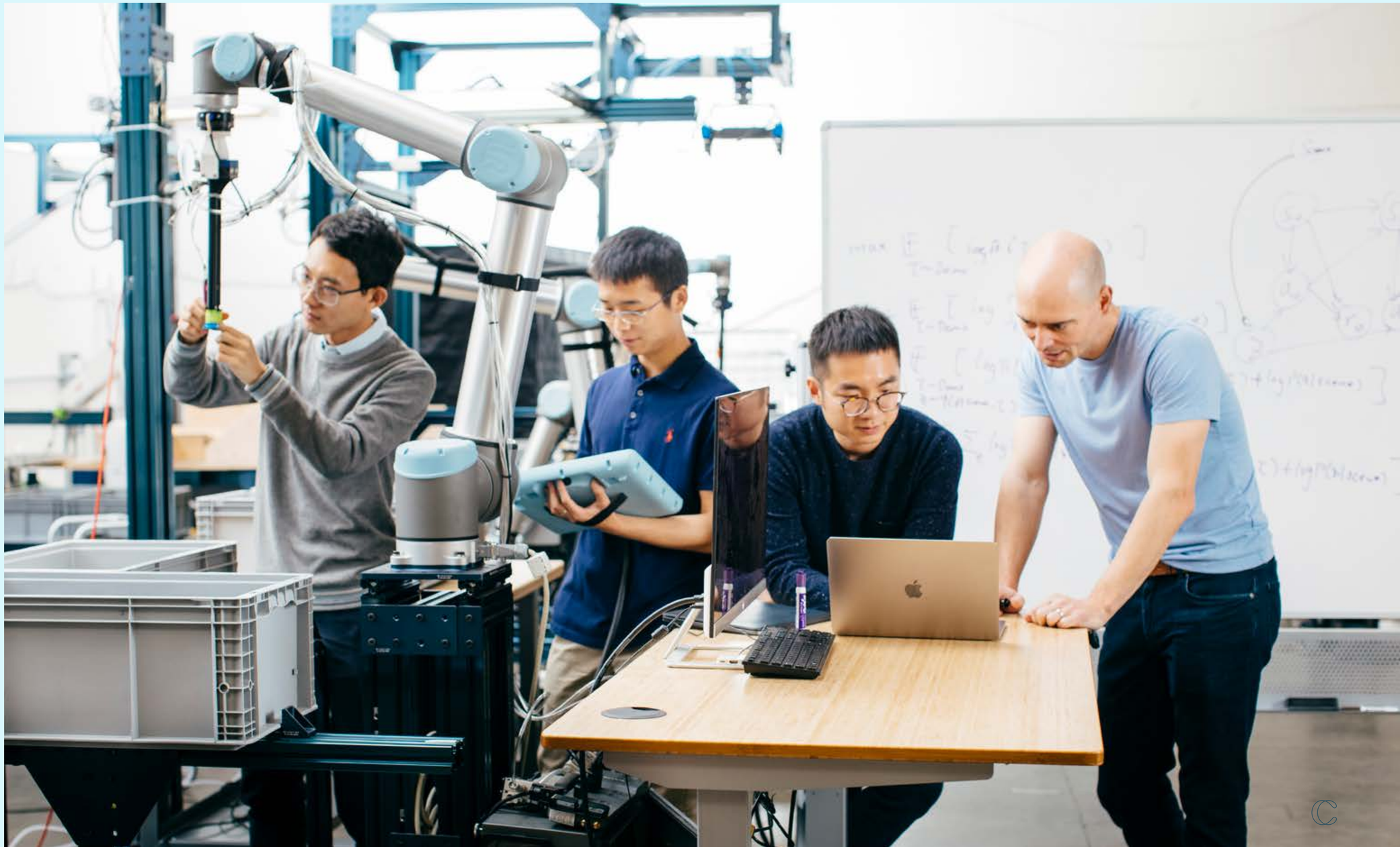
While the value is clear, evaluating AI solutions can be daunting. That's because there's no reliable way to judge a provider's AI capabilities without seeing their ability to handle the scale, complexity, and change of your operations.

A well executed AI Benchmark is an efficient and credible way to **test AI capabilities in the context of your own operations** — providing the confidence you need to invest at scale.

About Covariant

Covariant is building the Covariant Brain: universal AI that allows robots to see, reason and act on the world around them. Founded in 2017 by the world's top AI researchers and roboticists from UC Berkeley and OpenAI, Covariant is bringing the latest artificial intelligence research breakthroughs to the biggest industry opportunities. The company is headquartered in Berkeley, CA.

For more information, visit covariant.ai



Appendix

Examples of object selection and scene design

Object Selection

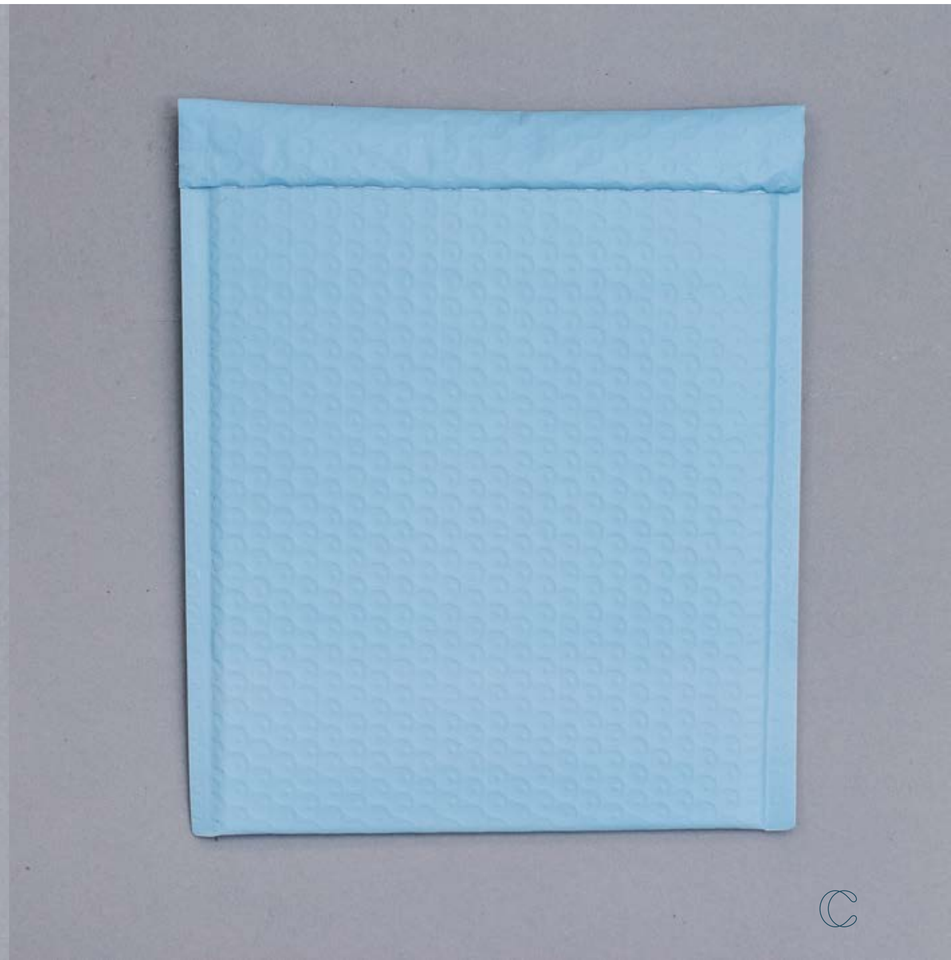
1. Optics

Reflective, transparent, or translucent materials

Object Selection

2. Texture

Objects with low contrast
or repetitive patterns



Object Selection

3. Shape

Complex and non-uniform shapes

Object Selection

4. Flexible Packaging

Deformable packaging that lacks rigidity

Object Selection

5. Size

Small or thin

Scene Design

1. Chaotic Bin

Single SKU or multiple objects packed chaotically

Scene Design

2. Orderly Packed Bin

Packed orderly, but with no space in between objects

Scene Design

3. Compartmentalized

Objects placed in divided bins within the bin

Scene Design

4. Object Placement

Distributed in obscured positions within the bin when nearly empty - standing along the side, stacked on top of each other, or in the corner