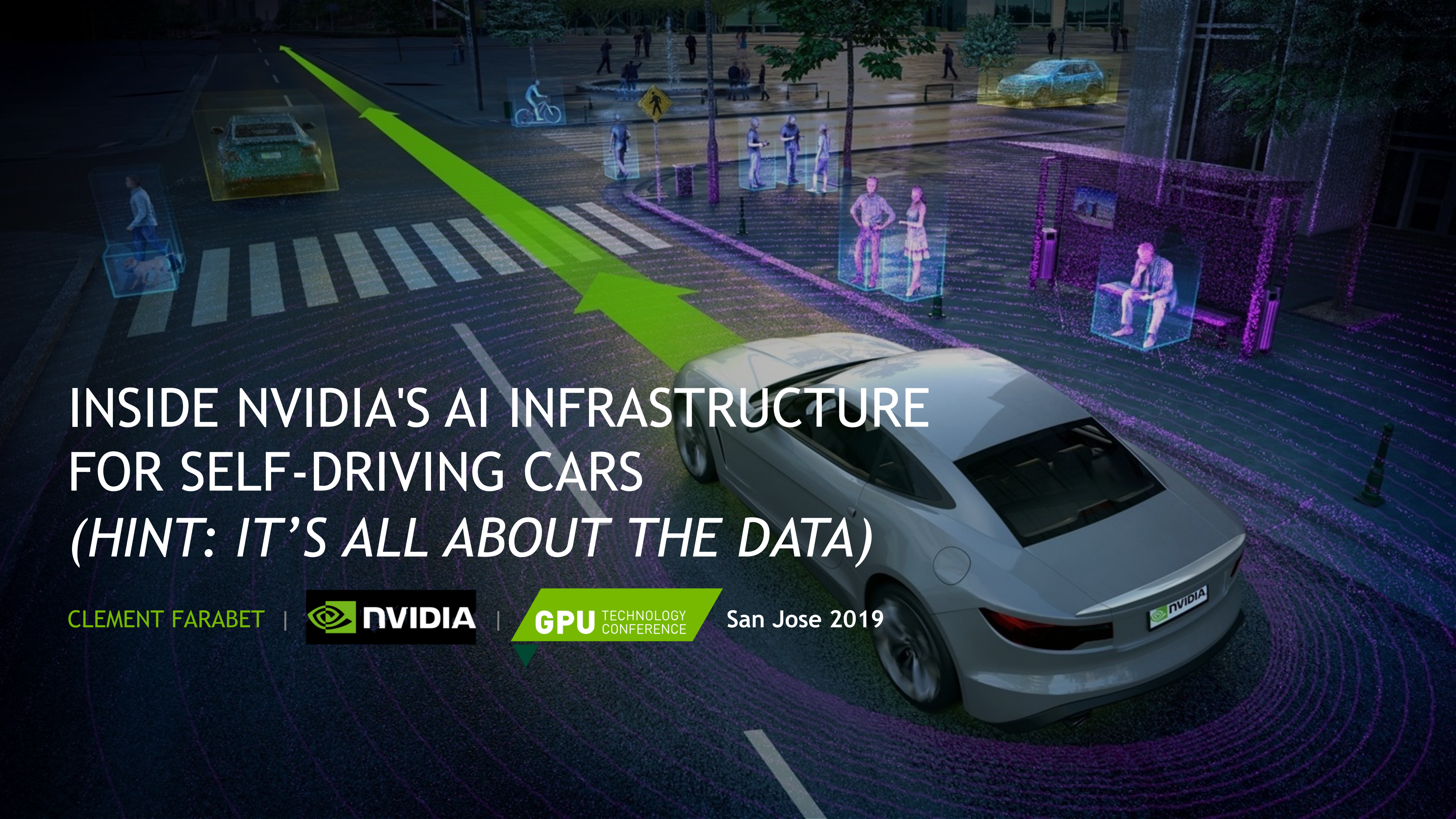


INSIDE NVIDIA'S AI INFRASTRUCTURE FOR SELF-DRIVING CARS

(HINT: IT'S ALL ABOUT THE DATA)

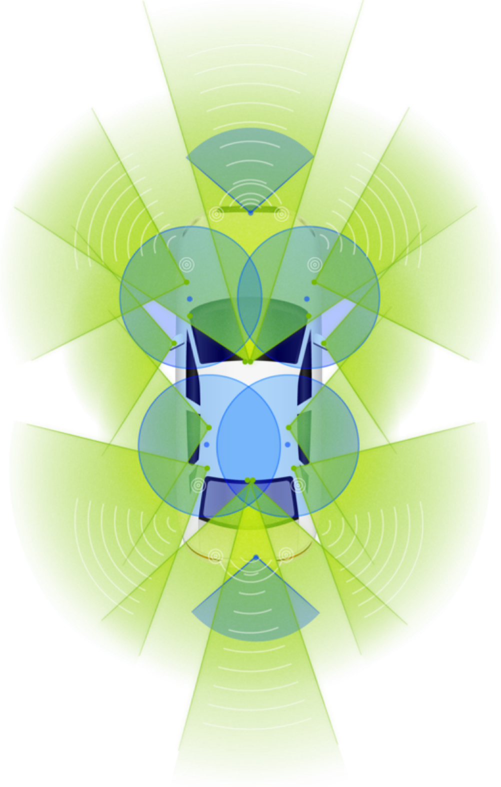
CLEMENT FARABET | NVIDIA | GPU TECHNOLOGY CONFERENCE San Jose 2019

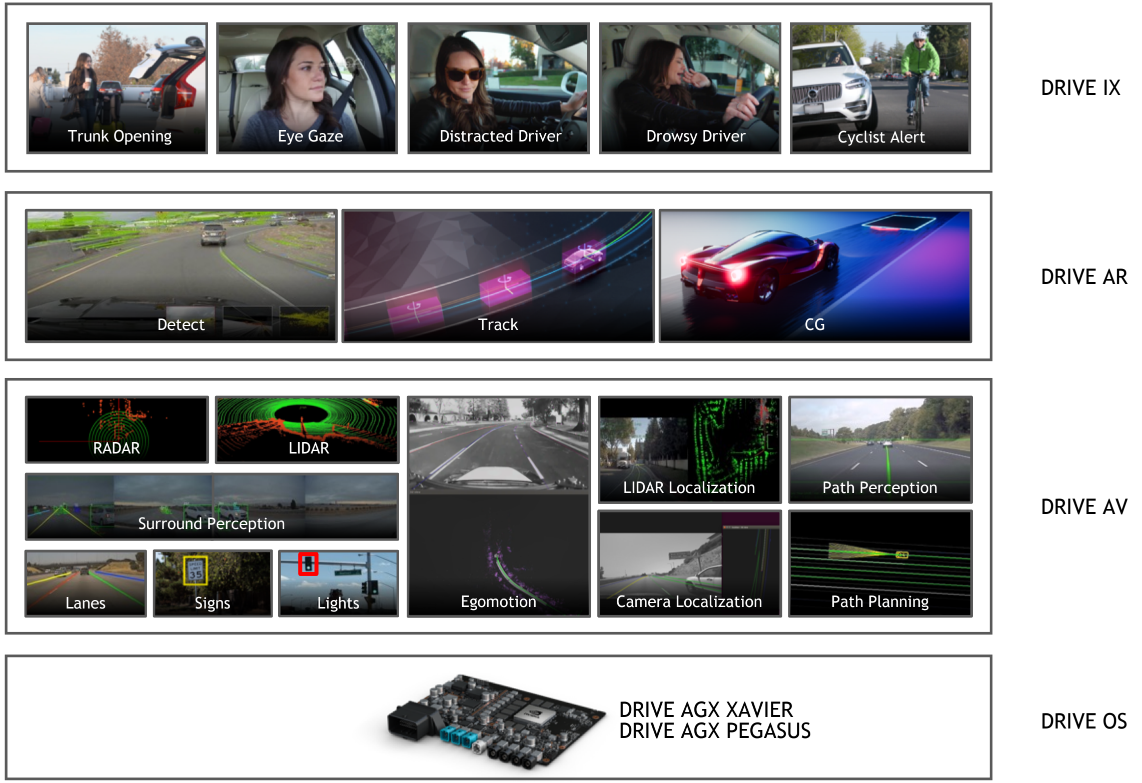
<u>Self-driving cars</u>

requires tremendously large datasets for training and testing

NVIDIA DRIVE: SOFTWARE-DEFINED CAR

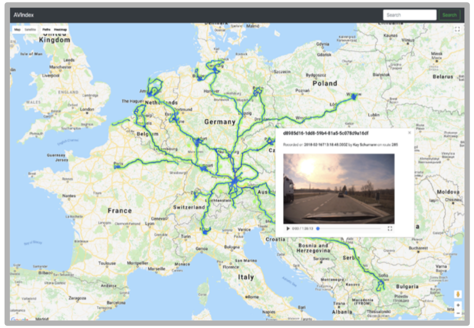
Powerful and Efficient AI, CV, AR, HPC | Rich Software Development Platform
Functional Safety | Open Platform | 370+ partners developing on DRIVE





BUILDING AI FOR SDC IS HARD

Every neural net in our DRIVE Software stack needs to handle 1000s of conditions and geolocations



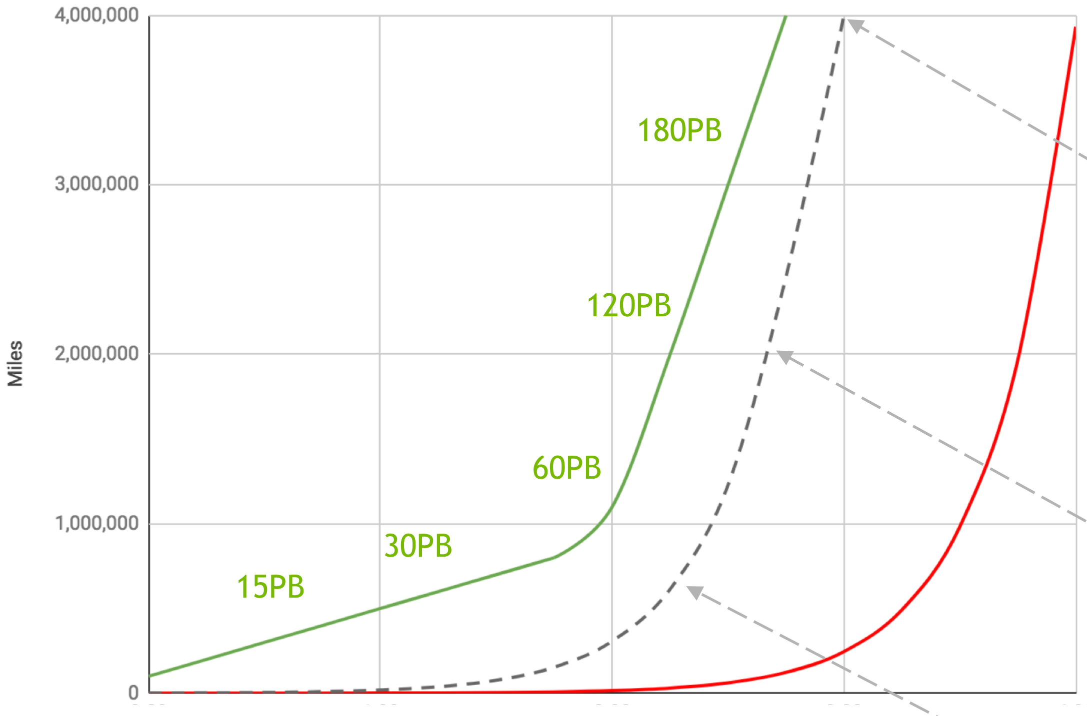
WHAT TESTING SCALE ARE WE TALKING ABOUT?

We’re on our way to 100s PB of real test data = millions of real miles
+ 1,000s DRIVE Constellation nodes for offline testing alone
& billions of simulated miles

— Target robustness per model (miles)

-- Test dataset size required (miles)

— NVIDIA’s ongoing data collection (miles)



24h test
on 3,200 Nodes*

24h test
on 1,600 Nodes*

Real-time test runs in 24h
on 400 Nodes*

* DRIVE PEGASUS Nodes

NVIDIA.

SDC SCALE TODAY AT NVIDIA

12-camera+Radar+Lidar
RIG mounted on 30 cars

1,500 labelers

4,000 GPUs in cluster
= 500 PFLOPs

1PB+ raw data
collected/month

20M objects labeled/mo

100 DRIVE
Pegasus in cluster
(Constellations)

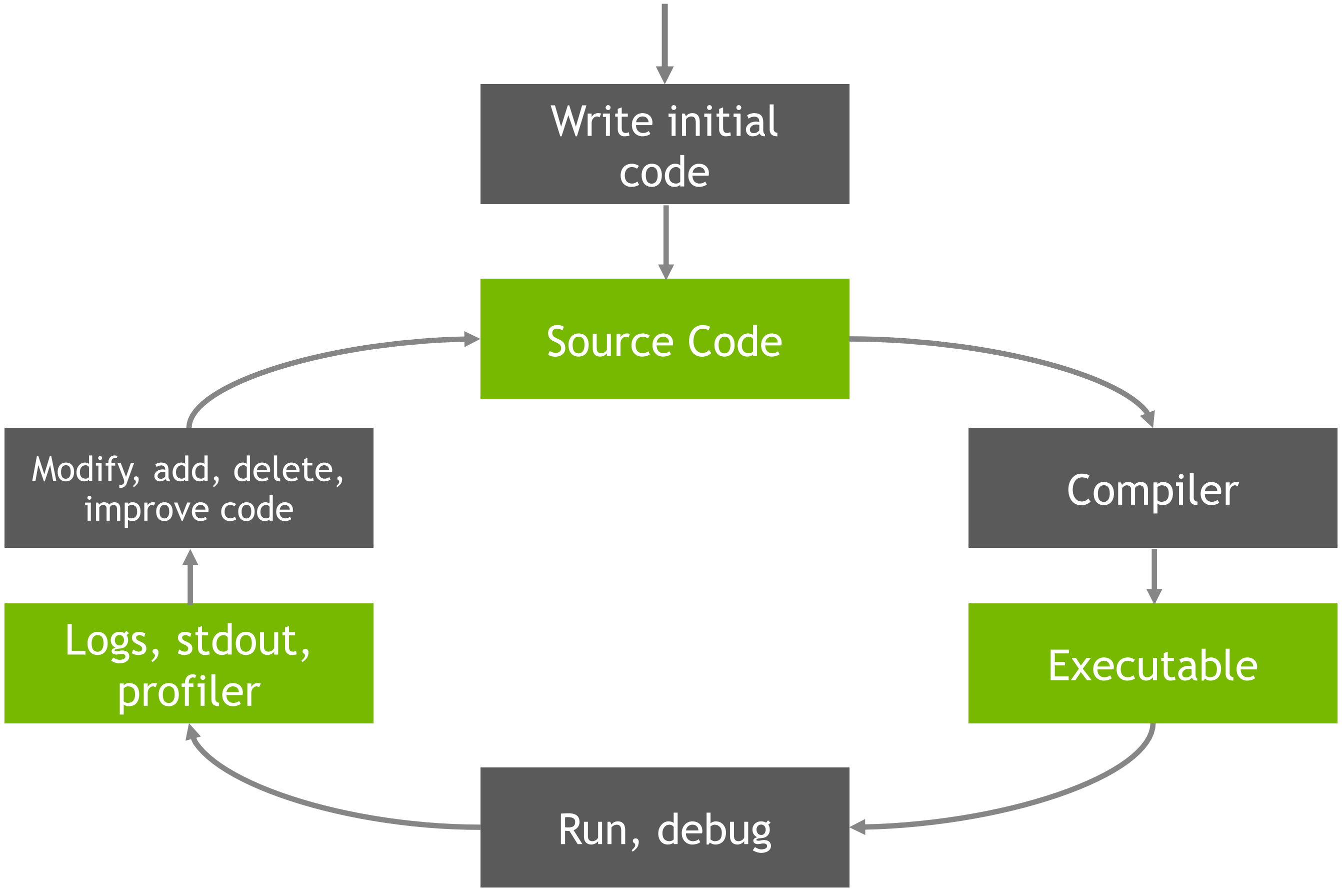
15PB raw active
training+test dataset

20 unique models
50 labeling tasks

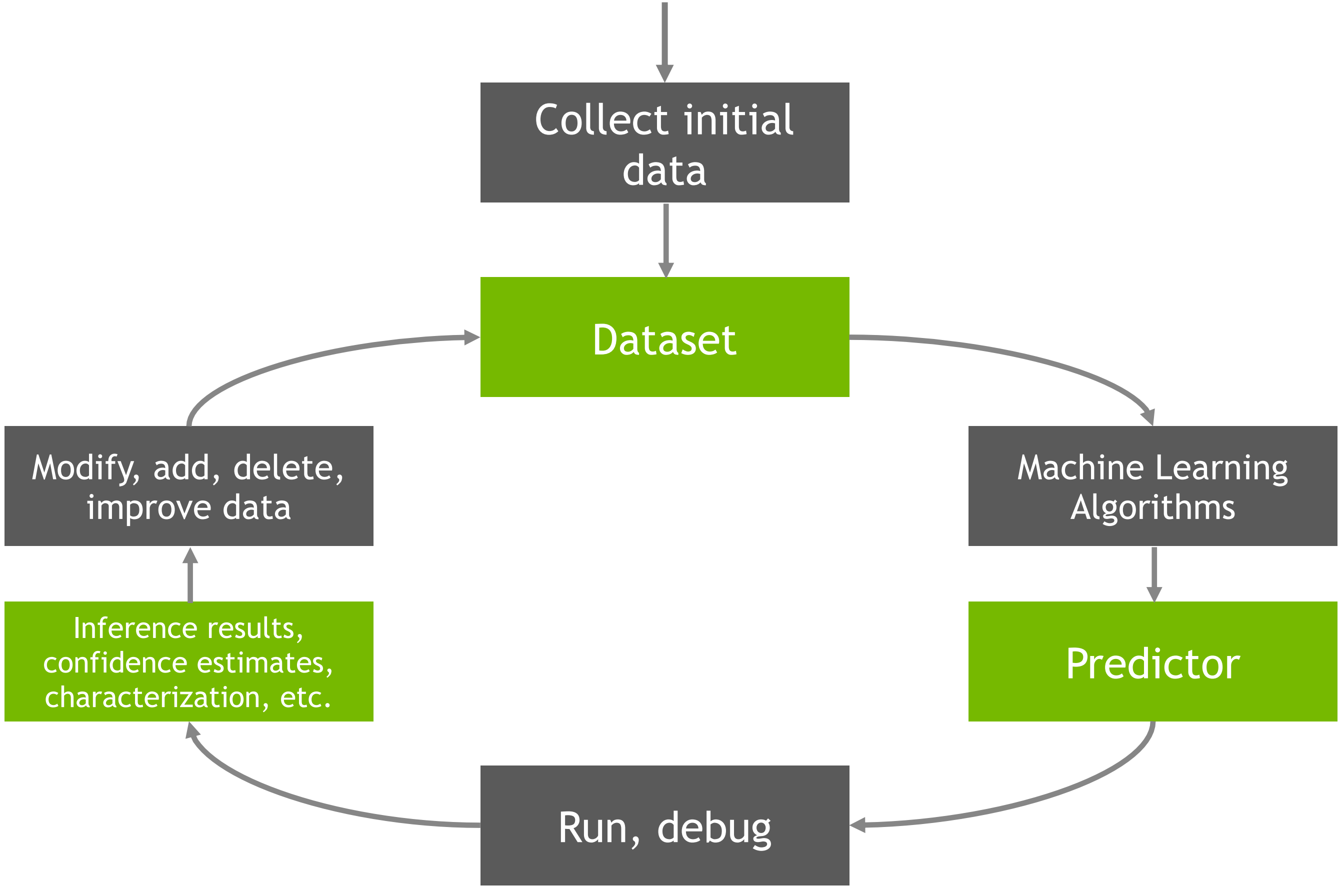
1PB of in-rack object
cache per 72 GPUs,
30PB provisioned

Creating the right datasets
is the cornerstone of
machine learning.

TRADITIONAL SW DEVELOPMENT

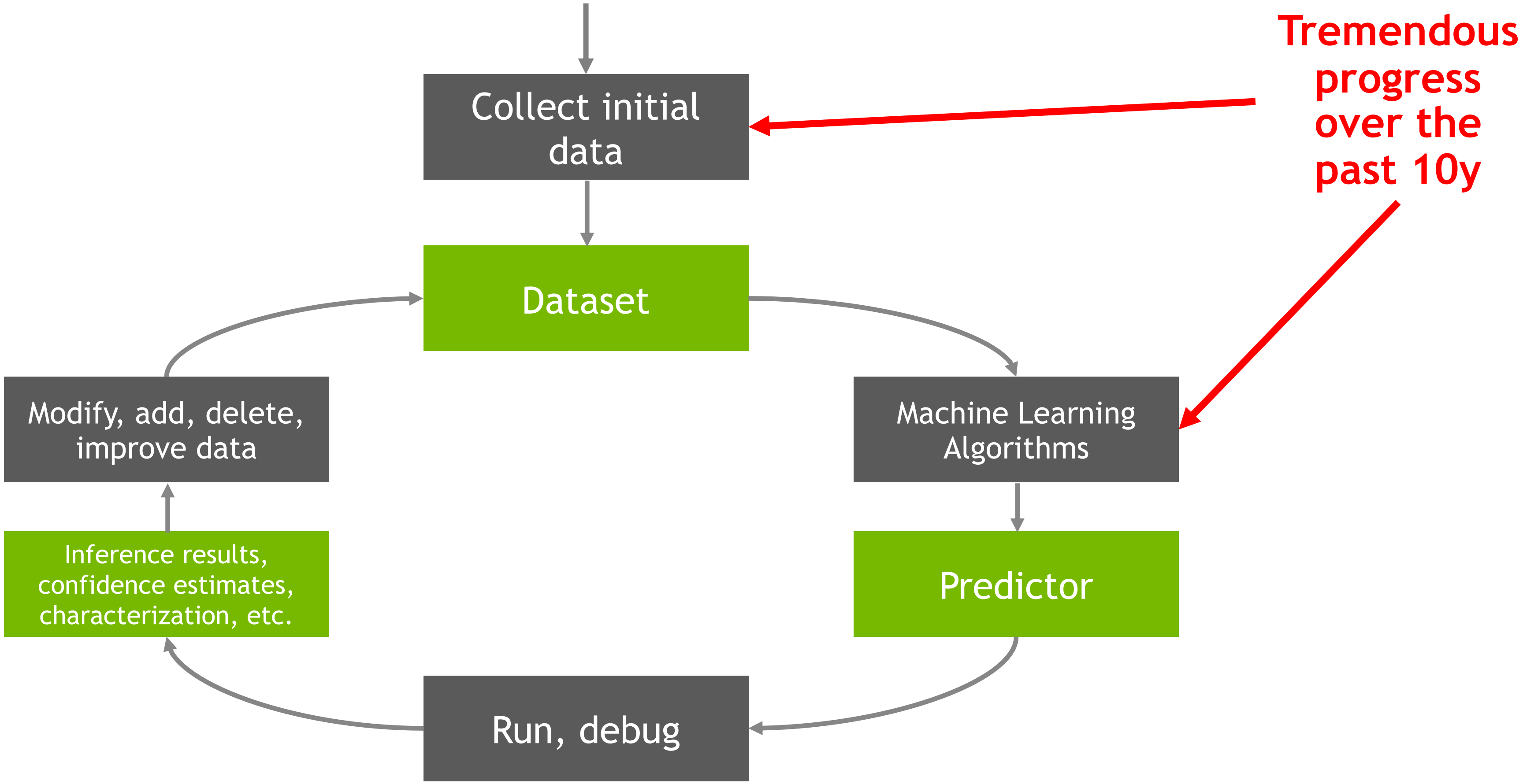


ML-BASED SW DEVELOPMENT

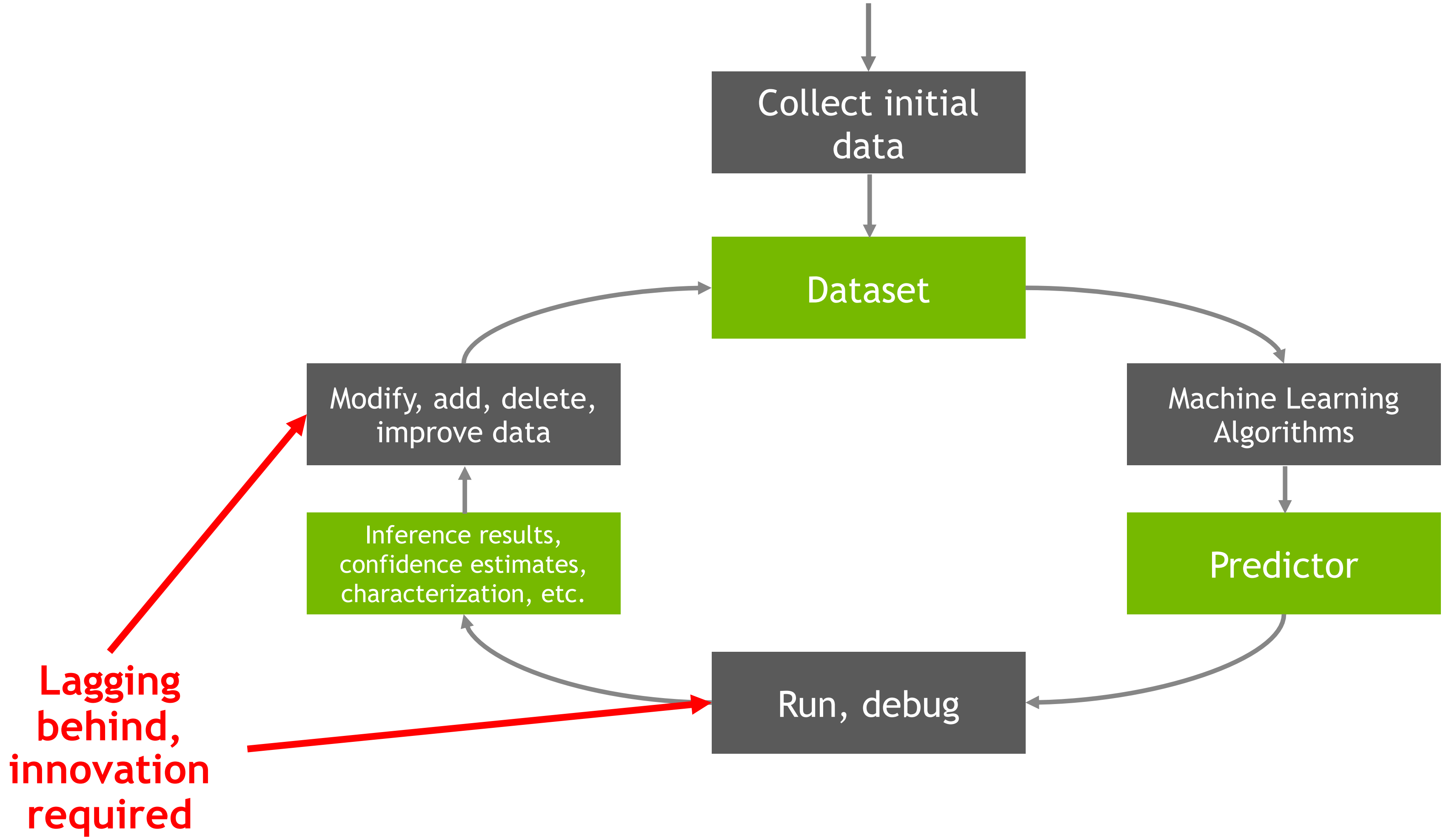


TRADITIONAL SOFTWARE		ML-BASED SOFTWARE
Source code	→	Data
Compiler	→	DL/ML Algos
Executable	→	Predictor

ML-BASED SW DEVELOPMENT



ML-BASED SW DEVELOPMENT

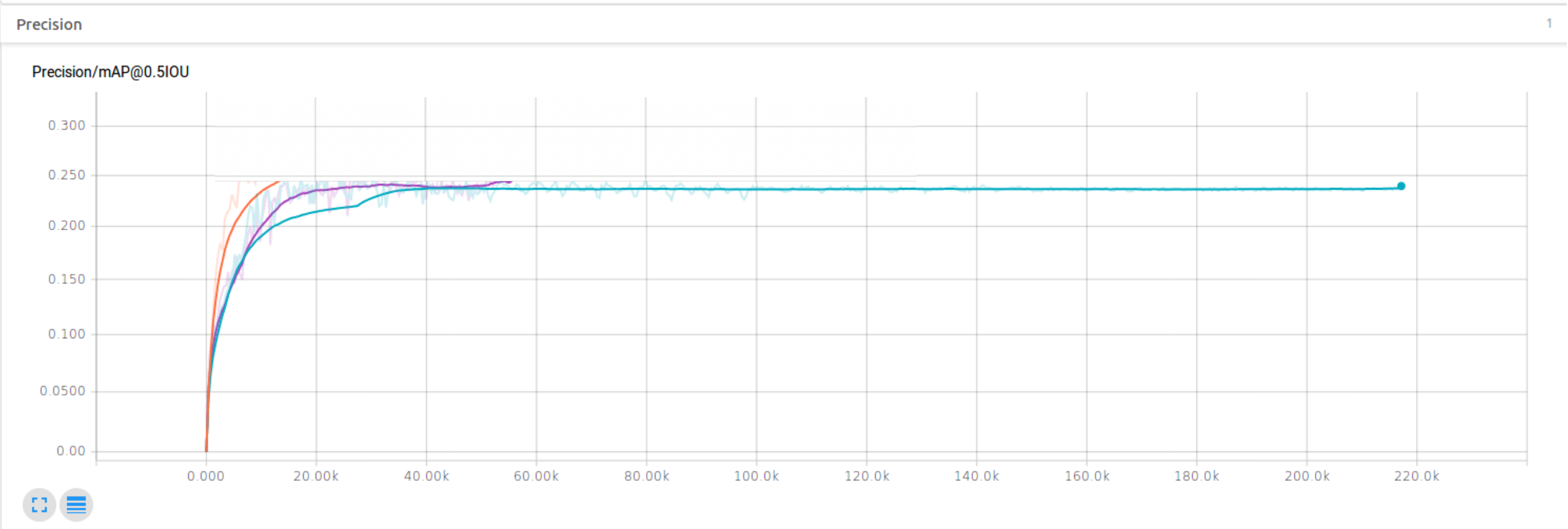


Active Learning

is a powerful paradigm to iteratively develop datasets

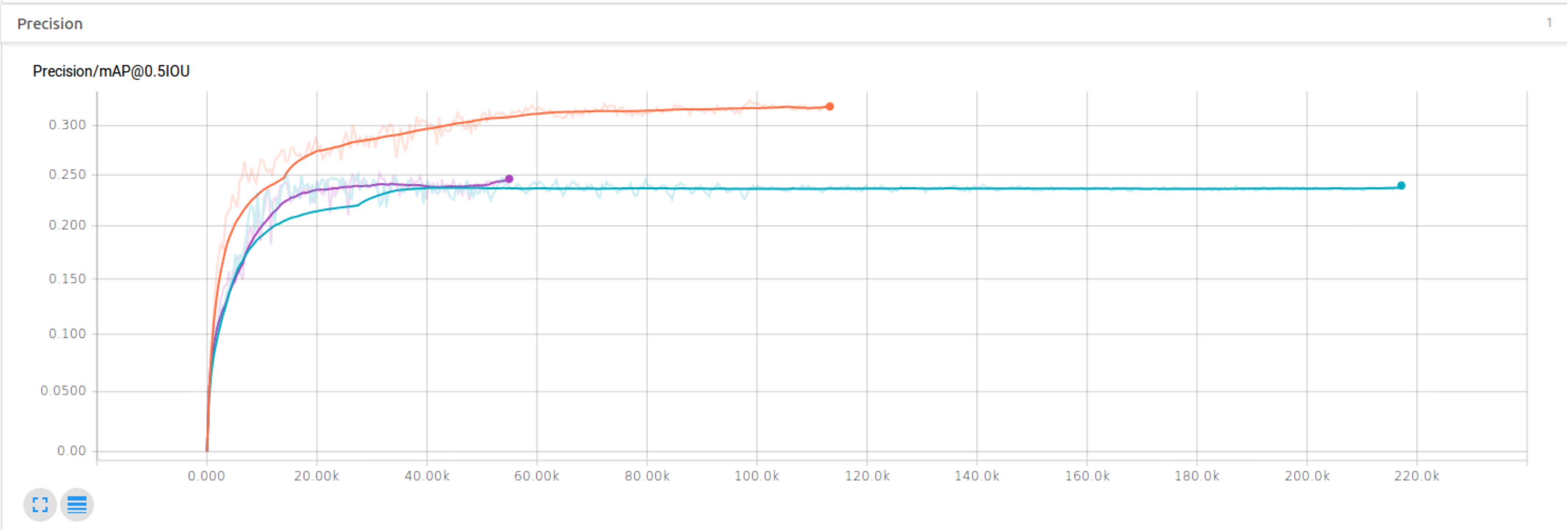
(== develop and debug traditional software)

ADD MORE RANDOM DATA... PLATEAU



Object detection performance. mAP as as function of epochs, for base model (blue), random strategy (purple) and active strategy (orange).

ACTIVE LEARNING => GET OUT OF PLATEAU!



Object detection performance. mAP as as function of epochs, for base model (blue), random strategy (purple) and active strategy (orange).

WHY? NOT ALL DATA CREATED EQUALLY

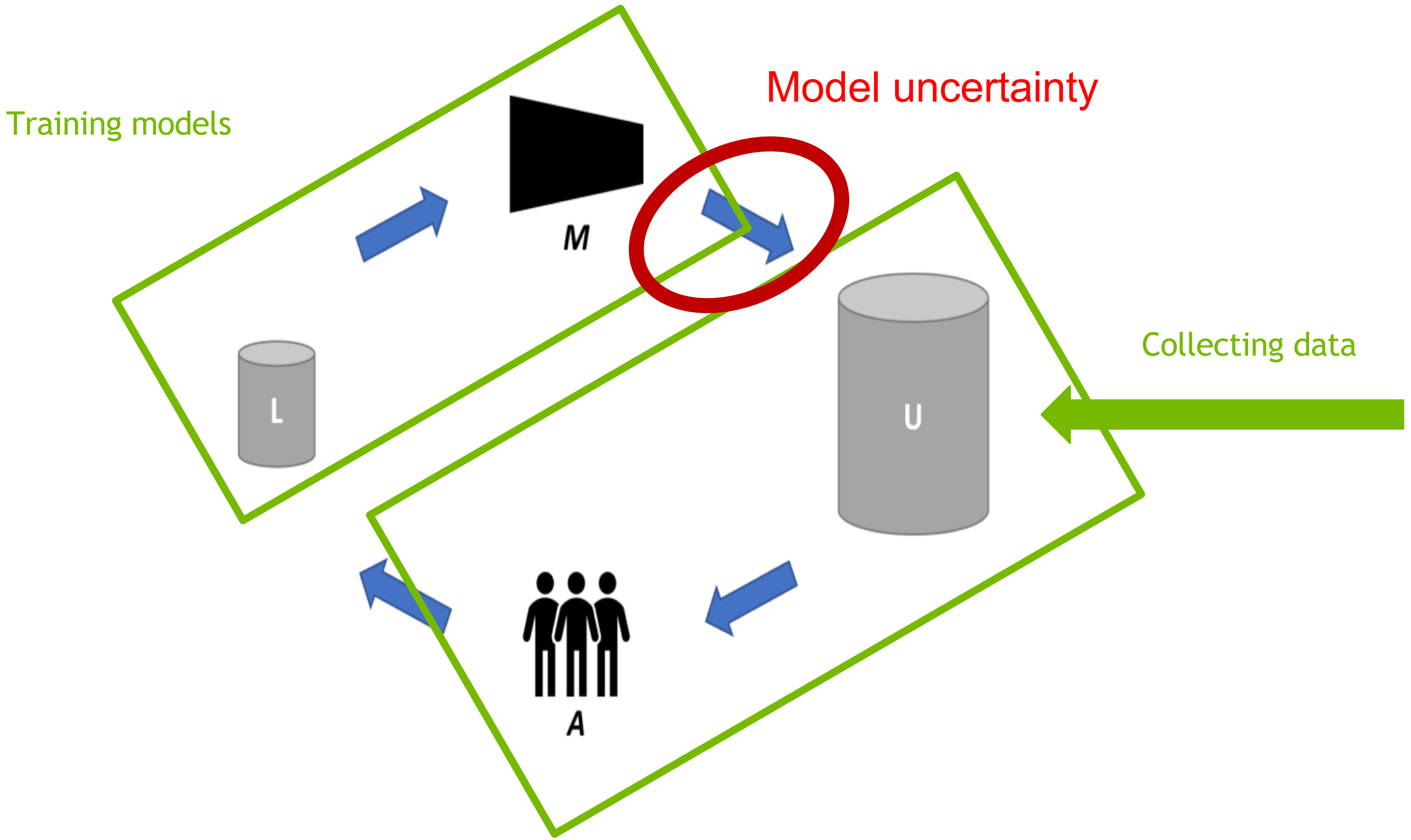
VS

Some Samples Are Much More Informative Than Others

1. How do we find the **most informative** unlabeled data to build the right datasets the fastest?

2. How do we build training **datasets that are 1/1000 the size** for the same result?

HOW ACTIVE LEARNING WORKS

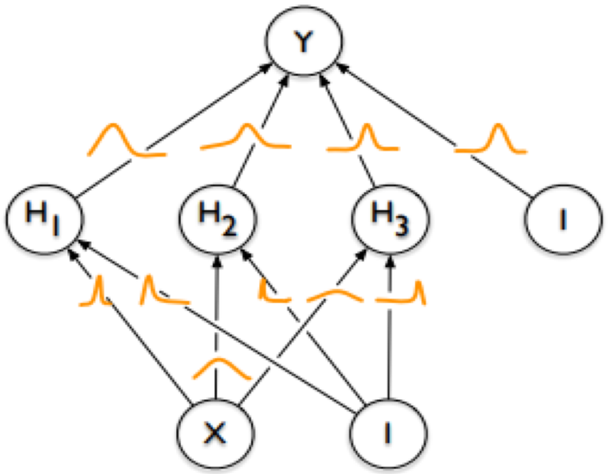


ACTIVE LEARNING NEEDS UNCERTAINTY

Bayesian Deep Networks (BNN)

Bayesian networks are the principled way to model uncertainty. However, they are computationally demanding:

- Training: Intractable without approximations.
- Testing: distributions need ~100 forward passes (varying the model)



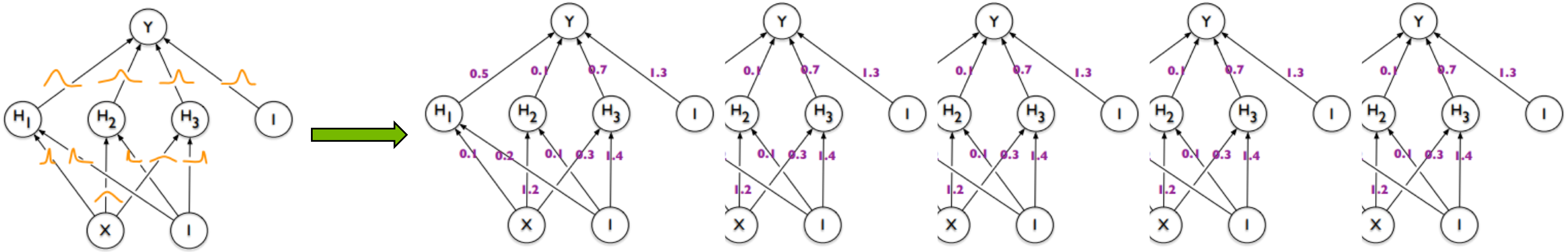
NVIDIA

OUR ACTIVE LEARNING APPROACH

Our approximation to BNN

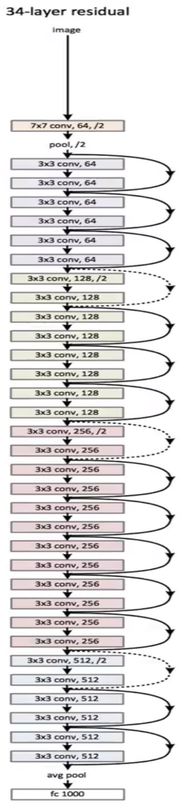
We proposed an **approximation** to BNN to train a network using ensembles:

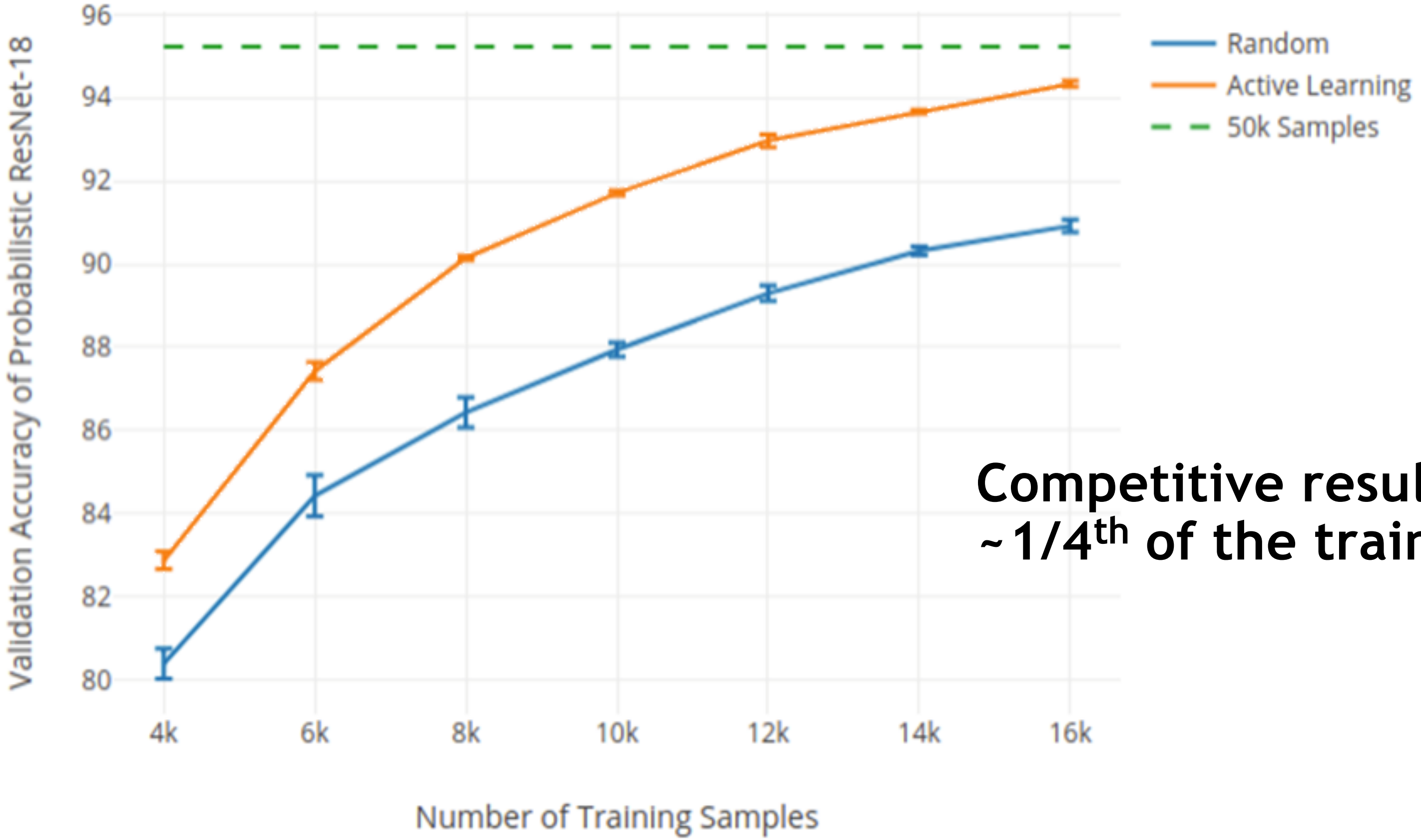
- Samples from the same distribution as the training set will have consensus while other samples will not.
- We regularize the weights in the ensemble to approximate probability distributions.



OUR ACTIVE LEARNING RESULTS

Quantitative Results on CIFAR10

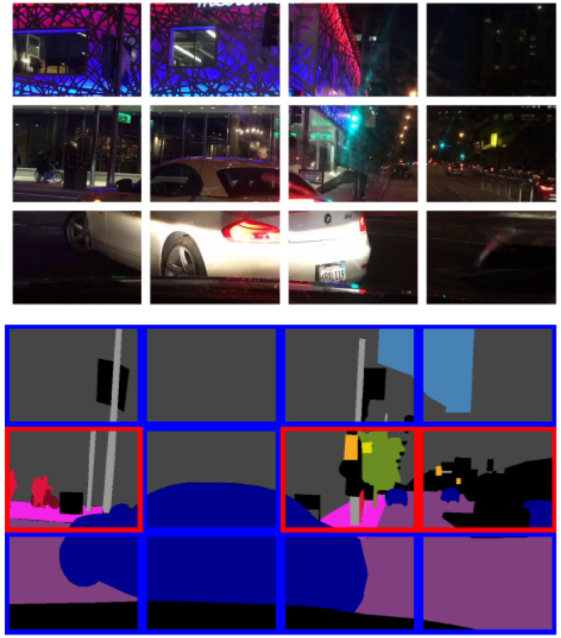


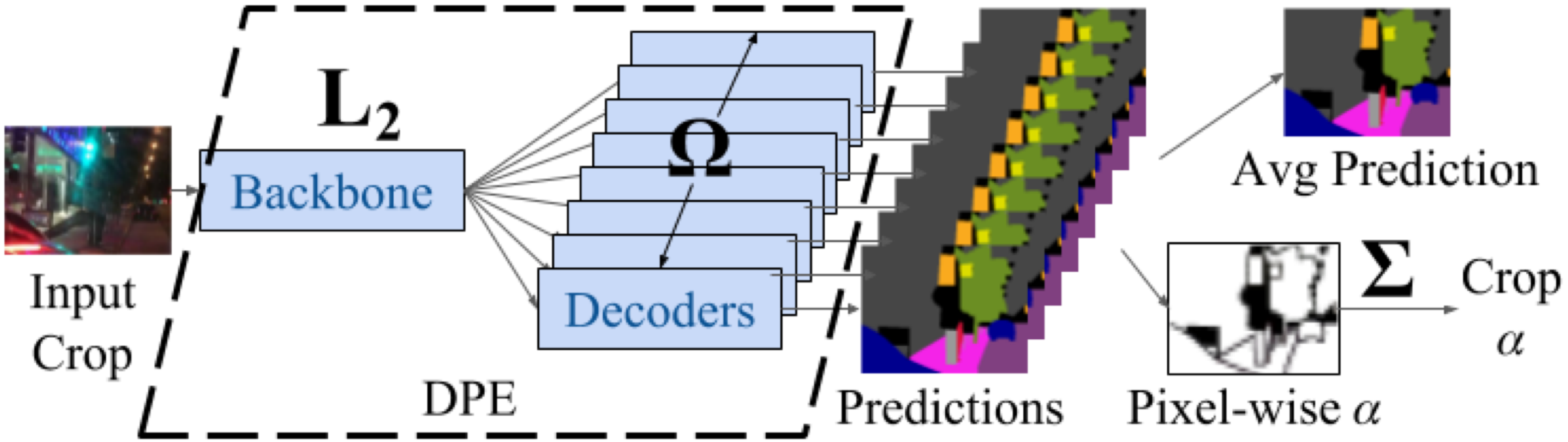


Competitive results using ~1/4th of the training data

OUR ACTIVE LEARNING RESULTS

Applied to more challenging problems like semantic segmentation





[*Chitta, Alvarez, Lesnikowski*], Deep Probabilistic Ensembles: Approximate …. (published at NeurIPS 2018 Workshop on Bayesian Deep Learning)

Getting **active learning to scale**
to the SDC problem is a massive challenge!

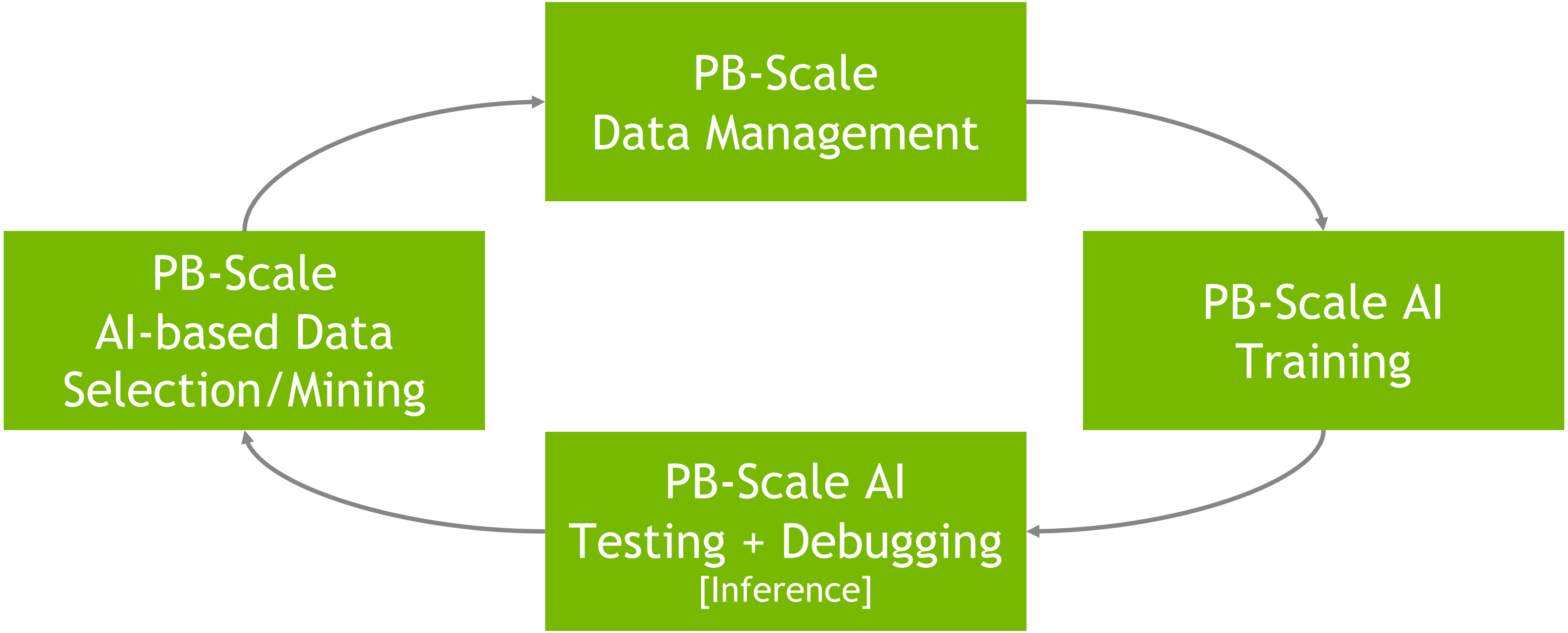
But it is also necessary: labeling cost, data collection and storage cost, training cost.

Project MagLev

NVIDIA's internal production grade ML infrastructure

MAGLEV

Goal: enable the full iterative ML development cycle (e.g. active learning), at the scale of self-driving car data.



MAGLEV COMPONENTS

UI/UX/CLI

Dashboard for MagLev experience, visualizing results, spinning up notebooks, sharing pipelines, data exploration / browsing

Datasets

"Storing, tracking and versioning datasets"

Artifacts and volumes management

Data traceability

ML Data representation

ML Data querying - Presto / Spark / Parquet

Workflows

"API and infra to describe and run workflows, manually or programmatically"

Workflow Infra/Services

Workflow Traceability

ML Pipelines

Persistence / Resuming

Experiments

"Track and view all results from DL/ML experiments, from models to metrics"

Results Saving

Metrics Traceability

Results Analysis

Hosted Notebooks

HyperOpt parameter tracking and sampling

Apps

"Python Building blocks to rapidly describe DL/ML apps, access data, produce metrics"

Read/Stream/Write data for DL/ML apps

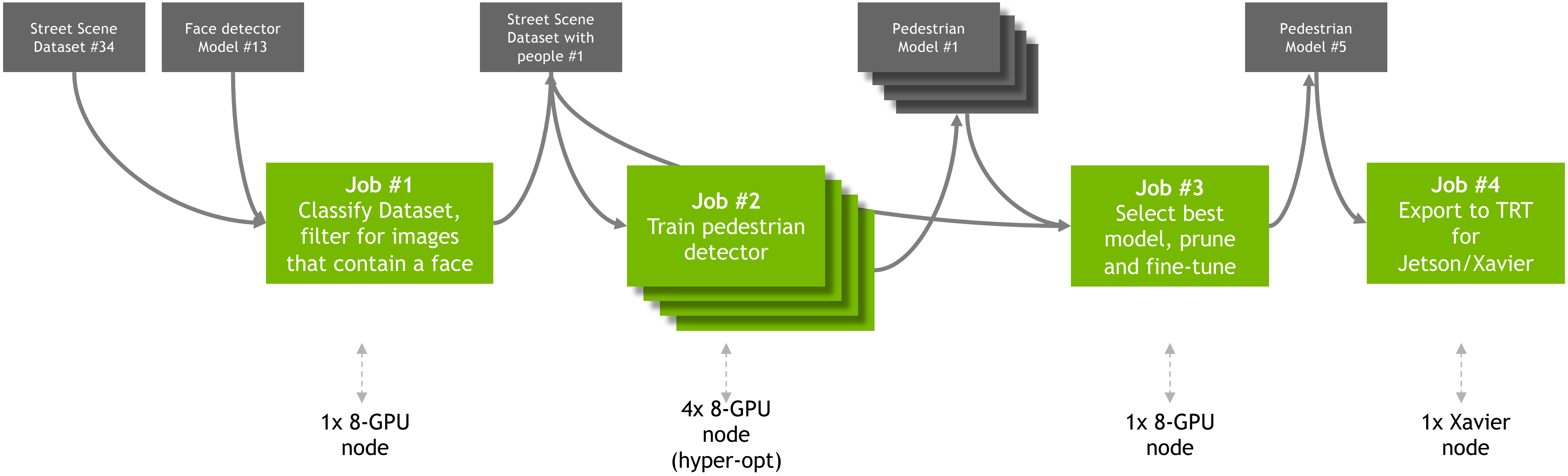
Off-the-shelf models

Generic vertical (AV/Medical/...) operators

Pruning, Exporting, Testing

WORKFLOWS IN MAGLEV

Workflow = directed graph of jobs.
Each job is described by inputs and outputs: datasets and models.
Datasets and models 1st-class citizens, tracked/versioned assets.



WORKFLOWS IN MAGLEV

Step 1: Define the workflow as a list of steps in a YAML file

```yaml
localImage: dlav/common/image
steps:
- name: 0-train
  completions: 1
  gpus: 1
  command: drivenet train -e default_spec.txt -r /out
- name: 1-evaluate
  gpus: 1
  inputs:
      /in/0-train: {step: 0-train}
  command: drivenet evaluate -e default_spec.txt -m /in/0-train/weights/model.hdf5
```

Step 2: Execute the workflow

```
maglev run //dlav/common:workflow -- -f my.yaml -e saturnv -r <results dir>
```

EXAMPLE WORKFLOW: FIND BEST MODEL

Improving DNNs through massively parallel experimentation

Experiments are run is parallel as part of a predefined workflow

Define Workflow

Parallel Experiments

Model 1

Model 2

Model 3

Model 4

.

.

Model 50

Example:

- Run the 50 jobs in parallel
- Use 8 GPUs per job
- Total time → 1 day

Pick Best Model

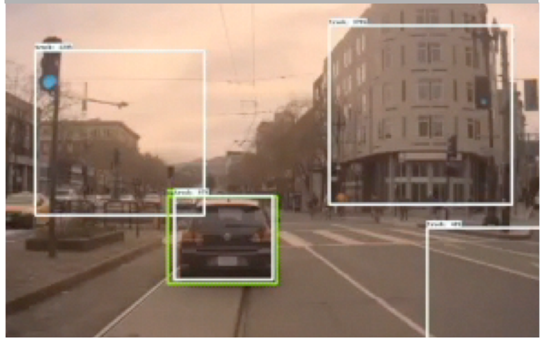
Prune

Re-train

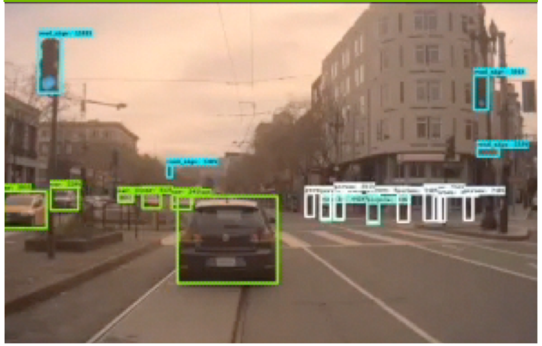
Evaluate

New experiment set parameters

Random hyper-parameters



Optimal hyper-parameters



NVIDIA

MAGLEV SERVICES

Runs on Kubernetes

Hybrid deployment:
1/ service cluster on AWS
2/ compute cluster at NVIDIA (SaturnV)

Multi-node training via MPI over k8s

Dataset management, versioning

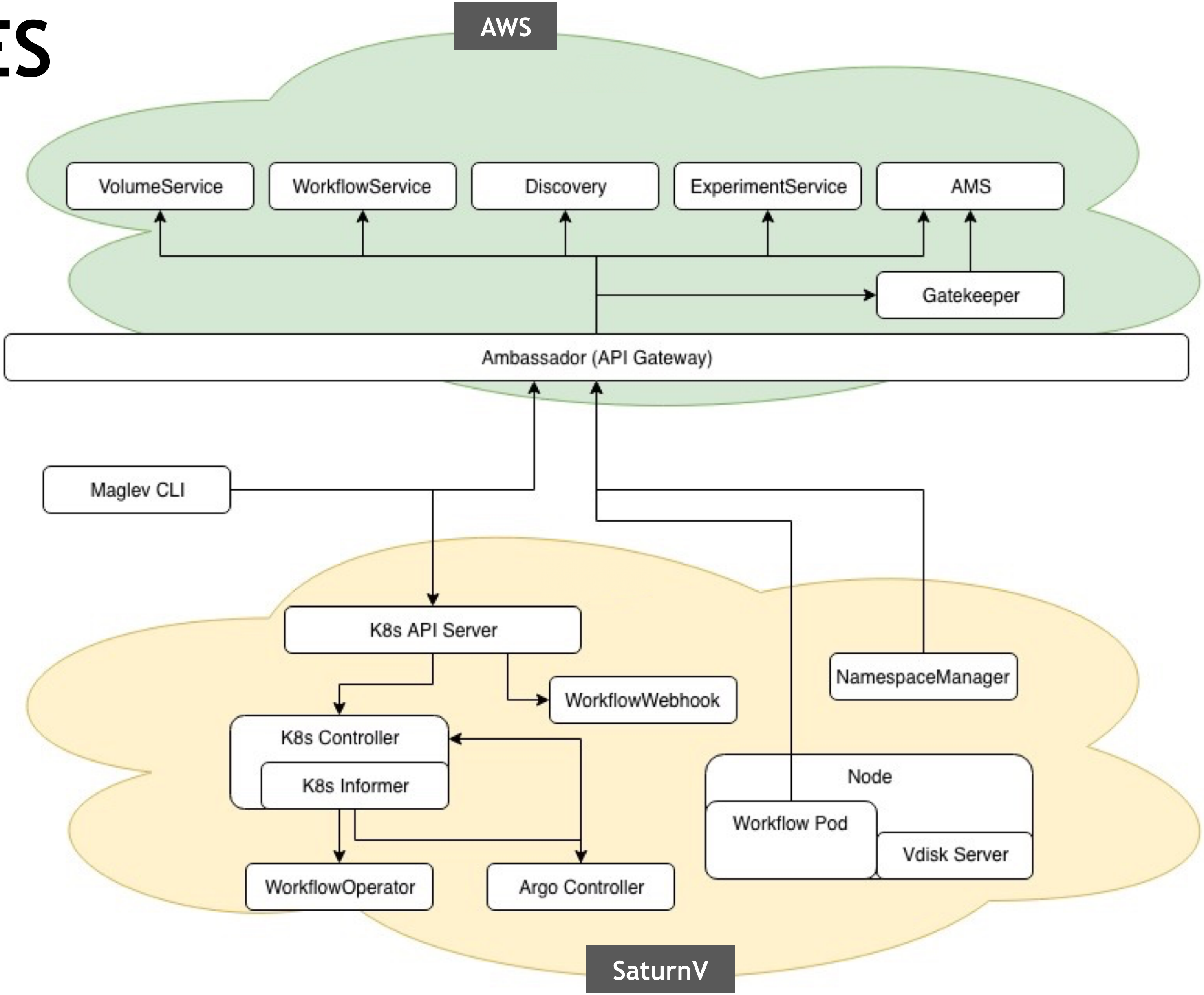
Workflow engine, based on Argo

Experiments management, versioning

Leverages
NVIDIA TensorRT
for inference

TensorRT

Leverages
NVIDIA GPU Cloud
Containers for
Pre-built DL/ML containers



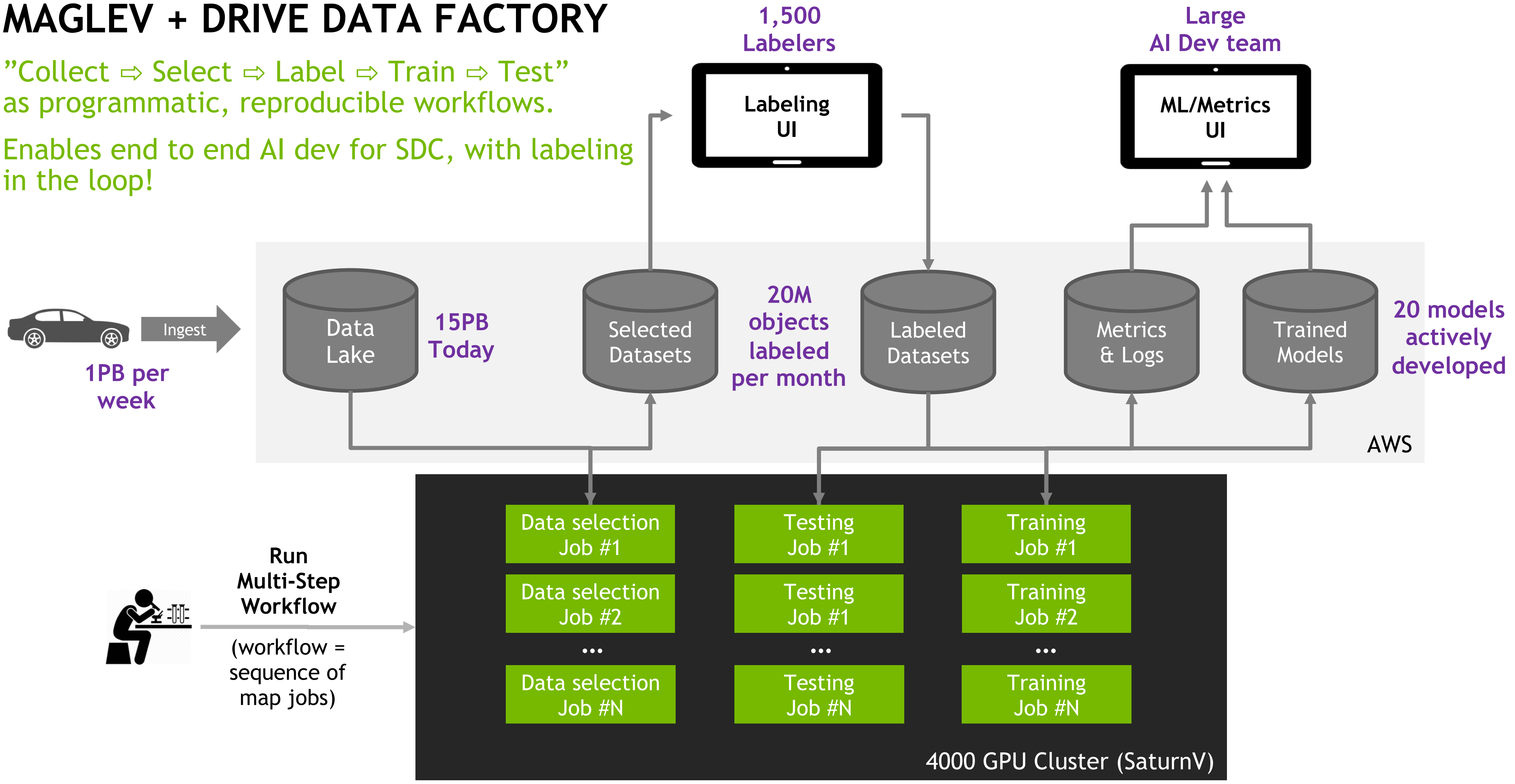
MagLev + DRIVE Data Factory

End to end infrastructure to support AI development for DRIVE

MAGLEV + DRIVE DATA FACTORY

"Collect ⇨ Select ⇨ Label ⇨ Train ⇨ Test"
as programmatic, reproducible workflows.

Enables end to end AI dev for SDC, with labeling in the loop!



NVIDIA

PB-SCALE DATA MANAGEMENT FOR DRIVE

Or how to build and feed datasets into workflows

Build and compose PB datasets

SSD Upload Store → DriveWorks Parser → Data Warehouse → Cloud Storage / Meta-Storage

Query Engines
- Presto
- Spark
- Hive

Clients
- Data+DNN Metrics
- Export Service
- Web-based SQL UI
- Notebooks
- HumanLoop

DriveWorks

- Using DriveWorks recorder app, capture 7-12 cameras in RAW, 3 LIDARs, 8 RADARs, IMU, GPS, CAN
- In-house “TransferPilot” systems automate upload from SSD to S3
- Spark ETL jobs parse data, index key metadata, and create Parquet files query-able by Presto database
- Custom clients and general SQL interface allow developers access to full dataset

DL App dataset consumption

Pipelines
- Load/Decode
- Batching
- Preprocessing
- Augmentation
- Ground-truth rasterization
- ...

Train

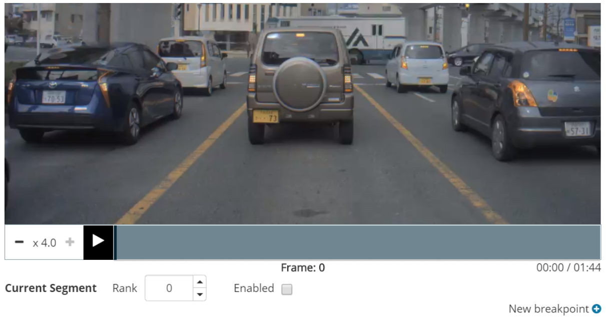
Pure Python-programmable client

DRIVE DATA CURATION

Finding the most valuable data to label or test

Manual Curation

Human labelers review targeted videos for sections of interest. "Fallback" option used for special scenarios.



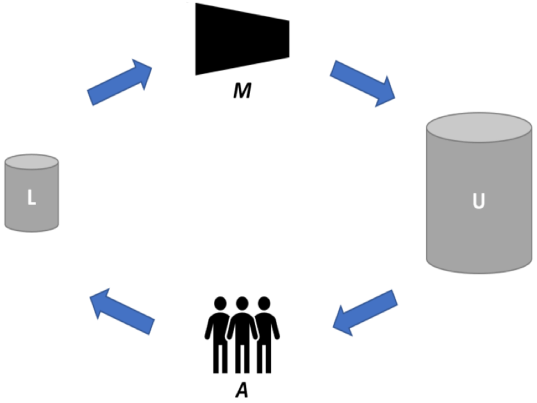
Query-based Curation

Query the lake for any metadata, including CAN signals ("speed > X"), segment tags ("visibility = raining"), or map data (example below: intersection = true)

frame	session_id	lat	lon	timestamp
	0cc01eca-9162-5a77-9470-d19cc26a9c77	36.08	-115.15	1,506,069,616,000
	33382bca-e1d9-514e-a18f-44d632762922	45.47	9.18	1,519,924,541,000

Active Learning

Evaluate a pretrained model on unlabeled data and see where it is uncertain. Label those "confusing" images.



DRIVE DATA LABELING

Maximize throughput and quality

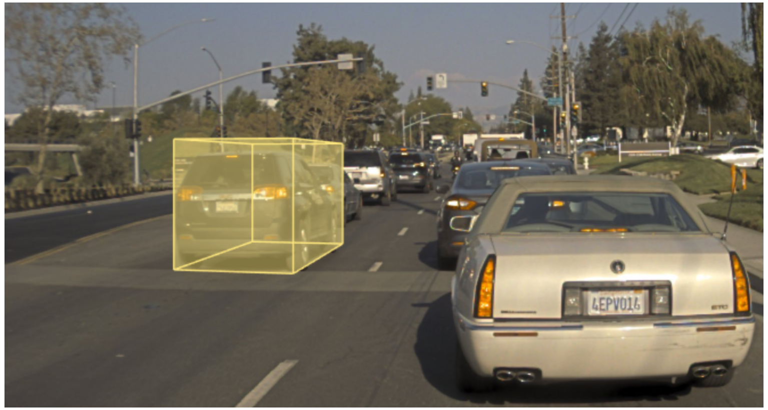
Every label is annotated and QA'ed by a separate professional labeler, with random expert audits to ensure consistency.

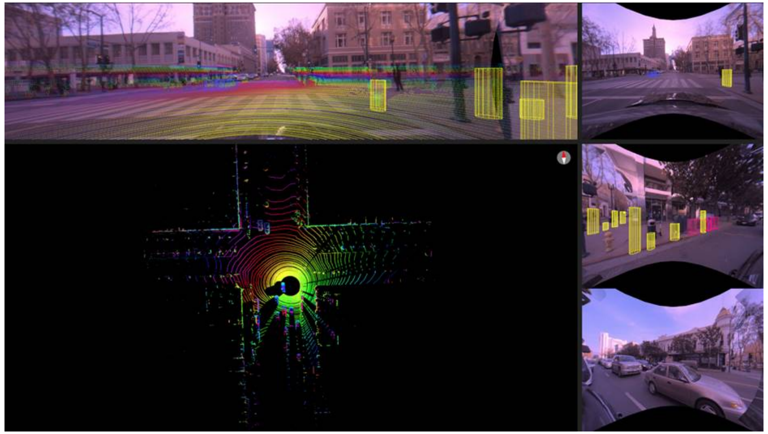
~1 million frames/crops labeled and QA'ed each month by a team of 1500+ labelers.

All done in HumanLoop, an web-based platform supporting:

- Bounding boxes (and cuboids)
- Instance segmentation
- Polyline annotations
- Object tracking in videos
- Hierarchical classification

50 unique active labeling projects today, covering project categories => 14+ DNNs





NVIDIA

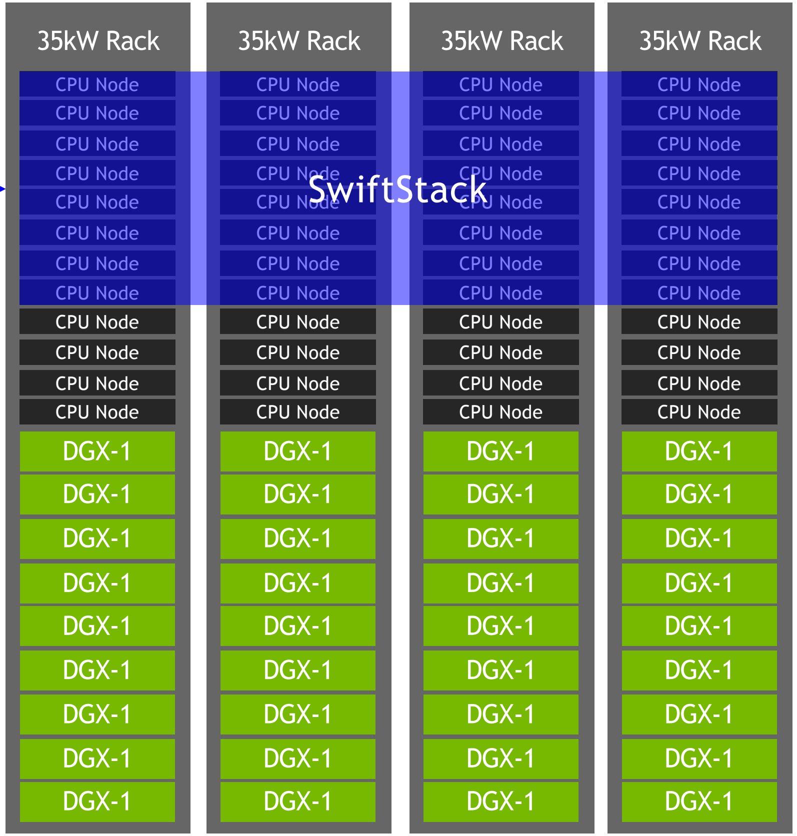
MAGLEV DEPLOYMENT | HW INFRA

MagLev Services

Services on AWS

EC2 ... S3

Kubernetes



Multi-PB Datasets stored on AWS S3

High-bandwidth interconnect to replicate them locally, on SwiftStack (S3-compat object API)

In-rack bandwidth between storage and DGX optimized for all our workloads (inference/mining and training)

Each rack:
9 DGX-1 = 72 TESLA V100 GPUs = 9 PFLOPs
1PB of object storage

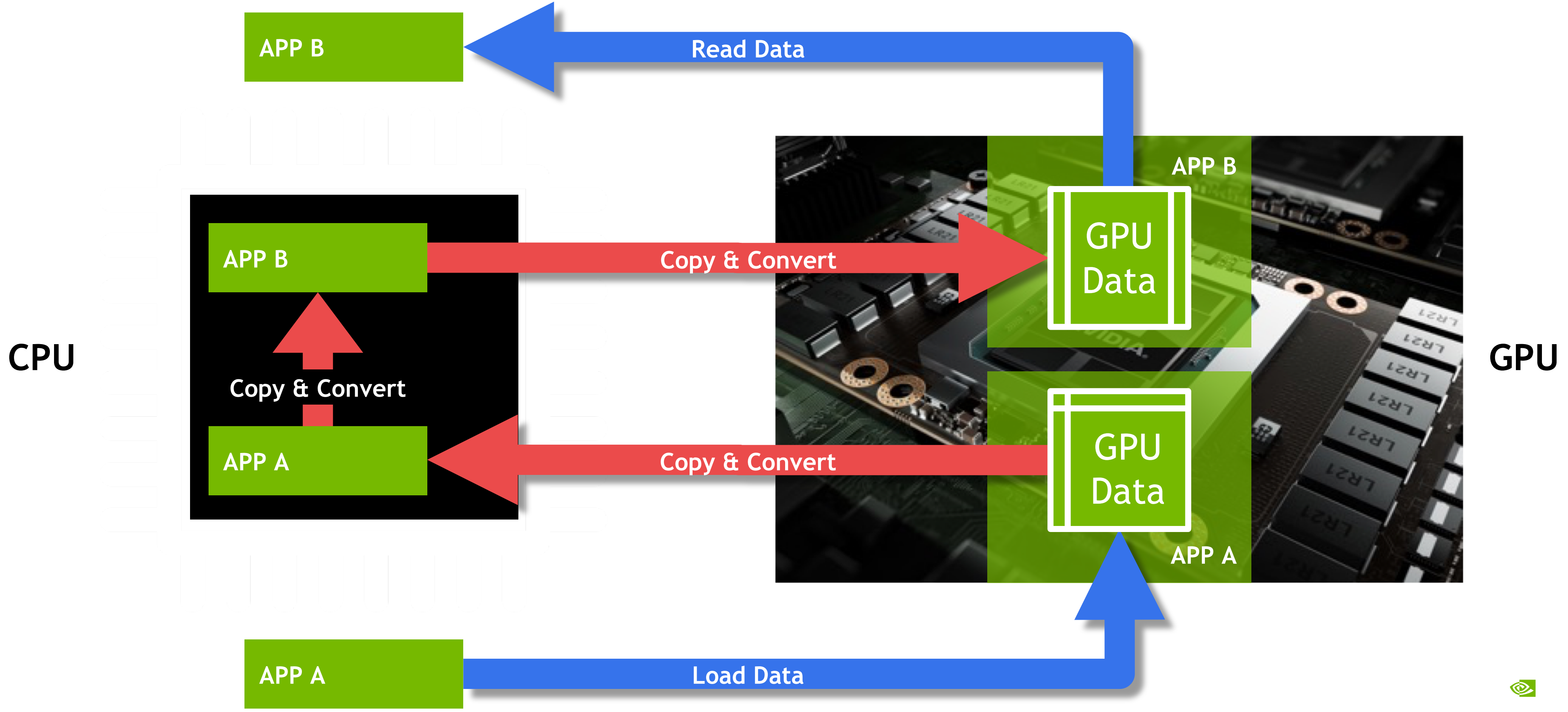
What if we could push data even closer to compute?

RAPIDS

Since building datasets is such an important part of the ML workflow… looks like we should move it to the GPU as well ☺

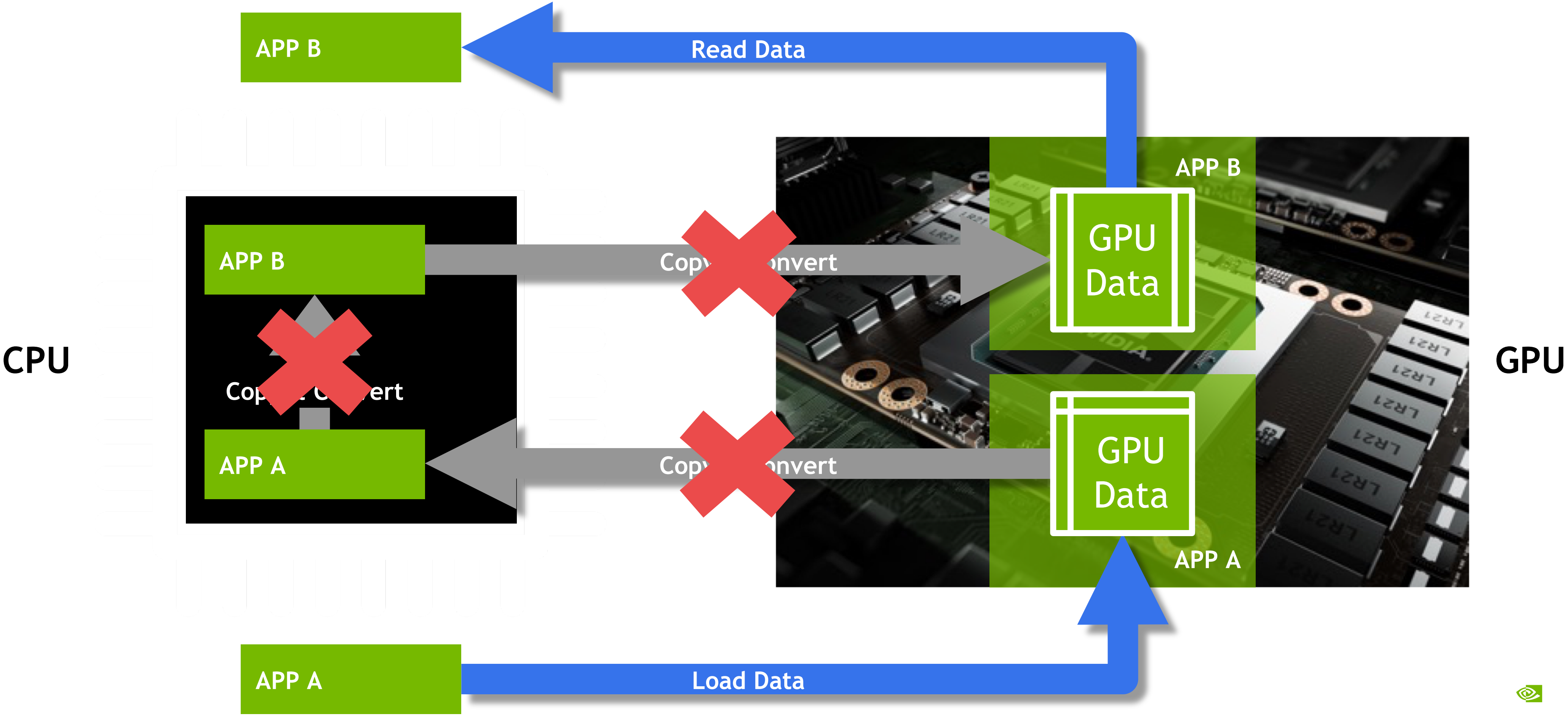
DATA MOVEMENT AND TRANSFORMATION

The bane of productivity and performance

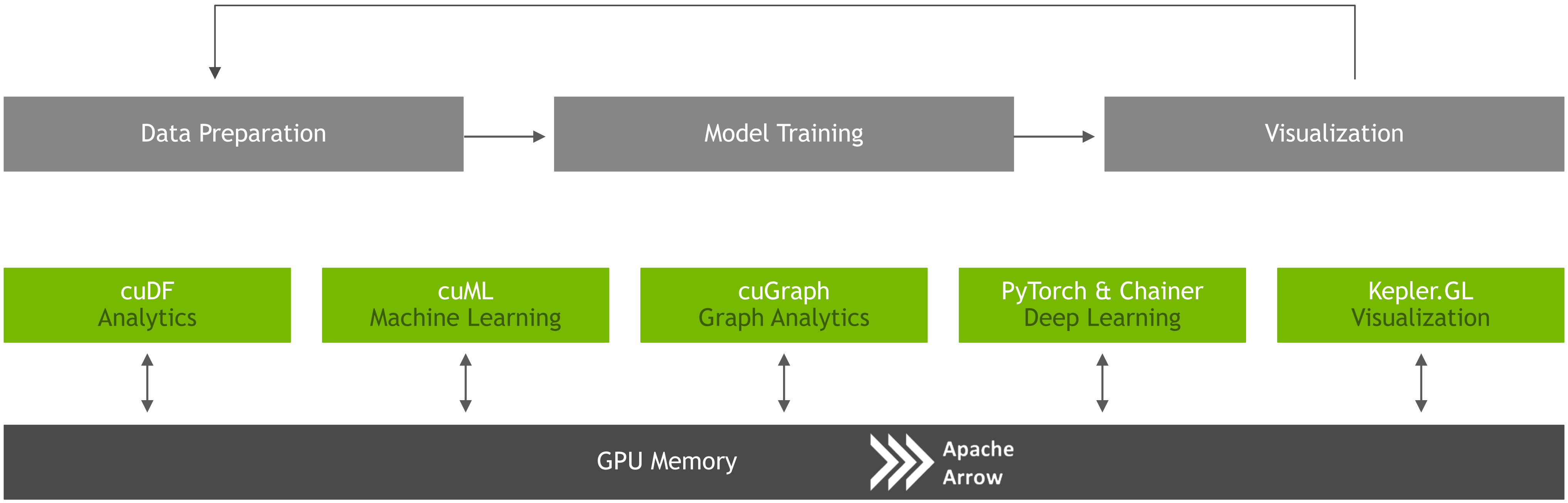


DATA MOVEMENT AND TRANSFORMATION

What if we could keep data on the GPU?

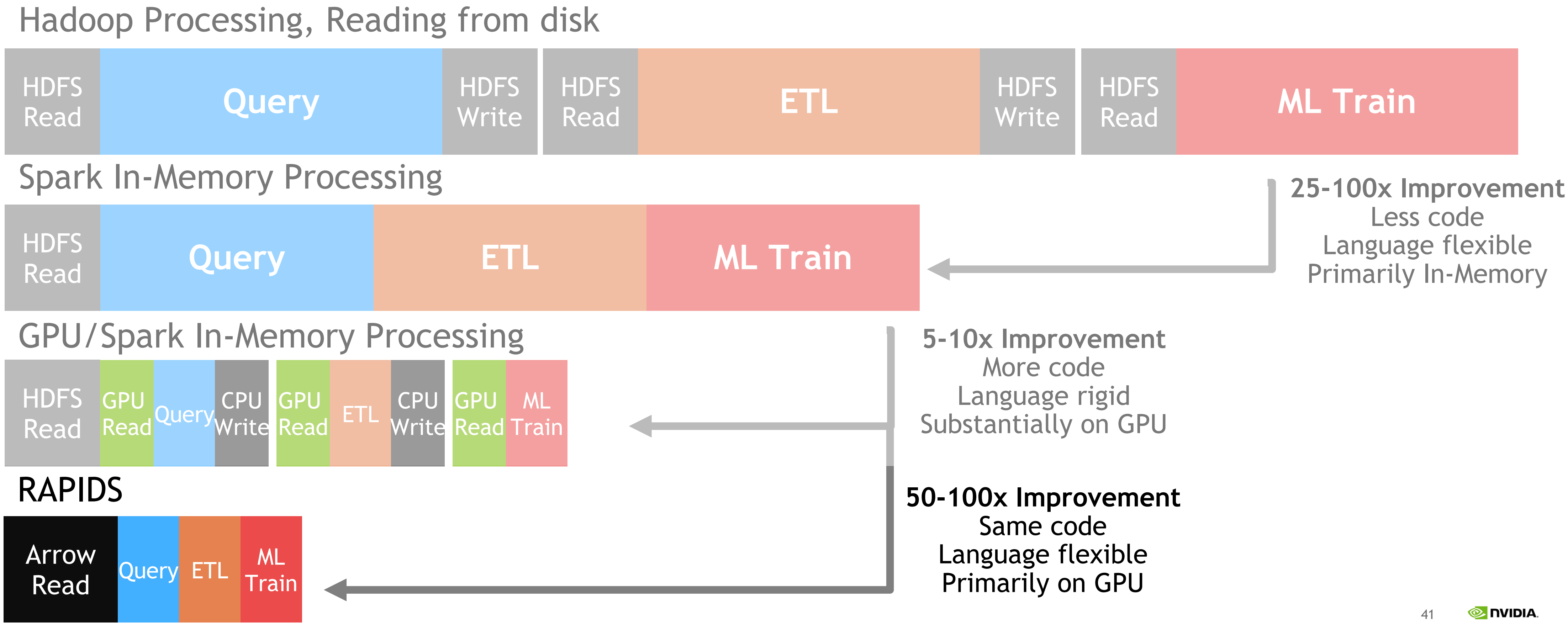


RAPIDS: END TO END DATA SCIENCE



DATA PROCESSING EVOLUTION

Faster Data Access = Less Data Movement



RAPIDS, NGC, TENSORRT

How do I get the software?

RAPIDS: rapids.ai
Github: github.com/rapidsai
Conda: anaconda.org/rapidsai
Pip (soon): pypi.org/project/[cudf,cuml]
NVIDIA GPU Cloud: ngc.nvidia.com/registry/nvidia-rapidsai-rapidsai
Docker: hub.docker.com/r/rapidsai/rapidsai

NVIDIA GPU CLOUD: ngc.nvidia.com
TENSORRT: developer.nvidia.com/tensorrt

LEARN MORE

Many other exciting sessions about our AI Infrastructure

S9613	Wed 10:00am	Deep Active Learning	Adam Lesnikowski
S9911	Wed 2:00pm	Determinism In Deep Learning	Duncan Riach
S9630	Thu 2:00pm	Scaling Up DL for Autonomous Driving	Jose Alvarez
S9987	Thu 9:00am	MagLev: NVIDIA’s Production-grade AI Platform	Divya Vavili, Yehia Khoja
S9577	Tue 9:00am	RAPIDS: The Platform Inside and Out	Josh Patterson

THANK YOU

rapids.ai
ngc.nvidia.com
github.com/rapidsAI
twitter.com/nvidiaAI
twitter.com/rapidsAI
twitter.com/datametrician
twitter.com/clmt