U.S. NAVAL RESEARCH LABORATORY

Developing "EMBODIED" autonomous robotic agents for the DoD with NVIDIA's Omniverse Platform

Naval AI Summit

May 15, 2024

US. Naval Research Laboratory

Joe Hays
Robotics and Machine Learning
Code 8234

Objectives

I hope that you leave this presentation with:

- A better understanding of what NVIDIA Omniverse is (and what it is not)
- How it can help you accelerate your developments (or not)
- An understanding of the GPU hardware requirements

U.S. NAVAL RESEARCH LABORATORY

So… what is NVIDIA Omniverse?

NVIDIA Omniverse™ is a platform of APIs, SDKs, and services that enable developers to easily integrate Universal Scene Description (OpenUSD) and RTX rendering technologies into existing software tools and simulation workflows for building AI systems.

1. **Asset Management**
2. **Bringing Assets to Omniverse**
3. **SDK for building custom capabilities**
4. **Visualization of assets**
5. **Adding physics**

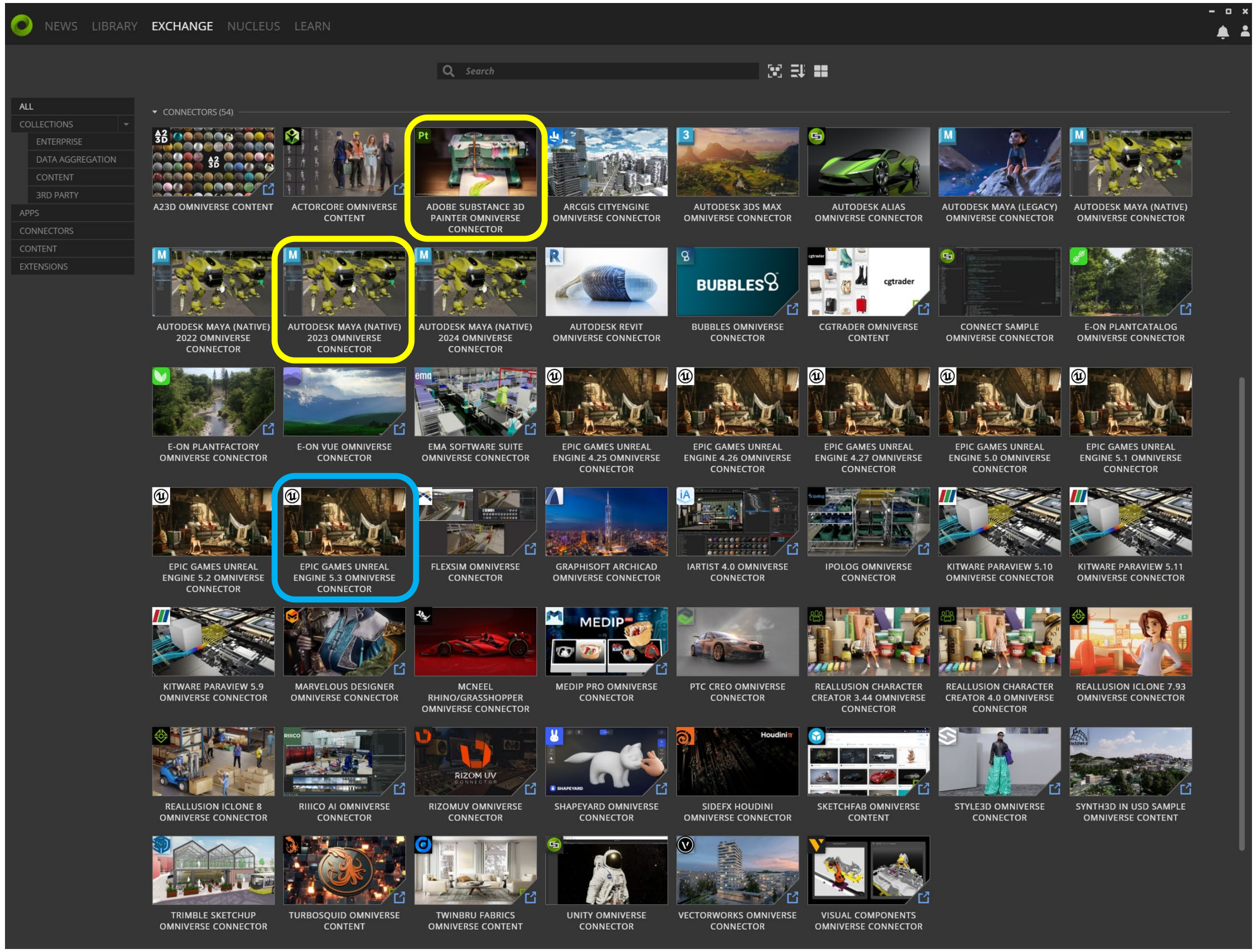
Omniverse Platform Components

Component	Name	Description
	Omniverse Nucleus	The central database and collaboration engine of Omniverse. With Nucleus, users share and modify representations of virtual worlds.
	Omniverse Connect	The libraries that connect popular content creation tools to Omniverse. With Connectors, users continue to work in their favorite software applications, such as SketchUp, Maya, and Unreal Engine, while benefitting from other Omniverse tooling.
	Omniverse Kit	The toolkit for building native Omniverse Apps, Extensions, and Microservices.
	Omniverse RTX Renderer	An advanced, multi-GPU renderer based on NVIDIA RTX™ that powers both real-time ray tracing and referenced path tracing.
	Omniverse Simulation	A powerful suite of tools and SDKs that simulate a physically accurate world.

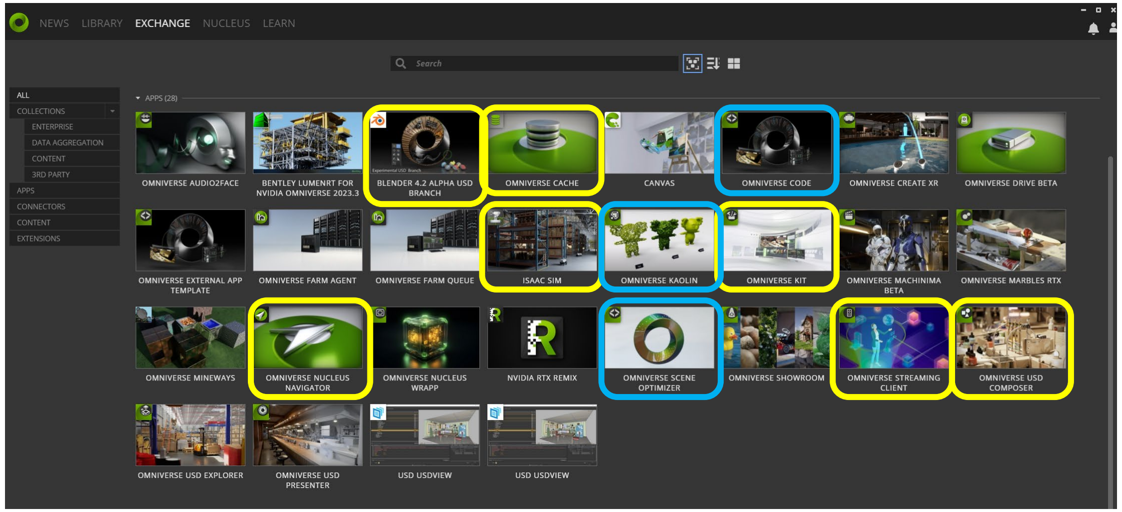
So... what is NVIDIA Omniverse? [Cont]

I like to call it an "umbrella" of many applications that address 3D content

Connectors →



Applications

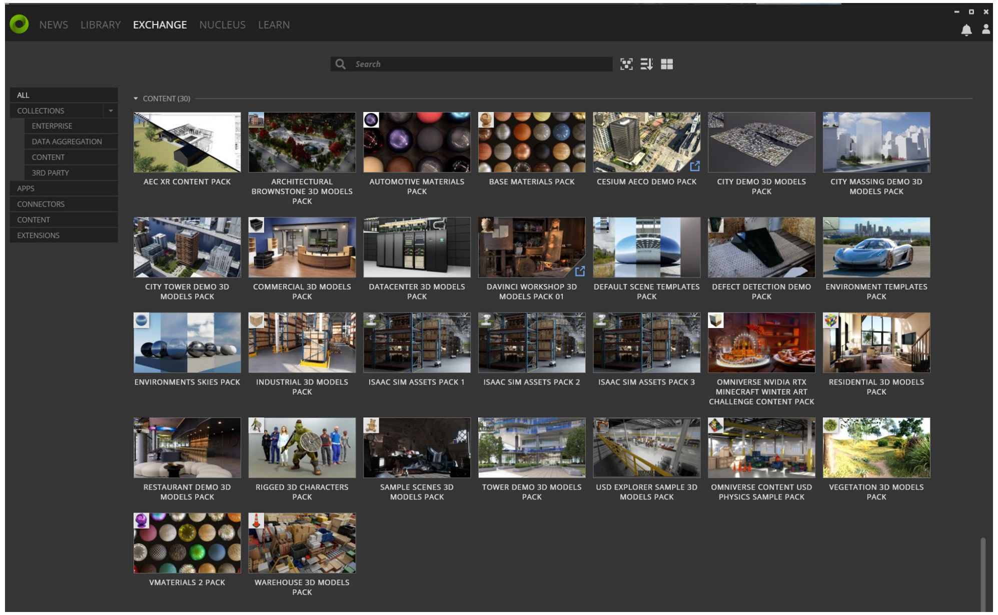


Actively Use

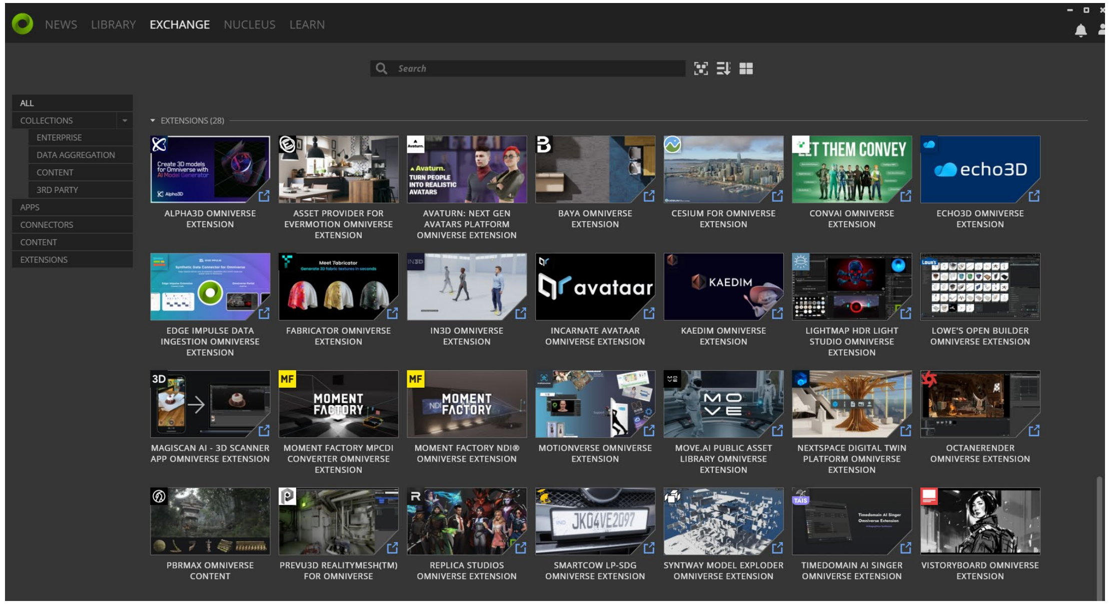
Anticipated Use

So… what is NVIDIA Omniverse? [Cont]

Content



Extensions

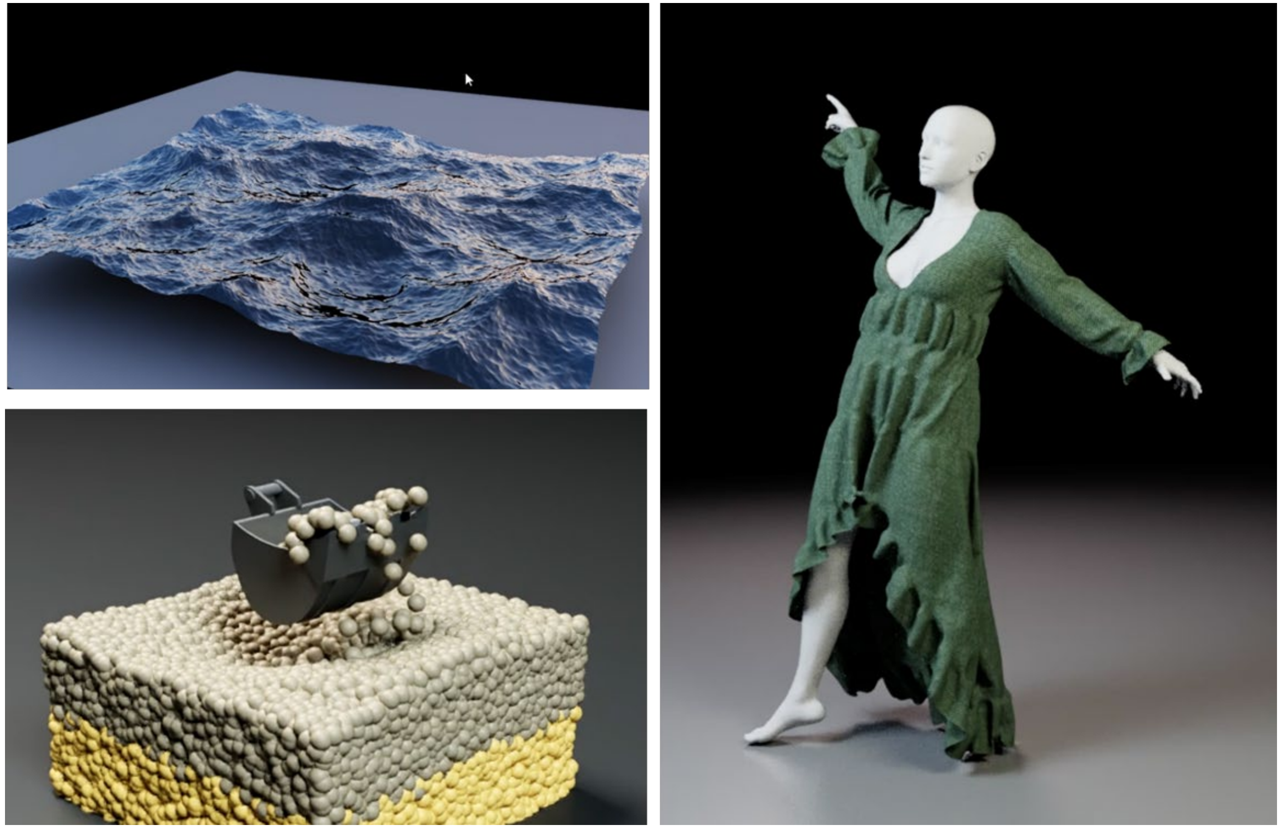


https://docs.omniverse.nvidia.com/extensions/latest/ext_simulation.html

- It's a massive eco-system
- It's honestly a bit overwhelming
- Join the "NRL_Omniverse" FlankSpeed Team
- Watch the trainings in the **FILES**
- Each Enterprise License is $5k! It's expensive, but extremely helpful

Omniverse WARP: programming extension

https://docs.omniverse.nvidia.com/extensions/latest/ext_warp.html

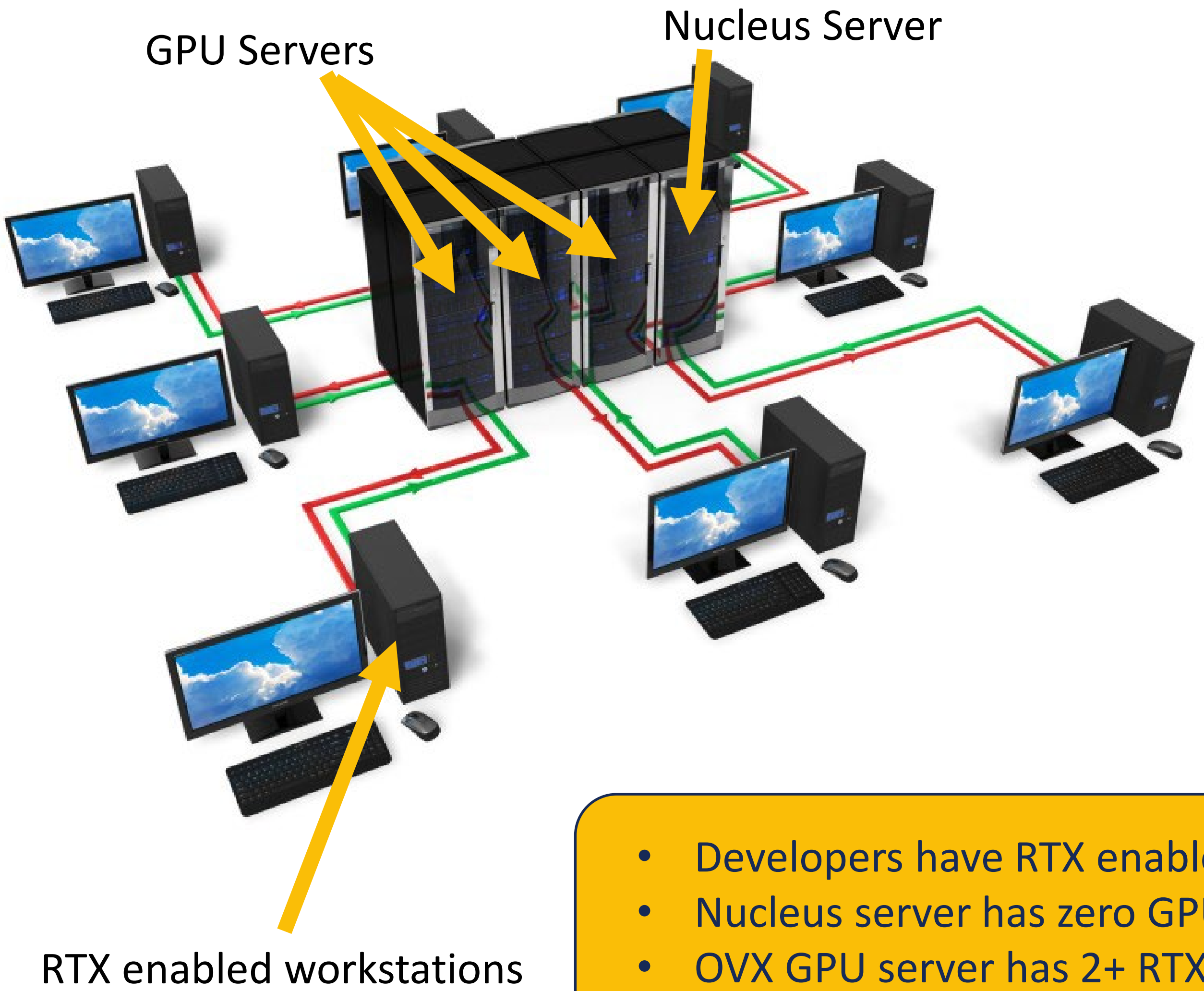


NVIDIA Warp is a Python framework for writing high-performance simulation and graphics code in Omniverse. Warp provides an easy way for Python coders to write kernel-based programs and benefit from GPU acceleration in Omniverse and OmniGraph.

KEY POINT:

The magic of Omniverse is **<u>ONLY</u>** unlocked if you absolutely **<u>minimize</u>** CPU-to-GPU communication!

Omniverse Nucleus: Asset Management



Omniverse Nucleus is the database and collaboration engine of Omniverse. With Nucleus, teams can have multiple users connected together live across several applications all at once. This fosters users to use the application(s) they are the most comfortable and quickest with and allows for rapid iteration.

Nucleus offers a set of fundamental services that allow a variety of client applications, renderers, and microservices all to share and modify representations of virtual worlds together.

- Developers have RTX enabled laptops/desktops
- Nucleus server has zero GPUs. It's sole purpose is to **host assets**
- OVX GPU server has 2+ RTX GPUs and handles the big jobs

U.S. NAVAL RESEARCH LABORATORY

Let’s Talk About GPU Hardware… (OVX vs DGX)

- GPUs are comprised of RAM and compute cores:
  - CUDA cores: general purpose scientific calcs
  - Tensor cores: optimized for tensor math
  - RT cores: optimized for ray-tracing math
- NVIDIA has two main families of GPUs for Machine Learning
  - DGX (e.g. A100, H100, H200, Blackwell, etc…)
    - H100: 80GB RAM, up-to 350W
      - Tons of Tensor cores
      - No RT cores
  - OVX (e.g. A40, A6000, RTX 40x0, RTX 30x0)
    - A40: 48GB RAM, up-to 300W
      - Mixture of Tensor, CUDA & RT cores

DGX/OVX Nodes **MUST** have same GPUs (no hybrid mix/match support)
Pods **MUST** matching nodes
DGX SuperPOD™ **MUST** have matching Pods

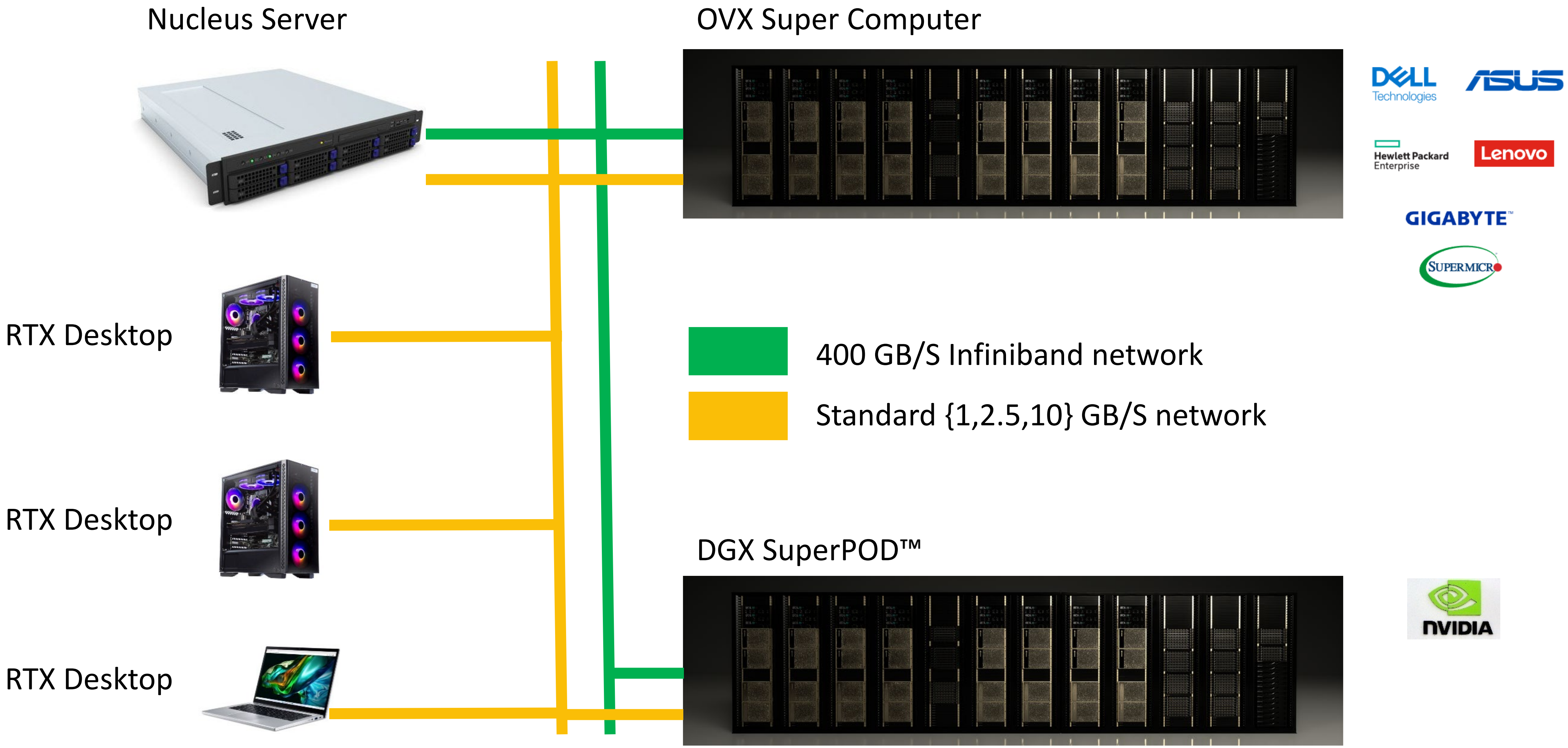
OVX/DGX Node
(e.g. 1+ GPUs)

Pod
(e.g. 2+ nodes)

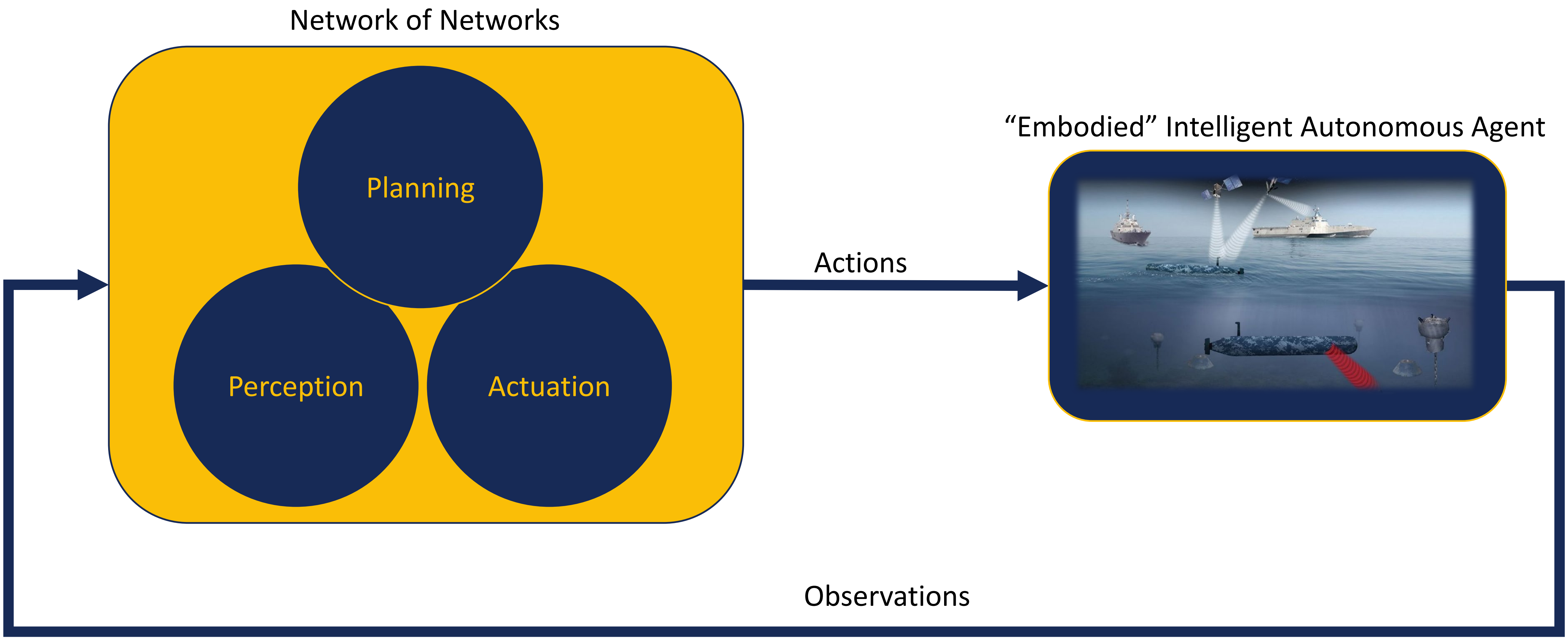
“SuperPOD”***
(e.g. 2+ Pods)

U.S. NAVAL RESEARCH LABORATORY

Need both OVX & DGX hardware to address all needs…



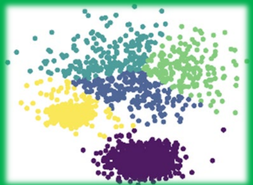
“Embodied” Intelligent Autonomous Systems: A Network of Networks



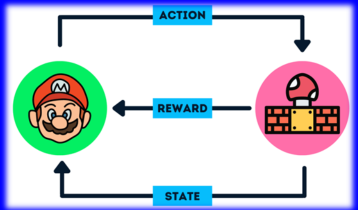
Machine Learning Families

Supervised Learning
e.g. CNN, Regression, etc…

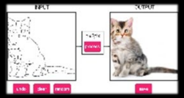

UnSupervised Learning
e.g. K-means, PCA, self-supervised learning etc…

Reinforcement Learning
e.g. PPO, SAC, etc…



Generative
e.g. cGANs, cVAEs, etc…



e.g. GANs, VAEs, etc…

DreamStudio

e.g. RSSM, GAIL, etc…

Fixed-Dataset

Exploration / Simulation

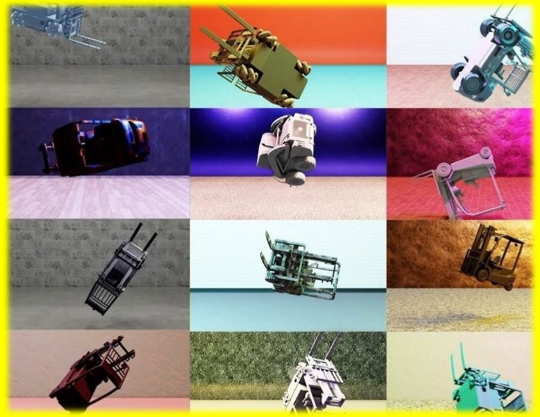
These two distinctions have significant GPU Training h/w implications

Relationships Between Learning Types In Context of GPU Acceleration

Studies:

1. Synthetic Dataset Generation

Needs RTX enabled GPUs (OVX: A40, L40)



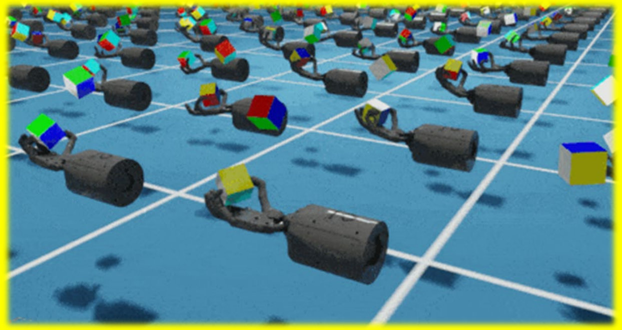
2. Fixed Dataset based learning

DGX: H100/H200/Blackwell GPUs highly preferred ***

3. Exploration/Simulation based Learning

Needs RTX enabled GPUs (OVX: A40, L40)

“On-policy” simulations → Data Sets don’t work for this!



OVX Hardware

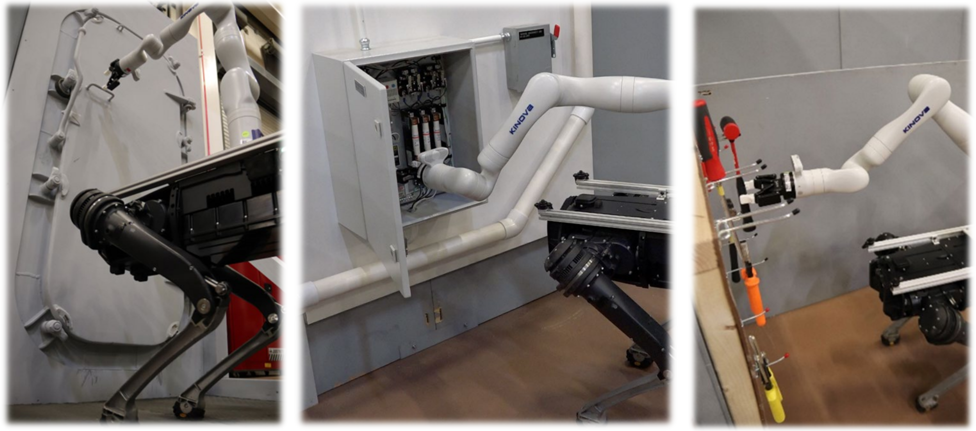
DGX Hardware

*** Can be done with OVX but not as capable

OVX Hardware

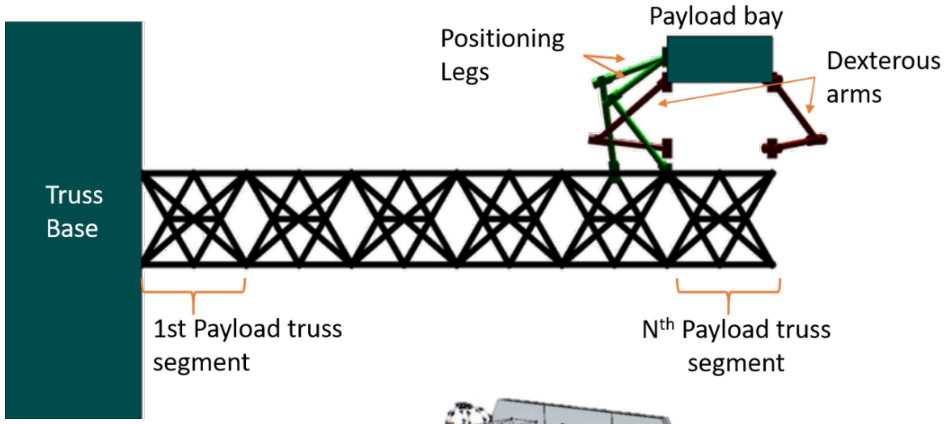
Intelligence "Embodied" Autonomous Agents [NRL Code 8234]

Naval Ship Maintenance Tasks



Door opening and panel maintenance using Kinova Gen3 Arm mounted on Ghost Robotics Vision 60 dog

Robotic In-Space Assembly



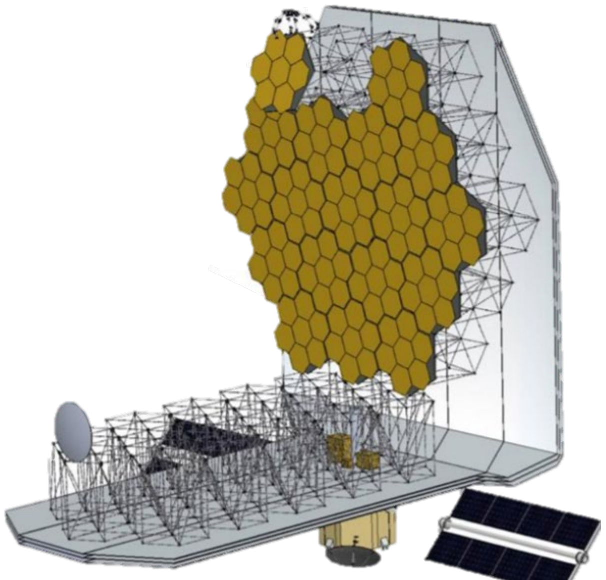
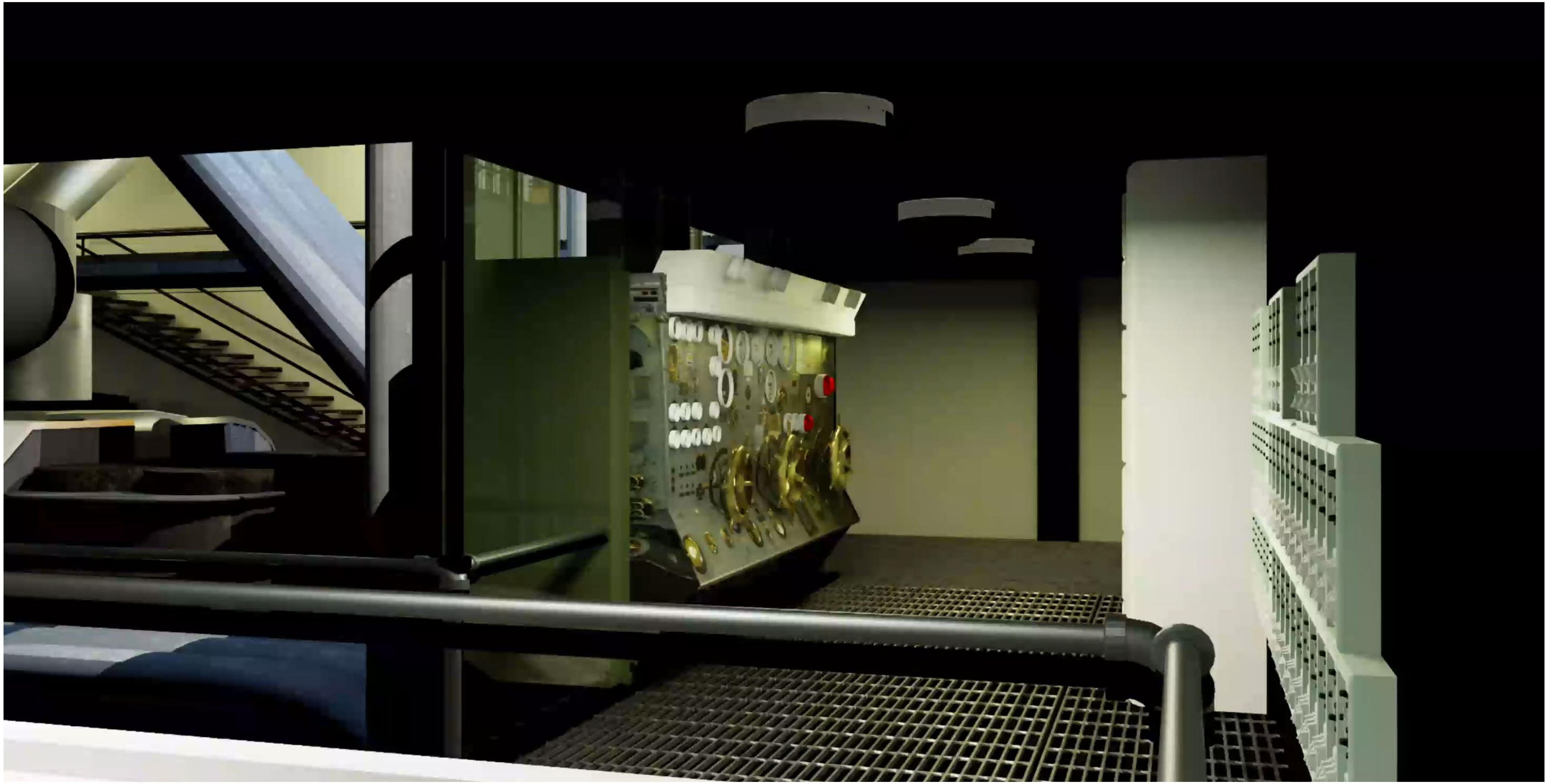


Image: **NASA in-Space Assembled Telescope (iSAT) Study**
https://exoplanets.nasa.gov/exep/technology/in-space-assembly/iSAT_study/

Ship interior hallway

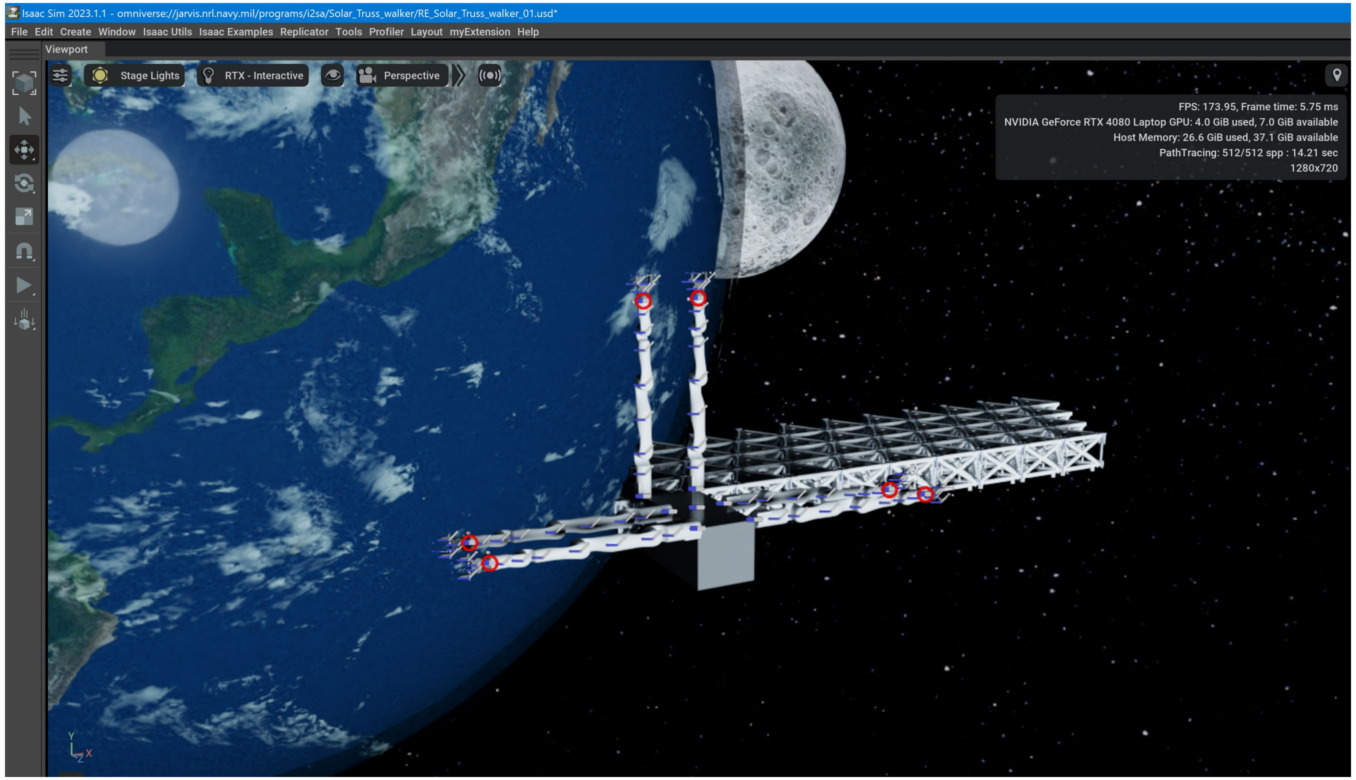
Ship mechanical room with fire



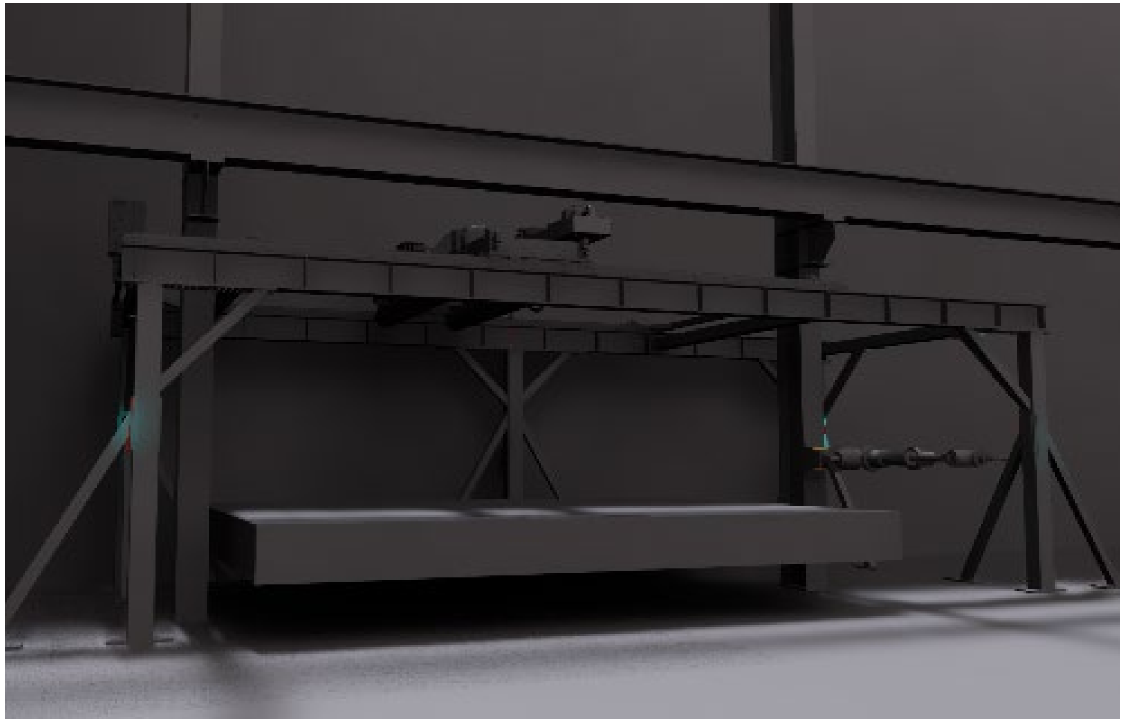
Digital Twin of Test Setup at NRL's LASR lab

Orbital Assembly Worksite & Lab Digial Twin Test Environment

6 limbed robot + tri-truss structure assembly in orbit



Gravity-offset-table (GOT) test facility



*** NOTE: we just converted CAD on this one… we have not yet gotten to texturing and such

Some Lessons Learned…

NRL Code 8234 has been using Omniverse for about 18 months

- Small team
- Still very much a work in progress
- We primarily based our studies on Isaac Sim based simulation

Lots of Navy & Army teams have reached out to us to learn about Omniverse. Some lessons learned include:

- 3D asset development is a **<u>significant</u>** labor cost
  - We hired 2 recent grad 3D digital artists (trained to make PIXAR movies)
  - Training them to understand Machine Learning needs in 3D asset development → new requirements/needs Universities did not train them in
    - We're developing "best practices" guidelines for the 3D artists driven from the Machine Learning/Python Developer's perspectives:
      - File organization: linking objects versus embedding
      - Hooks for automated parameter randomization ***
  - Just because it has the right geometry and visual textures DOES NOT mean that it's ready to be used for machine learning trainings! Models must be updated for
    - Class labeling
    - Automated parameter variation
    - Maximizing object reuse

- ****** SHARED NUCLEUS SERVERS BETWEEN NAVY ORGANIZATIONS WILL HELP CONSIDERABLY HERE! ******
  - Currently lots of red tape to get this setup
- ****** SHARED EXTENSION/TOOLS DEVELOPED BY DOD USERS WILL HELP CONSIDERABLY! ******
  - Need a shared Git repo (available to all DoD) with utilities/tools valuable to the DoD that NVIDIA does not (will not) develop
    - Currently lots of red tape to get this setup
  - Lots of missing needed utilities to make developers more productive (e.g. move linked Nucleus asset and update all references through all Nucleus)

Getting Started…

- Don't run off and buy a big GPU server
- Start by getting each developer a reasonable RTX enabled desktop/laptop (e.g. desktop + RTX 4090 or laptop + RTX 4080)
- Stand up a Nucleus server for your hosting your assets. Just need a decent file server. No GPUs.
- Once you get some experience, then look at a getting a supped up OVX node with multiple A40s. IF you have budget, then you can move to a POD or OVX version of a "SuperPOD"
  - You will have many months of work just getting 3D digital assets made, machine learning pipelines refined using the expansive Omniverse SDK… you won't be ready for massive training sessions at the beginning.

U.S. NAVAL RESEARCH LABORATORY

Steps toward domain adaption: augmenting NVIDIA Replicator / Orbit Cloner

- Simulation based training has the Achilles' heel of needing to bridge the sim-2-real gap.
- Photorealistic physical simulations (like that possible with Isaac Sim) are a step toward helping bridge this gap (for some applications!)
- NRL Code 8234 is developing synthetic data augmentation wrappers around the existing NVIDIA Omniverse Replicator and open source Orbit extension's Cloner:
  - Enables parameter variations ranging from: visual texturing, lighting, objection placement, object replacement, and physical properties such as mass/inertia/friction, etc…
  - Automagically randomized curriculum condition on loss performance during training:
  - Leverage extensibility of USD open-source standard
    - Parameter variation types, ranges, and schedule persisted within linked USD object files.
    - Training codes parse linked USD files and build a list of environment/model parameter variations
    - Allows automated scheduling of variations to maximize stability while learning a generalized policy capable of bridging the sim-2-real gap…
  - Can be used for both fixed synthetic data-sets generation and exploration/simulation based trainings (i.e. RL)
- Making this extension available in a shared Git repo for all DoD would help other labs successfully bridge the sim-2-real gap in their respective developments.

Key Take-Aways

- NVIDIA Omniverse is capable of significantly accelerate machine learning developments that involve:
  - synthetic dataset
  - simulation / exploratory
- Omniverse **requires** RTX enabled GPUs.
  - For large servers, this means you MUST have an OVX-based server (possibly in addition to a DGX-based server for fixed data-set based trainings)
- Start simple, get developers their own dedicated RTX enabled desktop/laptop and a shared Nucleus server to start.
  - Add more powerful OVX nodes, Pods, “OVX-like” SuperPODs when the team’s capabilities mature
- Join the “NRL_Omniverse” FlankSpeed Team
- Be ready to invest in 3D asset development… it takes time
- As more DoD groups begin to embrace Omniverse shared Nucleus servers for 3D asset management and shared Git repos for utilities across DoD installations will drastically accelerate productivity

Thank you. Questions?