

The Future of Deep Learning with Deep Lake



Executive Summary

One out of three ML projects fail due to the lack of a solid data foundation. Projects suffer from low-quality data, under-utilized compute resources, and significant labor overhead required to build and maintain large amounts of data. For projects involving tabular data, traditional data lakes provide critical features such as time traveling, SQL queries, ingesting data with ACID transactions, and visualizing terabyte-scale datasets for analytical workloads. These features break down data silos, enable data-driven decision making, improve operational efficiency, and reduce costs across organizations. However, most of these features are not available for deep learning workloads.

Deep Lake maintains the benefits of a vanilla data lake with one key difference: it stores complex data such as images, videos, annotations, as well as tabular data, as columns¹, and it rapidly streams the data to deep learning frameworks without sacrificing GPU utilization. As deep learning rapidly takes over traditional computational pipelines, storing datasets in a Deep Lake is becoming the new norm.

¹ The exact term for this kind of mathematical representation is known as tensors. Tensors are fundamental data structures used by deep learning systems.

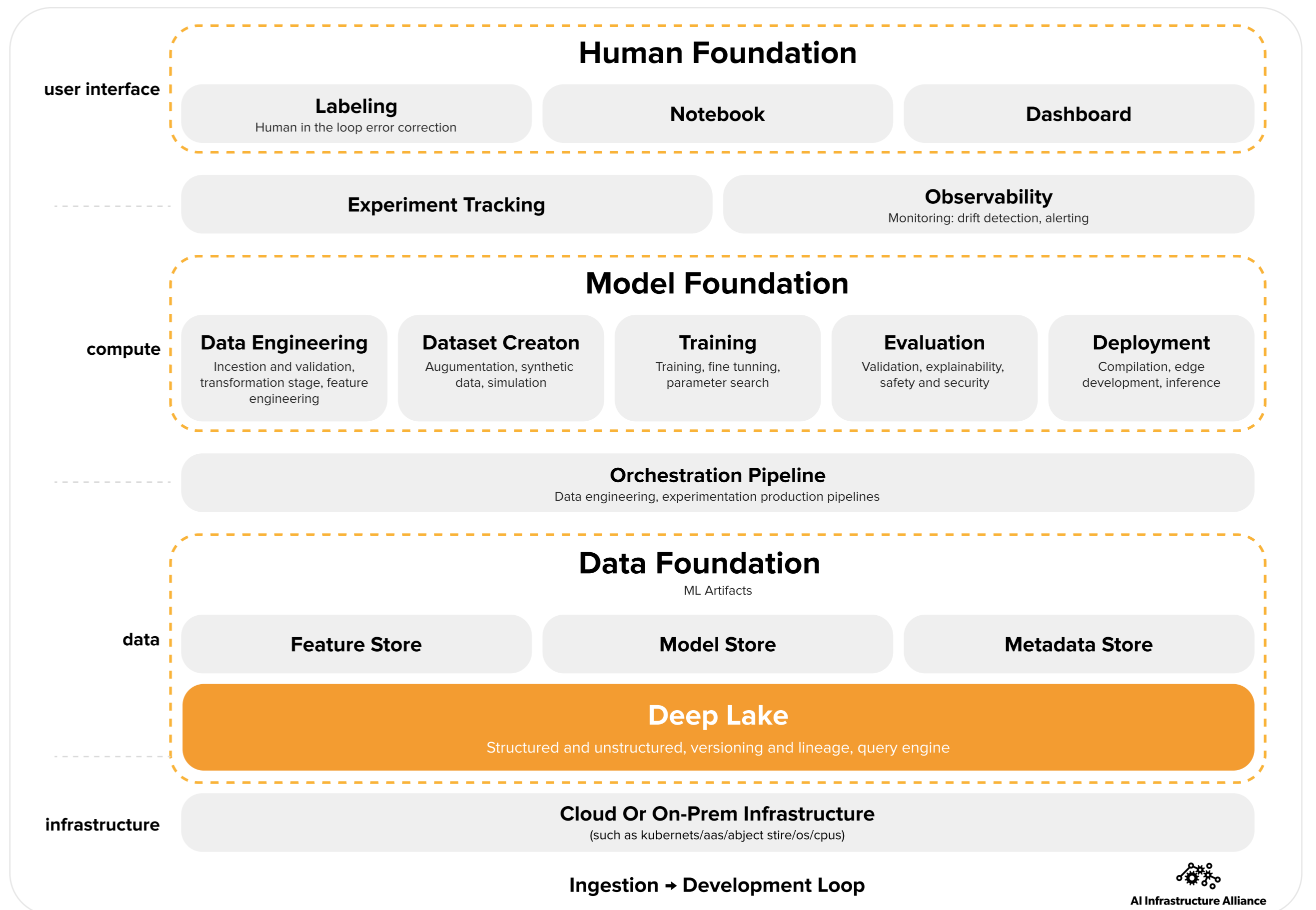
Sections

Introduction: Deep Lake, Data Lake for Deep Learning	1
Why use a Data Lake?	2
What is a Deep Lake?	3
Deep Dive into Deep Lake	4
Version Control: Git for data	5
Visualize: In-browser visualization engine	6
Query: Rapid queries with Tensor Query language (TQL)	7
Materialize: Format native to deep learning	8
Stream: Streaming Data Loaders	9
Integrating Modern Data Stack and MLOps	10
How Deep Lake is Revolutionizing Deep Learning	11
Alternatives to Deep Lake	12
How to Get Started with Activeloop for your Deep Lake	13
Conclusion	14

Introduction: Deep Lake, Data Lake for Deep Learning

Solid data infrastructure is required for maintaining the operational functionality of deep learning models. To support rapid model iterations, data scientists need to connect disparate data sources, identify important data samples and failure modes.

Automation of dataset engineering activities include cleaning, transforming, low latency data retrieval, dataset versioning, and detecting data drift. Furthermore, the infrastructure should provide data provenance, quality control, and integrate with external annotation tools.



Since there are no off-the-shelf tools for deep learning data infrastructure, nor there is a standard data format, organizations resort to building in-house solutions. Large companies such as Meta, Tesla, and Porsche spend years and millions of dollars to build this software, and they require dozens of highly-paid data engineers to maintain it.

At smaller companies, sometimes unknowingly, data scientists spend most of their time building ad-hoc solutions while getting distracted from doing what they do best - training models, pushing them into production, shipping AI features into their products, and solving core business problems.

Activeloop enables Machine Learning teams to ship AI products faster by replacing complex data infrastructure with a data lake for deep learning applications called **Deep Lake** (an architectural blueprint for managing Deep Learning data at scale—part)

Why use a data lake?

Gartner identified the 3 "V"s of modern data-centric systems and why traditional databases and legacy Enterprise Data Warehouses (EDW) are ill-suited for them. The three "V"s are *Volume*, *Variety*, and *Velocity*. *Volume* refers to the large size of modern data systems that dwarfs anything that came before. Much of that data is in semi-structured and unstructured formats, requiring data storage to be flexible in its schema or even schema-less to accommodate the *variety* of data (*Variety*). In its turn, *Velocity* indicates that data arrives in real-time, sometimes at a very high rate, and actionable insights should be inferred in real-time.

Data Lakes are an attempt to leverage modern storage technologies to address these limitations, particularly in the cloud. Data Lakes allow organizations to overcome the technical limitations traditional databases impose on the size (*Volume*) of data while also allowing semi-structured and unstructured data (*Variety*) to be stored alongside structured data.

As data lakes enable data to be appended in real-time without requiring agonizing ETL workflows, the data can be analyzed in real-time (*Velocity*).

They also provide a way to break down data silos and gain insights that were previously hidden in disparate data sources.

Data lakes are usually built on top of cloud-based storage services, such as Amazon S3 or Azure Blob Storage.

This allows organizations to take advantage of the scalability, durability, and cost-effectiveness of these services. Data Lakes typically rely on standard tabular formats such as Parquet or Avro, and additionally integrate with other cloud-based services, such as Amazon Athena or Presto, to provide powerful data processing and analytical capabilities. There are many reasons why organizations use data lakes, such as:

Benefits

Breaking down data silos: Data lakes provide a central repository for all data, making it easier to break down data silos and gain insights that were previously hidden in disparate data sources.

Enabling data-driven decision making: Data lakes give organizations the ability to store, process, and analyze large amounts of data, which can be used to support data-driven decision making.

Improving operational efficiency: Data lakes can help organizations improve operational efficiency by providing a centralized platform for storing and processing data.

Reducing costs: Data lakes can help organizations reduce costs by taking advantage of the scalability and cost-effectiveness of cloud-based storage services.

If you are working on analytical workloads such as fraud detection, sales forecasting, or time-series analysis, for example, using a traditional data lake would be a perfect fit. However, they are very limited for deep learning applications because:

Limitations

Images, Videos, and other complex data types are stored independently

There is no native integration with deep learning frameworks

Missing gap between MLOps and Modern Data Stack

Queries focus on analytical workloads rather than dataset building



Enter Deep Lake - The Data Lake for Deep Learning

What is a Deep Lake?

A Deep Lake is a data lake specialized for deep learning use cases where the raw data includes images, videos, audio, and other unstructured data. The raw data are then materialized into deep learning native tensorial storage format and streamed to model training across the network.

Machine Learning with Deep Lake

Deep Lake provides key features that make it the optimal data storage platform for deep learning applications, including:

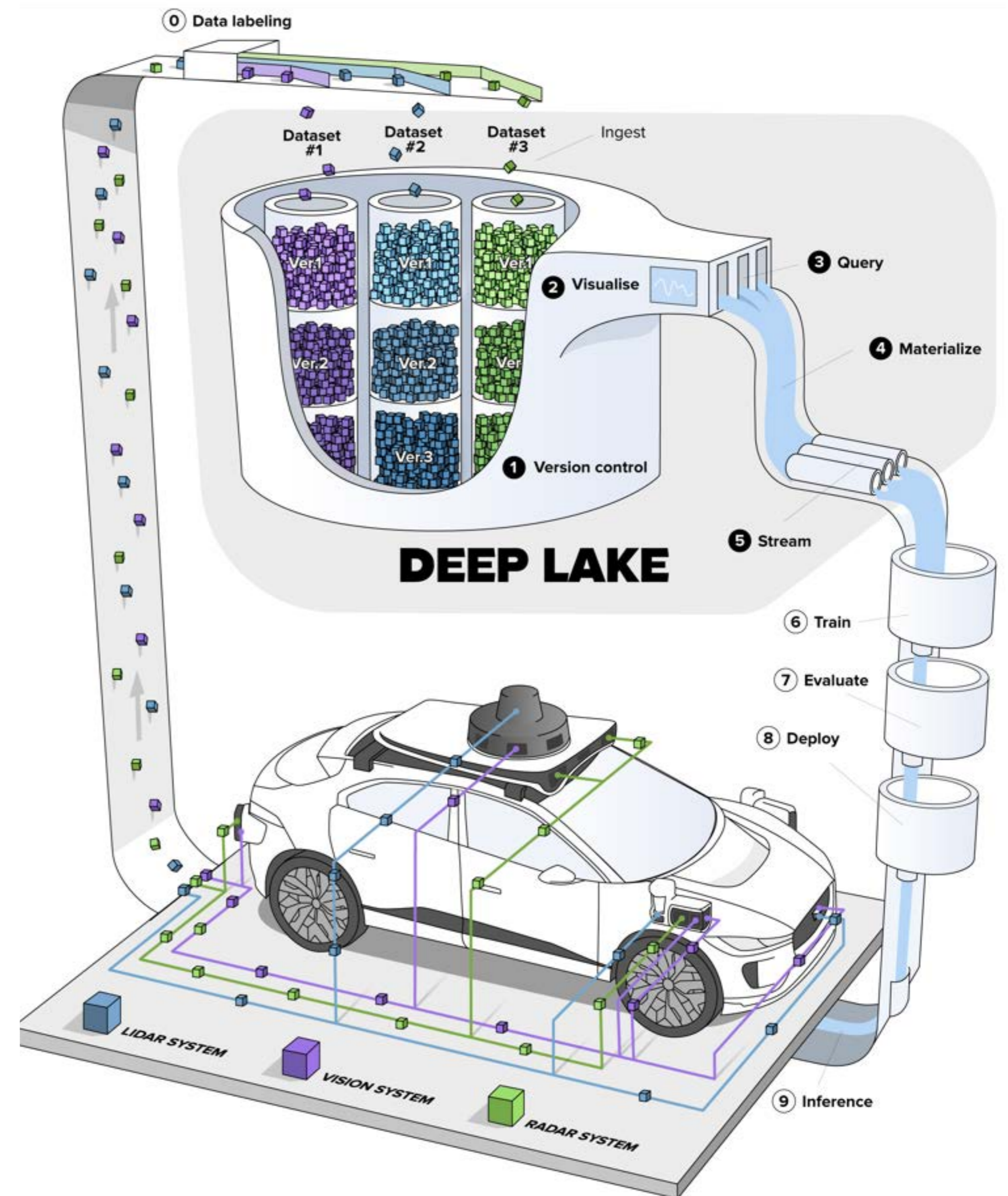
A scalable and efficient data storage system that can handle large amounts of complex data in a columnar fashion

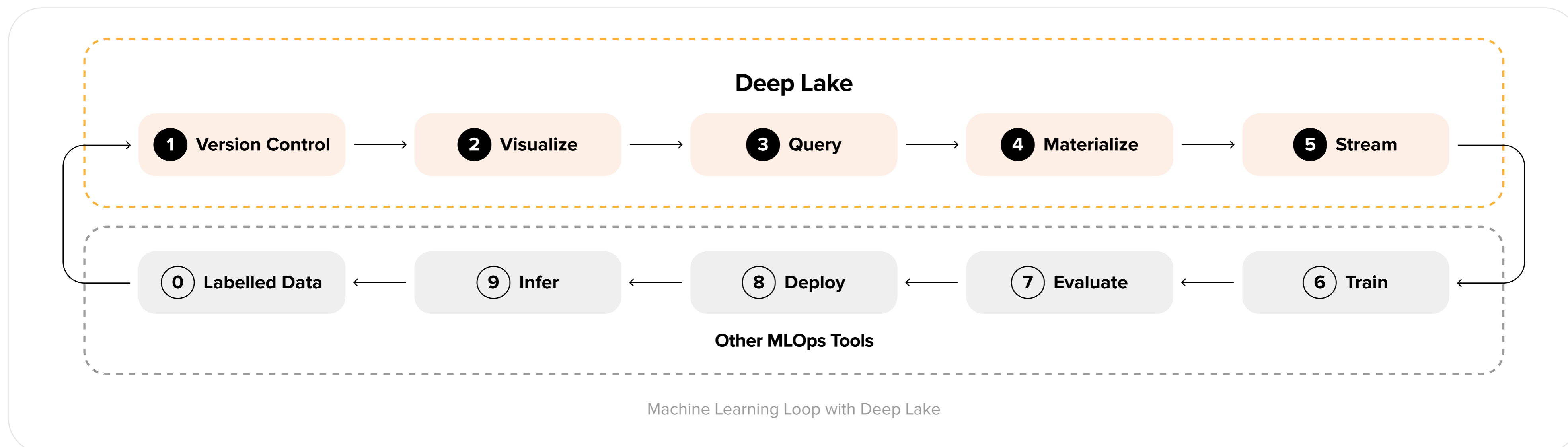
Querying and visualization engine to fully support multimodal data types

Native integration with deep learning frameworks and efficient streaming of data to models and back

Seamless connection with MLOps tools

In short, Deep Lake is an ideal storage platform for deep learning practitioners who are solving real-world problems using massive amounts of data.





Deep Dive into Deep Lake

Deep Lake is a repository of data that is used to train deep learning models. Deep learning is a type of machine learning that uses neural network algorithms to learn from data in a way that resembles human cognition.

Deep learning models learn from vast amounts of data and make better predictions with higher accuracy compared to traditional machine learning models.

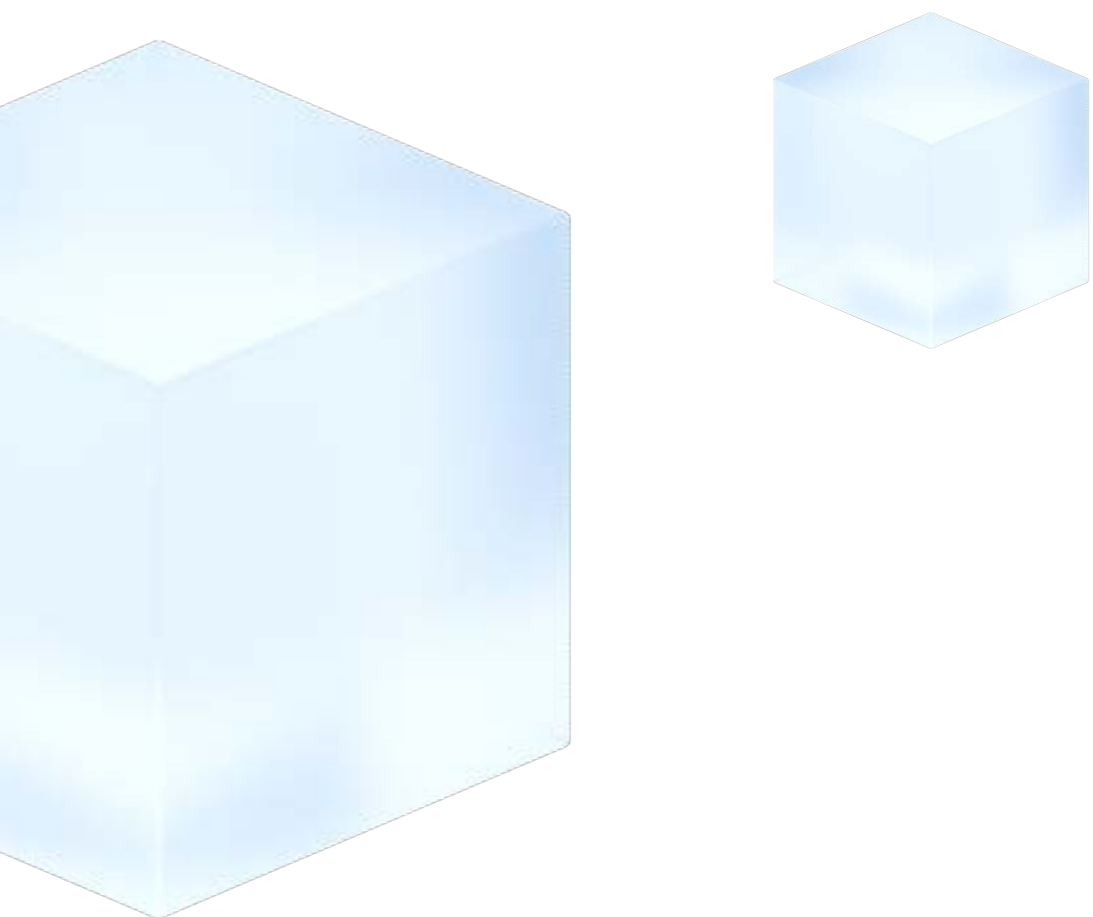
Deep Lake ingests large amounts of raw data in an unstructured form.

The data is collected from a variety of sources, such as raw camera sensors, annotation tools, or simulations. During ingestion they get structured into tensors. Advanced queries enable filtering the useful samples and materialize into deep learning optimized datasets.

Datasets then get streamed to deep learning frameworks while transformed on the fly. Visualization engine enables rapid inspection of data, while queries enable easy finding of edge-cases and transformations of your datasets.

While models generate outputs during training or inference, the data are stored along with inputs for further evaluation. This could be finding prediction mistakes or detecting data drifts.

To ensure full reproducibility of experiments, every modification to the dataset is version controlled and logged with experimentation tracking tools via an integration. Furthermore, Deep Lake integrates with annotational tools and the rest of the MLOps ecosystem to ensure end-to-end compatibility.



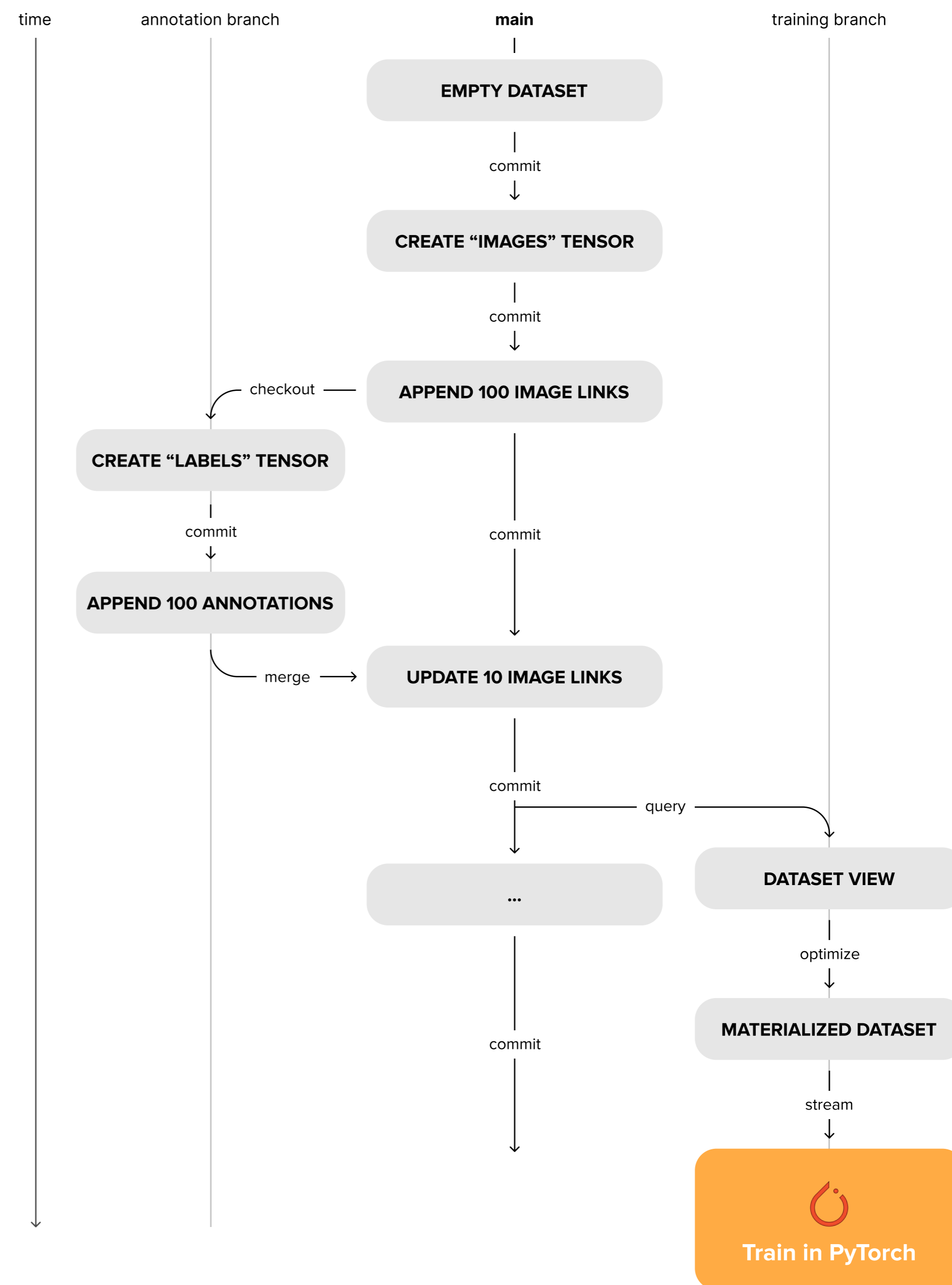
1

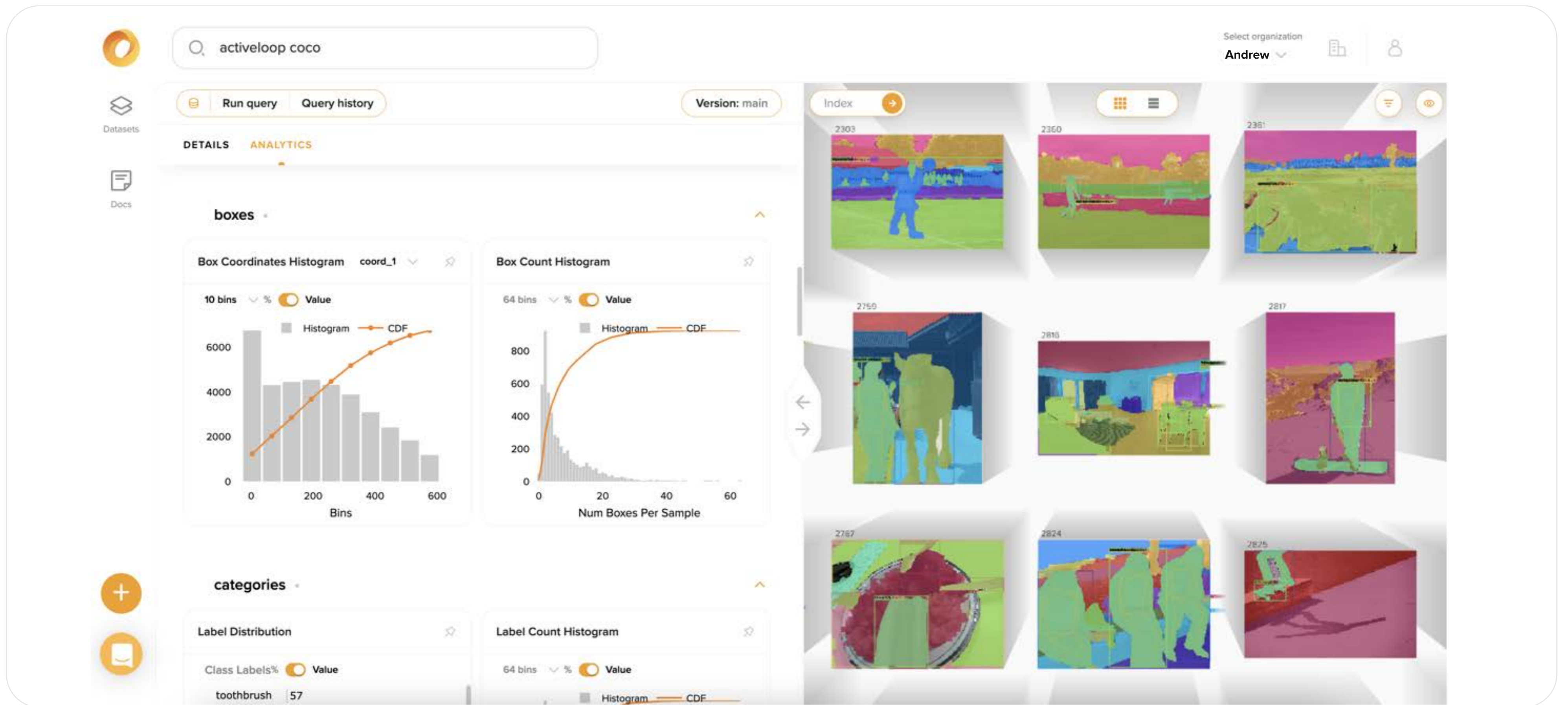
Version Control: Git for data

Typical data lakes offer time travel, which is a linear timeline of changes made to the dataset over time. However, time traveling does not solve for multiple versions of the same asset, such as multiple annotations obtained from different annotators.

Deep Lake offers git-like dataset version control. The unstructured data are cryptographically hashed before becoming part of the dataset. Then, any modification to the dataset is committed. Users are able to checkout to new branches and merge branches while resolving conflicts.

Dataset lineage for adding images, annotating, then running a query to stream to train a model





Viewing the dataset qualitatively as layered tensors (right) and quantitatively as distributions (left)

2 Visualize: In-browser visualization engine

Deep Lake allows the user to semantically visualize, seamlessly explore, and visually interact with datasets. Visualization includes both macroscopic (e.g., on a grid, clustered by embeddings, on a GIS map, graph-based, etc.) as well as microscopic multi-modal views of the data.

Exploration involves slicing, querying, filtering, combining, and examining distributions in an interactive user experience. Interaction involves in-place modification (optionally over an API) or seamless integration with other tools and API services.

3

Query: Rapid queries with Tensor Query language (TQL)

SQL operations and expressions, including arithmetic, logical composition, and array indexing work well on tabular data, but they have limited applications for multidimensional arrays. Data engineers often operate on multidimensional arrays and tensors, and supporting expressions for ML data operations is essential for advanced queries in Deep Lake.

Let's say you want to construct a dataset of 1,000 images and labels while the weather is "raining" and there are "bicycles" in front of the camera. Furthermore, you already have a prediction of an existing model and you want to order by prediction error.

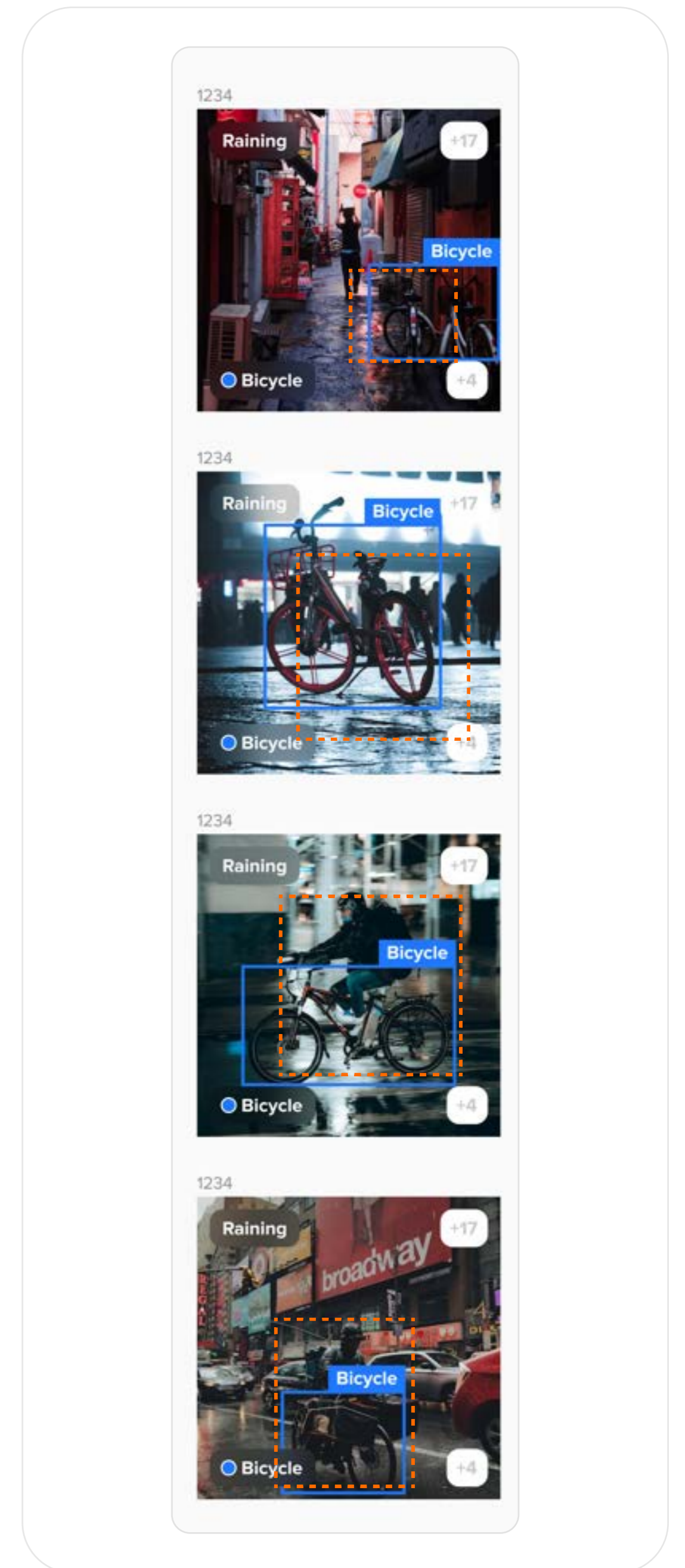
Finally, your model only takes 400px width/height input, so you need to crop the images and adjust bounding box arrays. The following query will construct the dataset and make it ready to be trained on the deep learning framework of your choice.

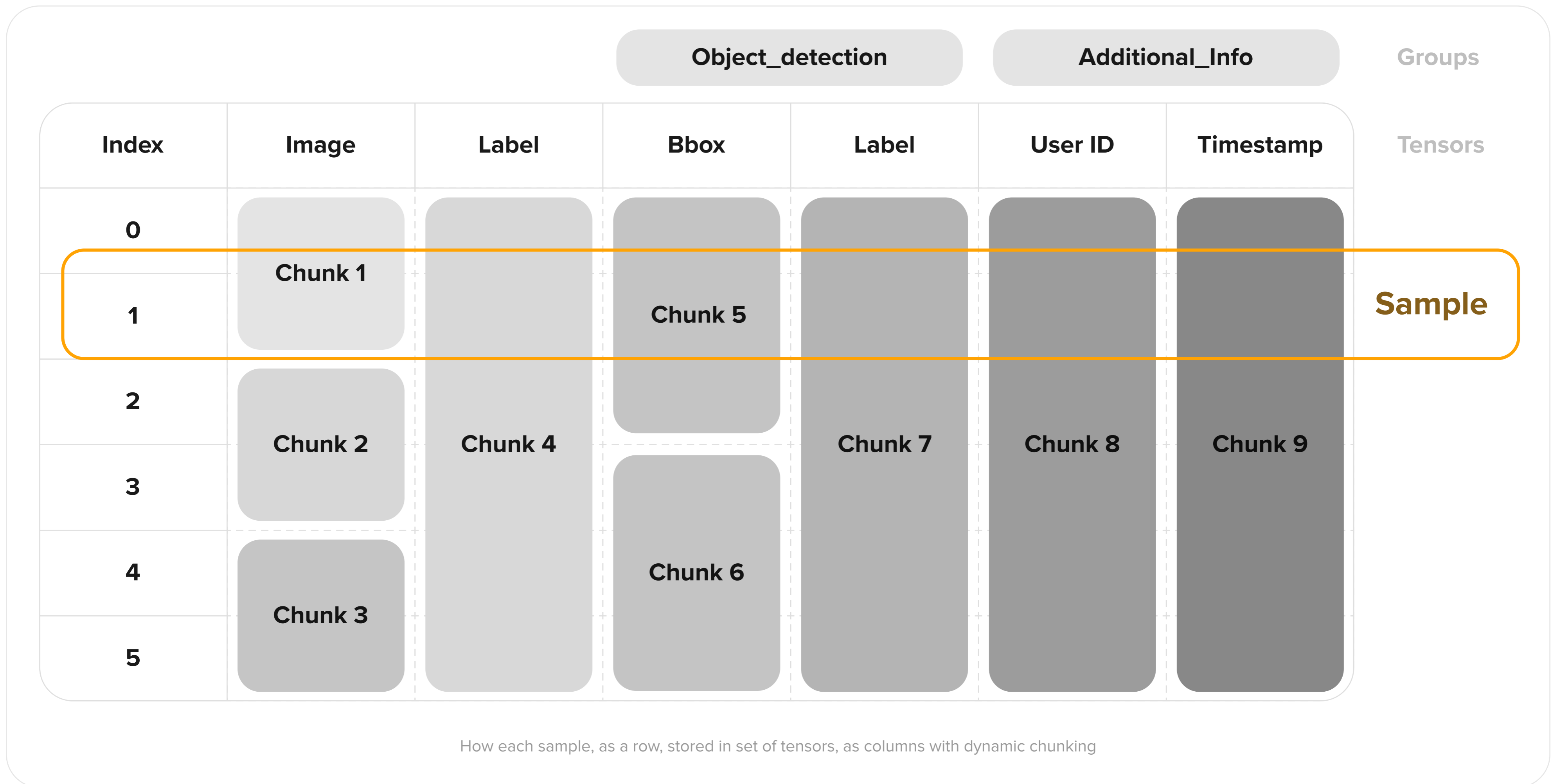
It would take a data scientist several hours and hundreds of lines of code to construct the dataset and start fine-tuning the model. However, with Deep Lake, they can do it with just one SQL command.

Advanced queries come with built-in NumPy-like array manipulations, ready-to-use ML evaluation functions, and inference models.

```
SELECT images [100:500, 100:500], boxes + ARRAY[-100, -100, 0, 0]
WHERE contains(categories, 'bicycle') and weather == 'raining'
ORDER BY AOI(boxes, prediction) desc
LIMIT 1000
```

Example query with indexing tensors inline with select and ordered by user-defined function computation.

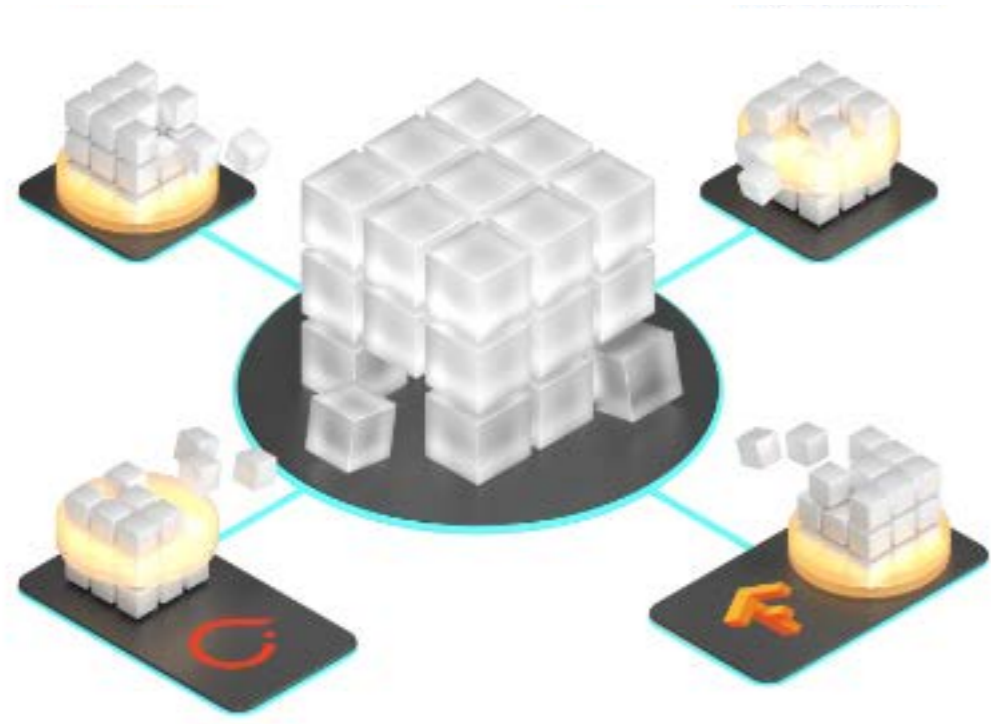




4 Materialize: Format native to deep learning

Once a data scientist has finalized a dataset view to start training a model, handling large numbers of files and copying the dataset to the computing machine is cumbersome. Often, GPUs become underutilized while the data are being copied from storage to the local machine.

Materialization of the dataset transforms the virtual view into deep learning ready tensors and enables streaming of the data from Deep Lake directly into the GPU very efficiently.



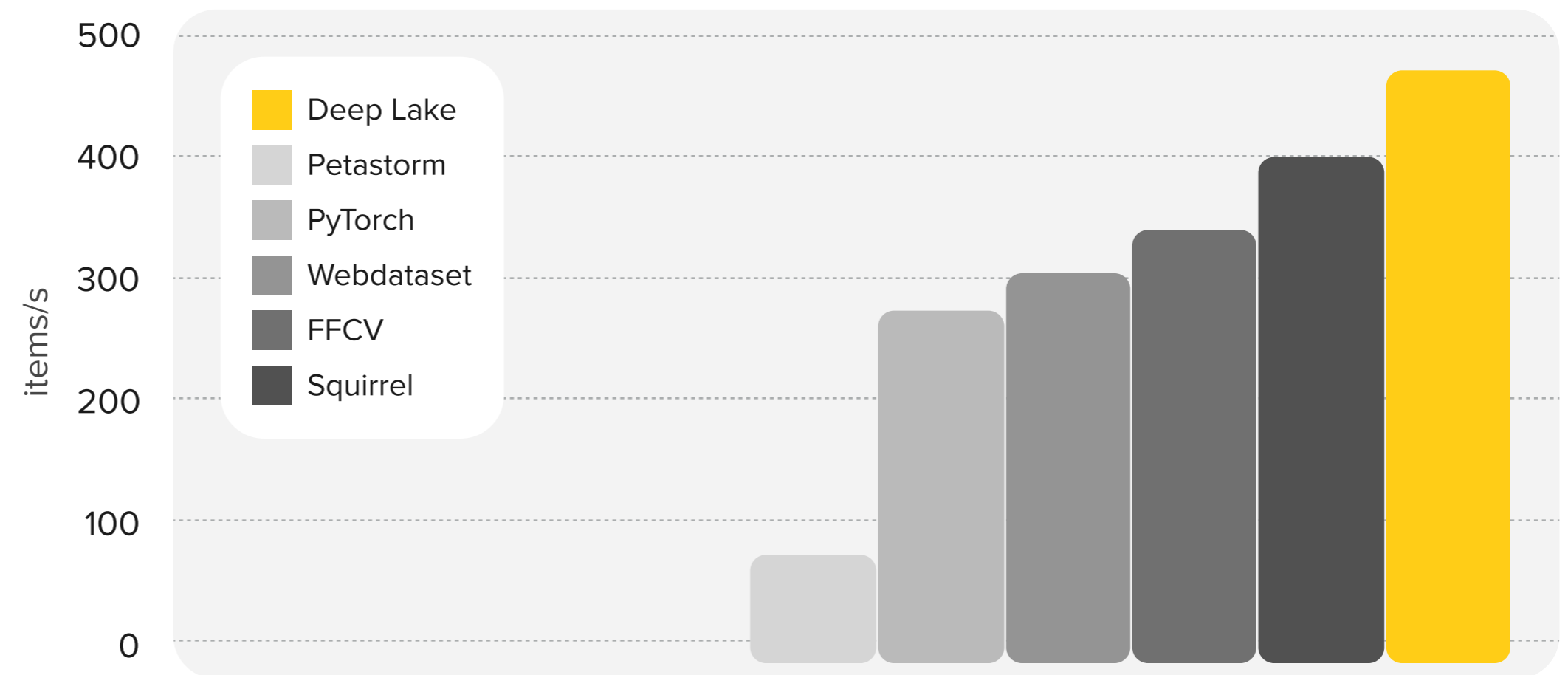
5 Stream: Streaming Data Loaders

With the recent development of deep learning frameworks, running computations on accelerated hardware such as Nvidia GPUs or Intel CPUs has become highly optimized. Rather than the computation itself, the bottleneck is now in the hand-off of the data to the models. Python presents severe limitations in multiprocessing/multithreading, causing issues with concurrent and asynchronous data transfer to the compute device.

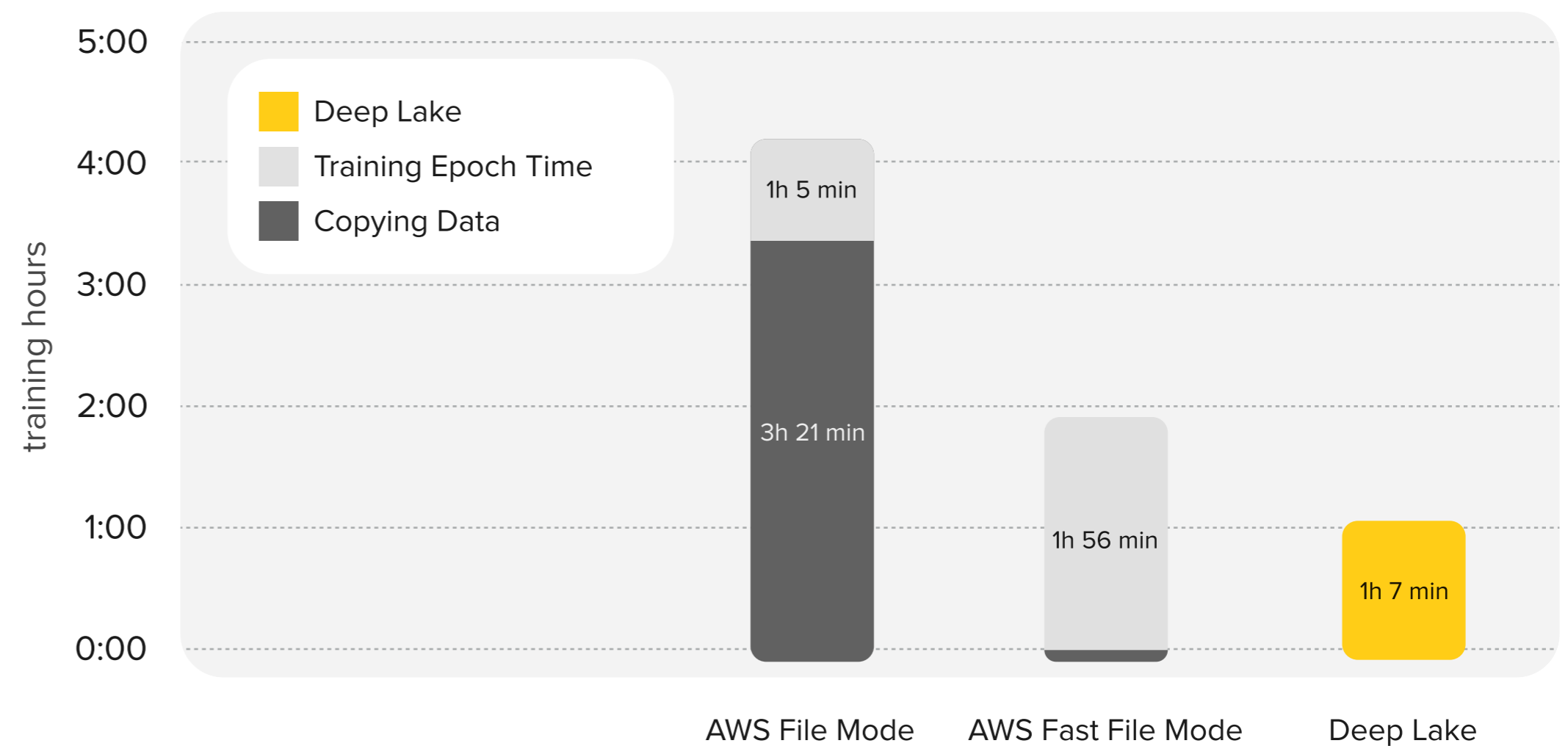
Deep Lake comes with native integration with deep learning frameworks such as PyTorch, Tensorflow, or JAX. The data loader efficiently streams the data from remote storage to the GPUs while models are being trained.



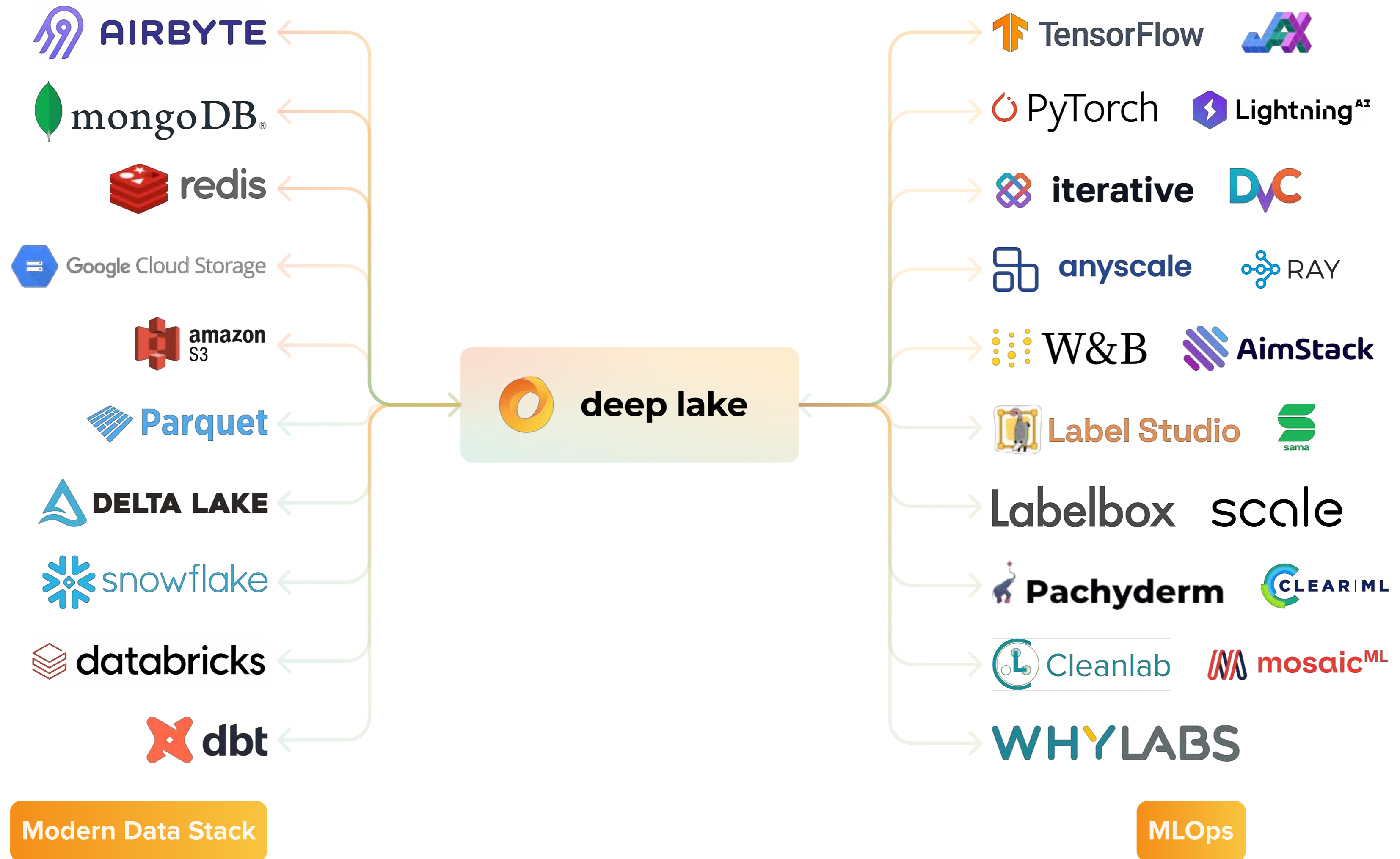
Iteration speed of images against other dataloaders
(higher better)



Training Imagenet on AWS
(lower better)



Integrating Modern Data Stack and MLOps



How Deep Lake is Revolutionizing Deep Learning

Data lakes are becoming increasingly popular for deep learning data management. Still, they are not optimized for the Volume and Variety of unstructured data common in deep-learning applications.

Deep Lake is a new data infrastructure tool that superpowers deep learning practitioners in managing their data. With its ease of use, features optimized for deep learning workflows, accessibility, and ability to connect any deep learning data source, Deep Lake is an organization's infrastructure of choice when building a deep learning storage framework.

Deep Lakes are increasing in popularity because they can handle various data types and can be scaled to large amounts of data. By structuring data in tensors native to the data structures used by deep-learning models, Deep Lakes achieve high performance when connecting data to training and production use cases.

The key benefits:

Driving revenue growth by enabling teams to ship AI products faster

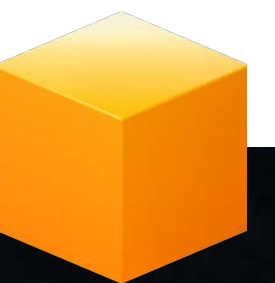
Saving money by reducing GPU compute cost

Increasing data scientist's focus on core business problems, and

Eliminating failed ML project risk because of no solid data foundation



Deep Lake runs on intel®



Alternatives to Deep Lake























Today, your unstructured data typically sits on an object storage such as S3 or a distributed file system. Every time a virtual machine starts, the data is copied to run training or inference. This results in compute inefficiencies including idle GPU utilization. Data scientists write their own custom pipelines to construct datasets which are not reusable.

Another option is to extend standard approaches such as Parquet or Arrow formats to contain complex data. This comes with benefits such as leveraging existing analytical tools including

Spark or Kafka. But those tools are specialized for tabular, time-series, and event data processing and will break with complex, unstructured data.

You might also consider building it in-house, but it will take years, and a dozen experienced data engineers to get the perfect data engine vertically aligned with your product core.

How to Get Started with Actiueloop for your Deep Lake

	Data Lake	 Deep Lake Community	 Deep Lake Enterprise
Data Layout	 (based on Parquet, Avro, ORC)	 (unstructured data)	
Version Control	 (time travel, single history)		 (verifiable)
ACID Collaboration			 (scalable)
Query Engine	 (integrated)	 (Python-native)	 (Tensor Query Language)
Streaming Dataloader			 (performant)
Integrations with MLOps			 (labeling, experiment tracking)
Visualization			
Role Based Access Control			
Audit Trail and Logging			



Deep Lake

Data Lake For Deep Learning

Conclusion

Deep Lake provides a cost-effective storage infrastructure for deep learning datasets and models. Using Deep Lake, an organization avoids the cost and complexity of deploying and maintaining tools for managing its datasets. Like data lakes, Deep Lake also provides flexibility in storing and accessing data, allowing organizations to find an optimal storage architecture for their data type and structure.

Data lakes can be a valuable addition to a deep learning workflow, providing cost-effective storage and flexible access to data. However, when selecting a data lake solution, it is essential to consider the needs of your deep learning workloads. For example, some data lakes may not offer the performance or scalability required for training large deep learning models. Others may not provide essential features for deep learning,

Deep Lake is explicitly built for deep learning data and ensures the success of your deep learning workflows. This saves time on building complex data infrastructure, increases the Machine Learning team's focus on core business problems, reduces ML project risk, and enables shipping AI models into production much faster.