Data and Agent Aware Inference & Training Network (DA-ITN)

A unified, intelligent, multi-plane architecture for AI learning, inference, and agent interactions

Hesham Moussa
hesham.moussa@huawei.com
Arashmid Akhavain
arashmid.akhavain@huawei.com

Venue: IETF 124 - NMRG
Date: November 2025

Contents

Related publications and drafts

Submitted draft v_00:
https://www.ietf.org/archive/id/draft-akhavain-moussa-ai-network-00.txt

Draft Version v_01:
https://github.com/DA-ITN/draft-akhavain-moussa-ai-network

IEEE network magazine:
Distributed Learning and Inference System: A Network Perspective

Background

AI expanding footprint:

- Driven by evolving model architectures: ChatGPT, Claude, Grok, DeepSeek
- Supported by dataset generation and consumption frameworks: Huggingface, Tensorflow
- Applications: Range from simple classification tasks, to applications that require super intelligence and reasoning capabilities. More recently, agentic applications have emerged transforming daily life activities.
- Success of AI requires= robust Training + Inference frameworks.

Training:

- Large amount of naturally distributed datasets, a variety of AI tasks and objectives, massive compute and storage resources, high-speed interconnects.
- Centralised vs. distributed

Inference:

- Low-latency, continuous operation, compute and storage resources, context window, high-speed interconnect, tool handling.
- Single-model vs. multi-model collaboration.
- Collaboration level

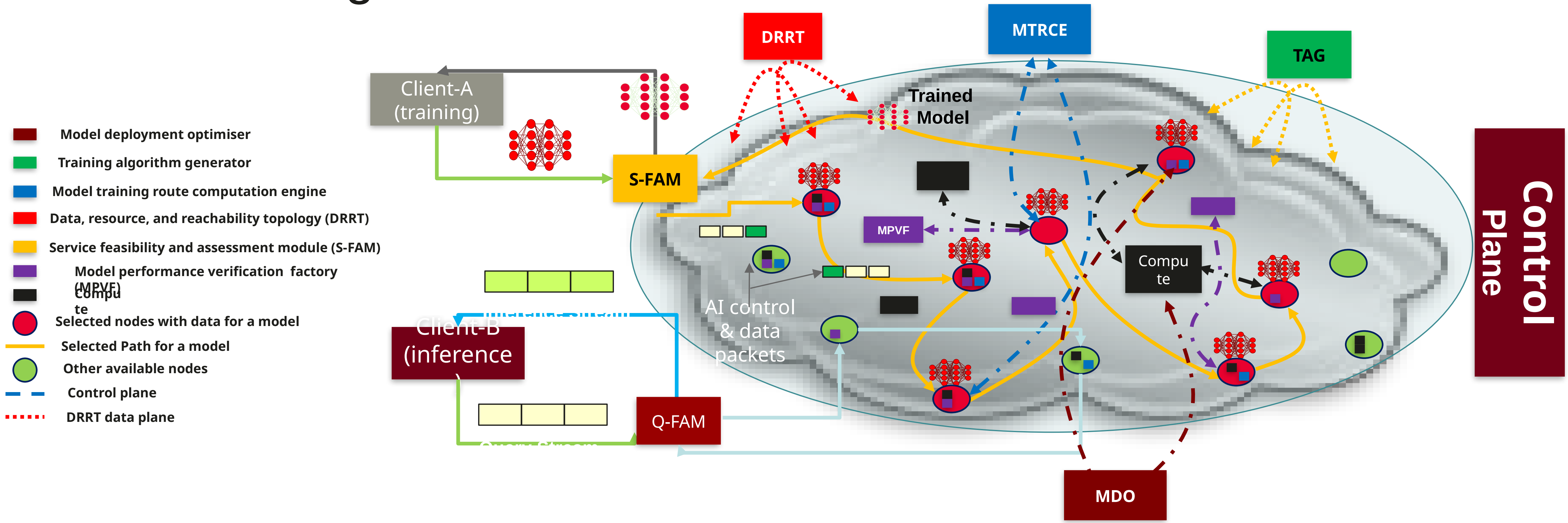
Problem Statement and Objective

- **Problem Statement:**
  - Traditional networks are not designed to meet the unique demands of AI systems including training, inference, and agentic interaction.
  - While they may offer sufficient bandwidth to transport datasets, models, queries, and agent messages, they lack the required architectural constructs and intelligence needed to support AI model training, efficient inference, and collaborative agent services.

- **Solution: Data & Agent Aware-Inference and Training Network (DA-ITN)**
  - A newly proposed next-gen application-layer/overlay network tailored for machine intelligence
  - It comprises core components for operations and intelligence, a dedicated data plane for transport, a control plane for coordination, an OAM plane for maintenance and exposure, and APIs that enable dynamic operations and integration.

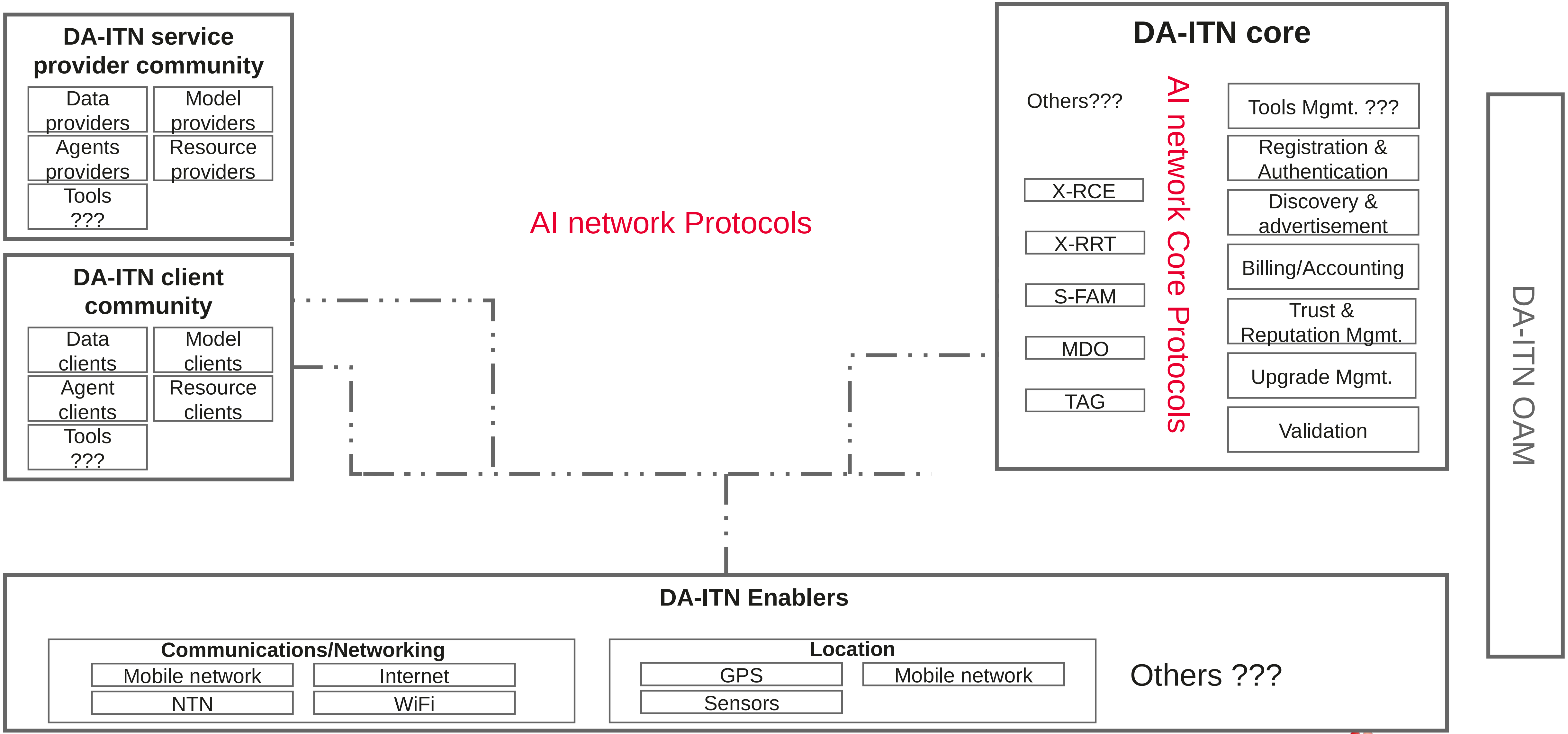
- **Goal and Objective:**
  - Build a unified, intelligent, multi-plane network that supports the full spectrum of AI services and fosters an ecosystem delivering value to providers, enablers, and clients.

DA-ITN: High Level View

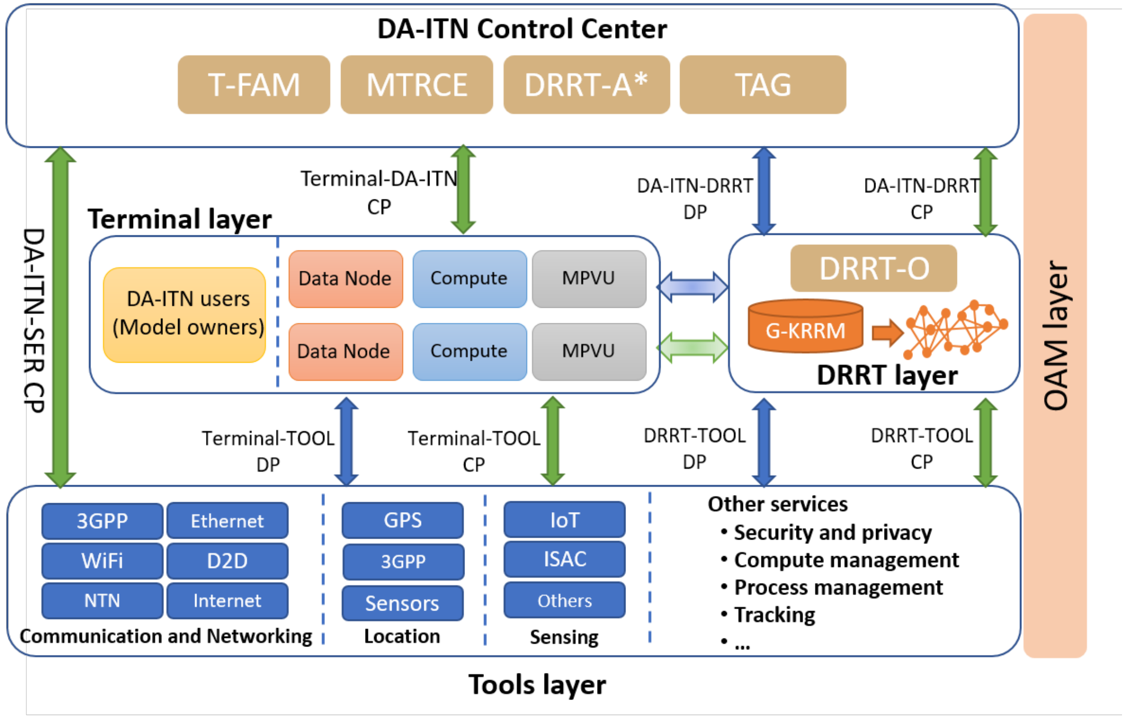


- DA-ITN delivers a unified control, data and OAM planes to support different AI related tasks.
- Training, inference, and agent-to-agent collaboration/interaction can potentially have specific sub-planes within DA-ITN control, data, and OAM planes.
- Supports centralized and distributed architectures.
- Allows support for different training methods (e.g. FL, SL, CL, etc.)

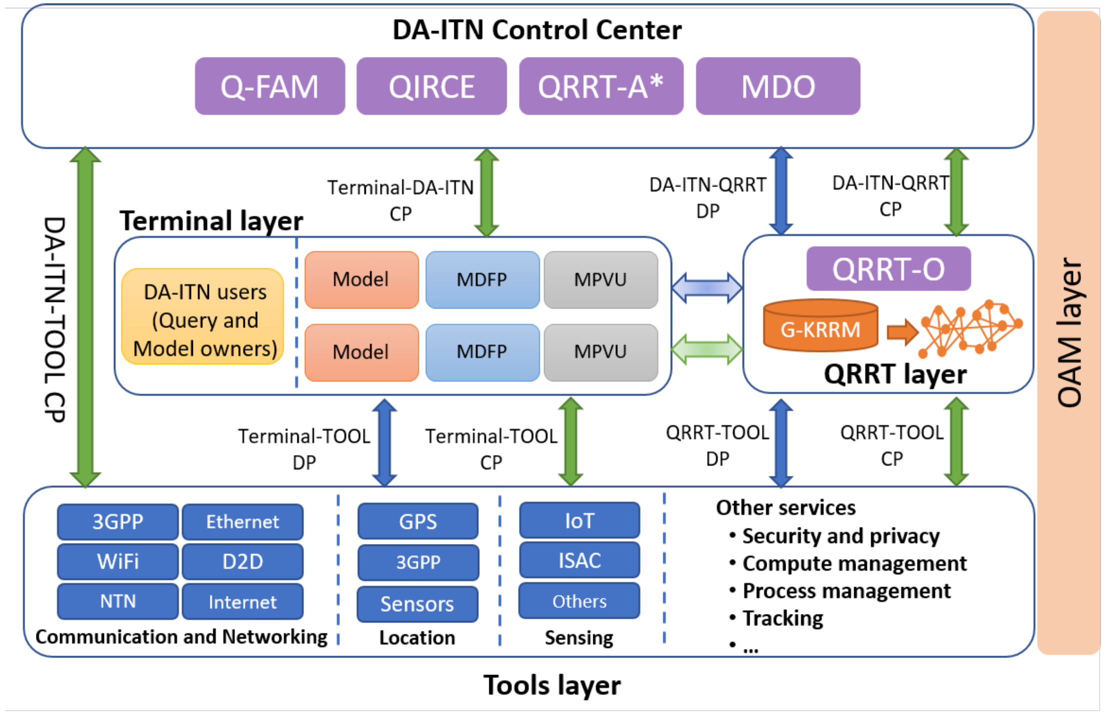
DA-ITN architecture and building blocks



DA-ITN Sample Frameworks



Training Framework



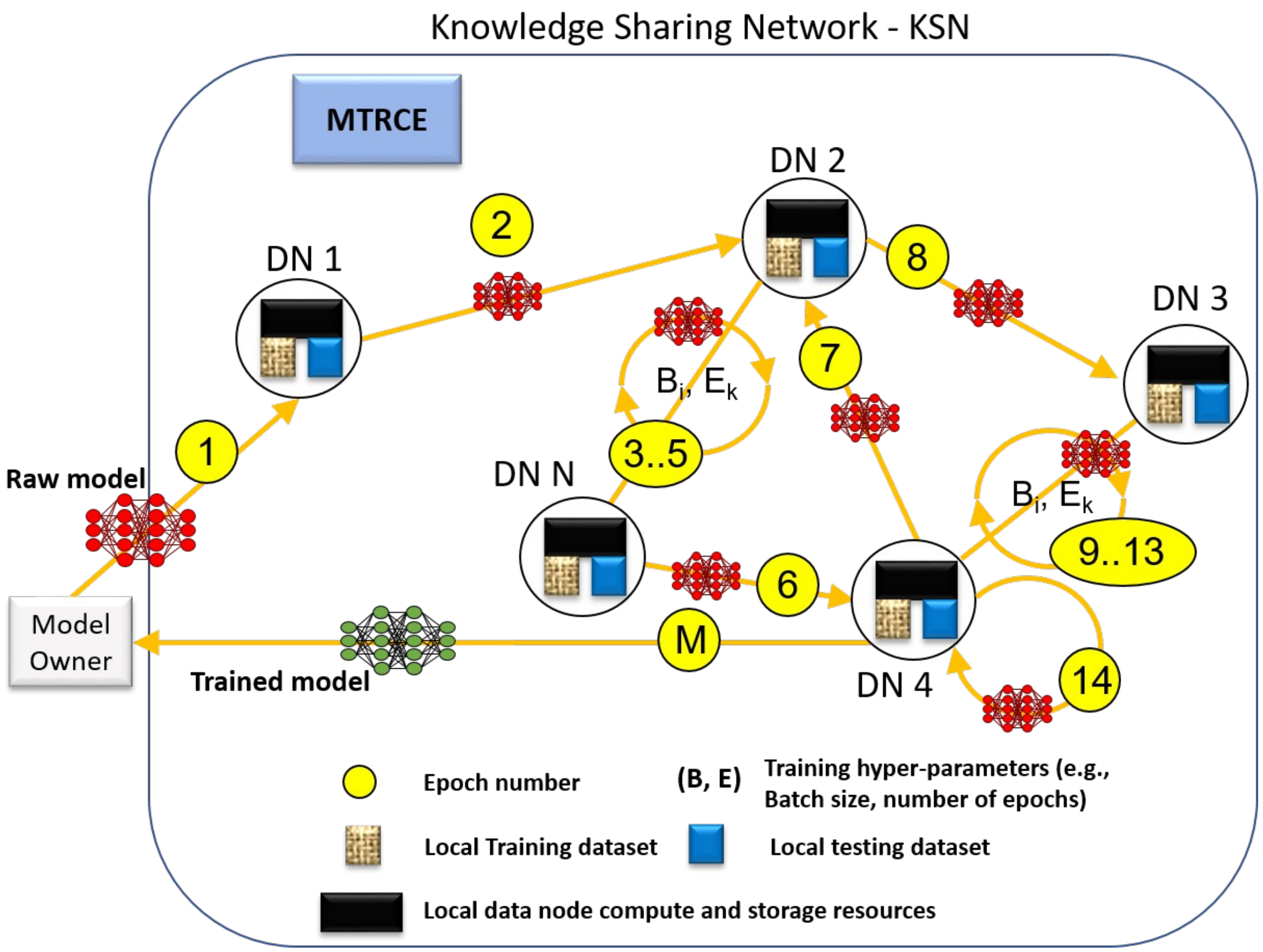
Inference Framework

Current Research: Distribute AI Model Training

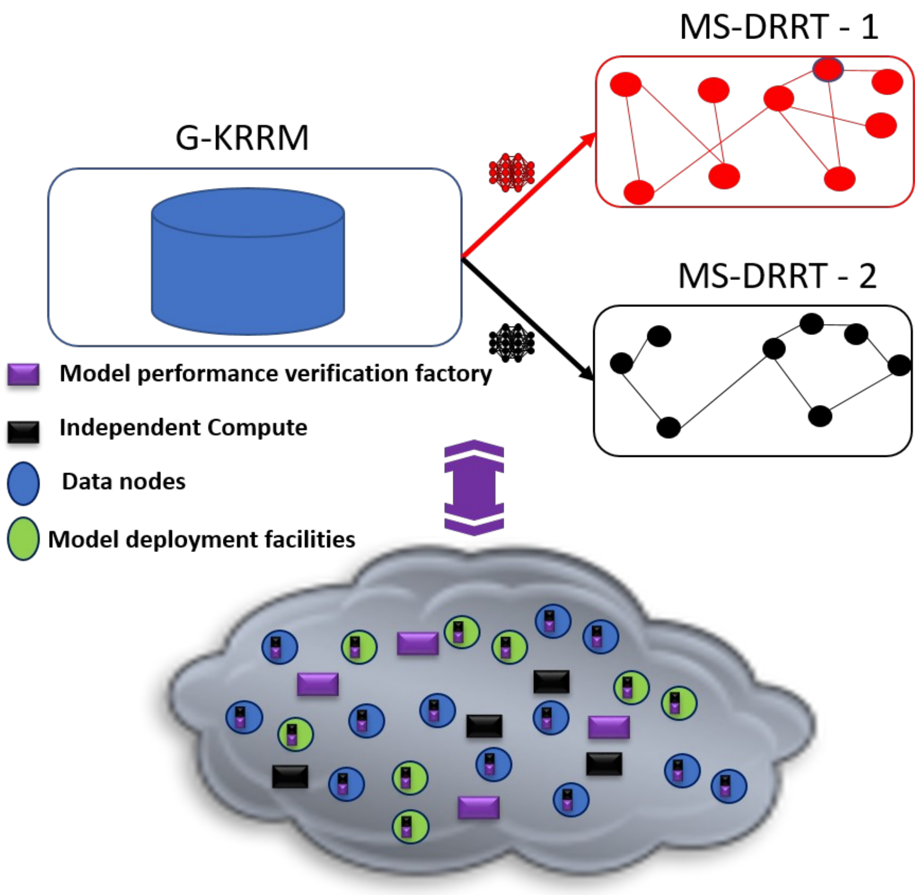
- Consider a large data corpus distributed across various equipment.
- A client with an AI Model that requires training submits the model to the DA-ITN along with a set of objectives and performance KPIs.
- The objective of the DA-ITN is to train the submitted model in the most efficient way while meeting clients objectives and KPIs
- Step 1: efficiency
  - It can be argued that there is an optimal subset of data that is sufficient for training each specific model (i.e., A model can avoid training on two nodes with similar knowledge content)
  - Therefore, there is a need to carve out this optimal subset from the global canvas for the purpose of efficient model training w.r.t communications, compute, storage, etc.
- Step 2: Optimal training strategy
  - After determining the optimal subset of training data, the model needs to be optimally trained on the data. This involves choosing the training paradigm (centralized vs decentralized) and choosing the optimal training hyper parameters
  - Optimal decisions need to account for data related aspects, compute, communication efficiency, privacy, and client objectives and KPIs

Model Training Route Compute Engine (MTRCE)

- Computes a specific training plan for optimal training performance given current dataset status, model architecture, training objectives and KPI, communication efficiency…etc.
- Determines the training hyperparameter (Batch size, number of epochs, etc.) for each node visit.
- Manages visits to Model Verification Performance Factory (**MVFP**)
- Depending on data fluidity and rate of change, **MTRCE** can either compute the entire training plan or simply slowly evolve the plan as the model progresses with training and as data demographics gets updated.



Data, Resource, and Reachability Topology



DRRTs are envisioned as dynamic, map-like structures that:

- ❑ Are carved out from a globally collected unstructured Knowledge, resource, and reachability maps based on the incoming model and model requirements
- ❑ Hold information related to data such as Data type, quality, volume, age, and dynamics
- ❑ Monitor resource availability and reachability status of the nodes in the **KSN**
- ❑ Used by the **DA-ITN** to make accurate model steering decisions to fulfill model training requirements. (They could also be used to enable other services including inference (RAG))

HUAWEI

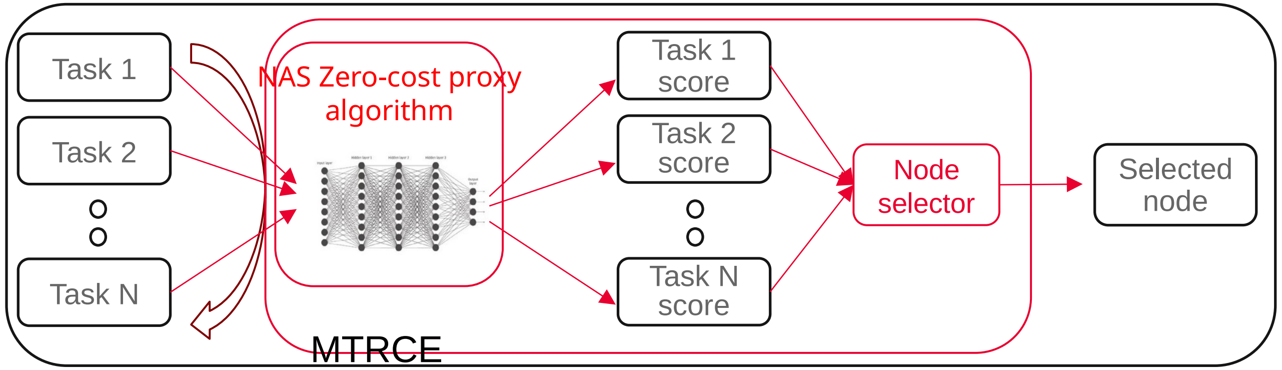
Continual Learning as an application

- Continual learning a training paradigm that involves training an AI model sequentially on tasks/datasets
- An AI model trained on a task and then moves to the next task and trains on it and so on
- A well-know challenge in continual learning is catastrophic forgetting (CF), where as the model moves from one task to the other, its performance on previously learned tasks degrades
- Many solutions for CF have been proposed in the literature and can be categorized into five main categories: Replay-based methods, Regularization-based methods, Optimization-based methods, Representation-based methods, and Architecture-based methods.
- Recently, it was shown that choosing the right order by which tasks are presented to the model can alleviate CF, especially if coupled with the typical CF solutions above

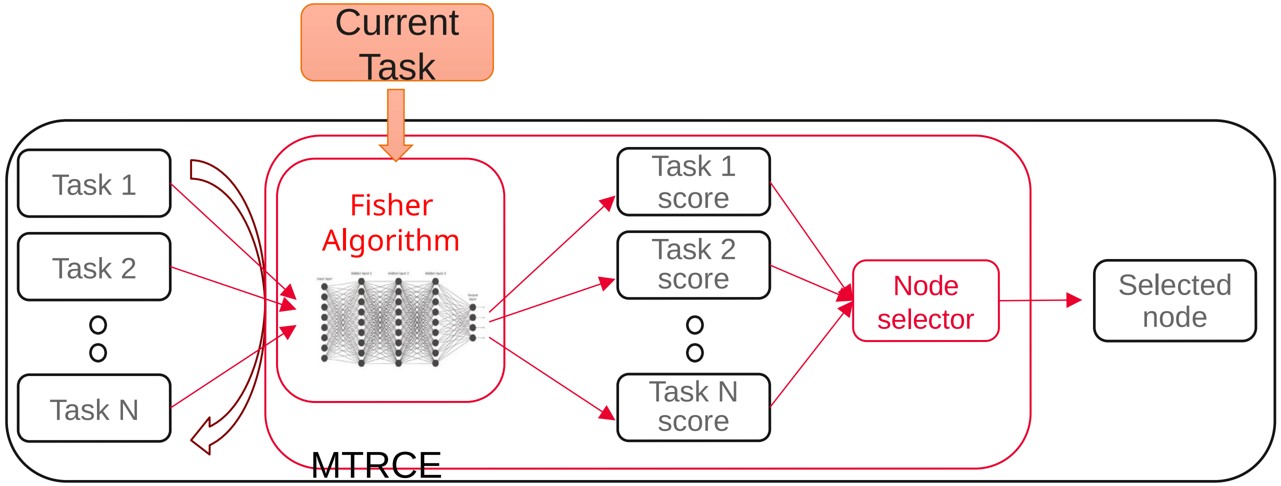
In this work, we devise two intelligent and light weight methods to choose the right order by which tasks are presented to the model and show that CF is alleviated

NWOT and Fisher MTRCE Algorithms

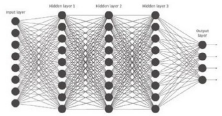
- In both methods, a minibatch of data is collected from a dataset and used to compute a score or a distance
  - In the NAS inspired solution, the score does not compare datasets to each other
  - In the Fisher based solution, the distance compares datasets to the current dataset that the model is being trained on



NAS inspired

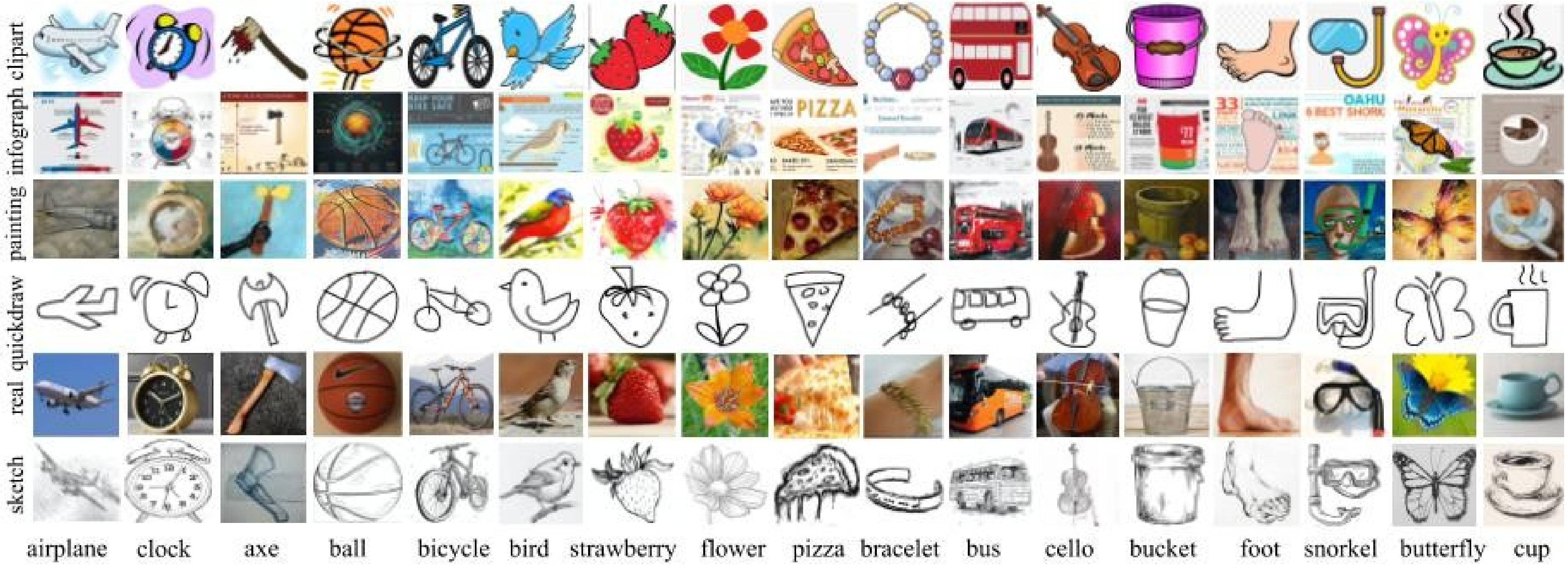


Fisher inspired



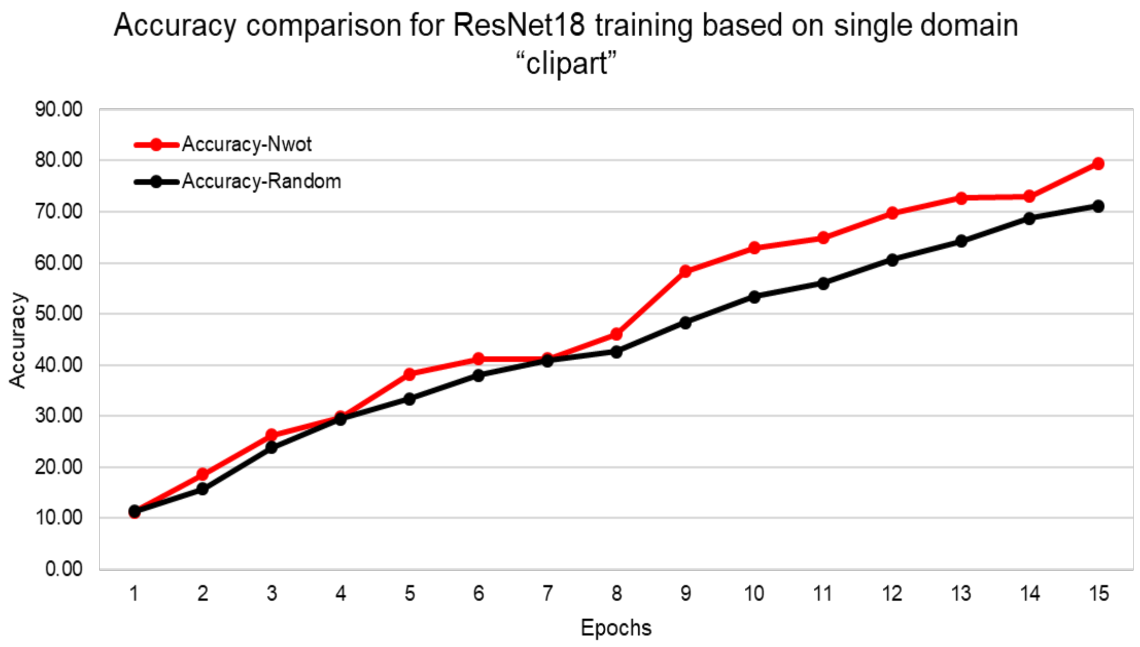
Model that requires the training

Dataset for experimentation

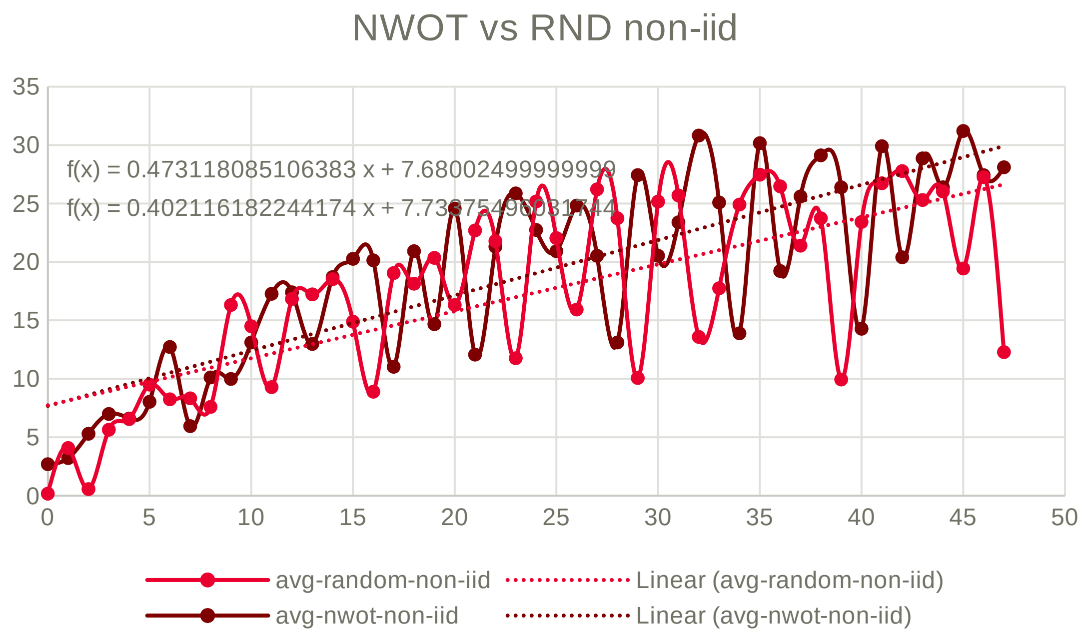


DomainNet Dataset

Sample results: NWOT

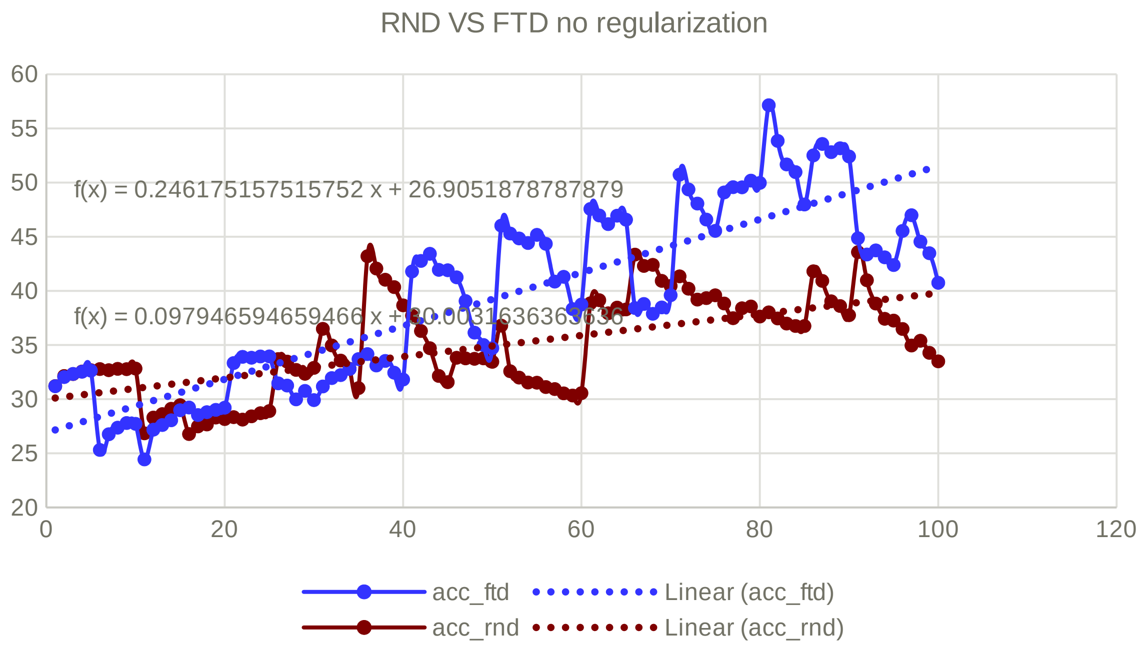


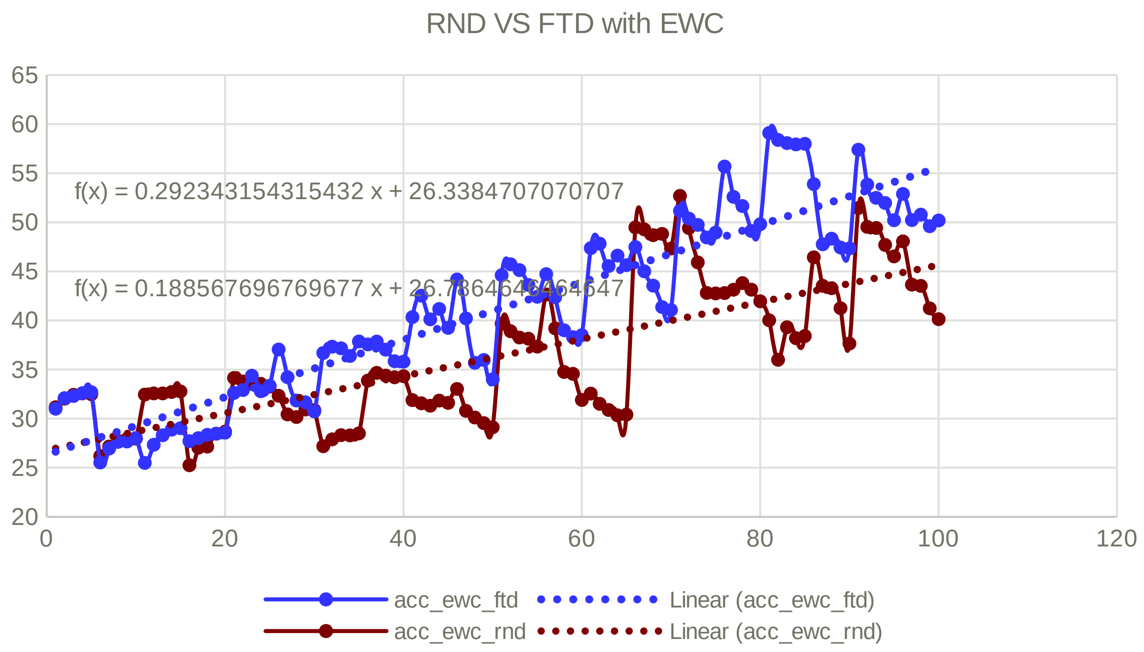
Full-IID experiment



Full-Non-IID experiment

Sample results: Fisher (semi-IID: inconsistent class distribution)





HUAWEI

Some Research Questions: MTRCE

- What alternative algorithms can be integrated into the MTRCE?
- Is a distinct MTRCE algorithm required for each learning paradigm (e.g., federated, knowledge distillation, centralized), task (e.g., NLP, prediction, speech recognition), or model type (e.g., Transformers, CNNs, RNNs)?
- How can training objectives and requirements be embedded into MTRCE operations?
- How should the MTRCE adapt to dynamic, continuously changing data?
- How can it handle diverse dataset distributions, from IID to highly non-IID?
- Can data privacy be preserved in MTRCE decision-making (e.g., without collecting minibatches)?
- How can performance be tested in a distributed manner without exposing private data?
- How can we handling scalability matters?
- How can the MTRCE be optimized for compute, communication, and storage efficiency?
- Should MTRCE inputs be standardized, or should algorithms freely request inputs with the DA-ITN control plane responding accordingly?
- What is the MTRCE equivalent for inference and agentic contexts?

Some Research Questions: DRRT

- How to measure and establish trust as new datasets/data nodes join the DA-ITN?
- How to describe dataset with a certain objective in mind?
- How can a dataset be described with a specific objective in mind? (e.g., how to describe a dataset to a model that wants to train on it?)
- Is there a universal dataset description that can be used to facilitate any AI model training services regardless of the training paradigm, training task, or AI model architecture? If so, what would such a descriptor look like?
- What requirements must descriptors meet (e.g., size, privacy, freshness, labeling, on-demand vs. pre-defined generation, cost)?
- Does the concept of a dataset topology even exist? And if it does, can we use dataset descriptors to generate them?
- What are some expected features of such topologies? Are DRRT topologies on-demand, or are they persistent structures? Are they created per AI model? What do the links between different points in these topologies signify? What is the span of each topology? Will we have a global DRRT or regional ones? Will we run into scalability problems? Can we use concepts such as abstraction and hierarchy?
- Can we carve out smaller DRRTs from a global one based on certain objectives and requirements? If so, how to do that? How to relate and utilize the carved out DRRT to a certain AI model training task (i.e., How can DRRTs integrated into DA-ITN?)
- Is there an equivalent construct for inference and agentic operations, and if so, what must those descriptors and topologies convey?

Thank you.

Bring digital to every person, home and organization for a fully connected, intelligent world.

HUAWEI