Multi-tasking, Meta-learning and Federated Learning

From Improved Performance to Privacy Preservation

Anshul Thakur

Department of Engineering Science
University of Oxford

7 Nov, 2025

Table of Contents

Table of Contents

What are Transforms?

- Transform data from one domain to other domain
  - E.g. Time domain to frequency domain
- Why transform data?
  - Analysis of data may be easier in transformed domain
  - Certain operations may only be feasible in transformed domain
  - Privacy Preservation! (potentially)
- Linear Transformation functions are invertible
- Deep neural networks are non-linear non-invertible transforms!!!

Transforms around us!

- **Discrete Cosine Transform:** Represents a finite signal as a combination of cosine functions of different frequencies
- Allows compression of signals. Used in:
  - digital images (JPEG)
  - digital video (MPEG)
  - digital audio (MP3, Dolby Digital)
  - speech coding
- Made streaming music, games and Netflix possible

Combination of Vectors

- Let $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \ \ldots \mathbf{v}_n\}$ be a set of N vectors and $\{\alpha_1, \alpha_2, \alpha_3 \ \ldots \alpha_n\}$ be a set of N scalars
- **Linear combination:** $\mathbf{y} = \sum_{i=1}^{N} \alpha_i \mathbf{v}_i$
- **Affine combination:** $\mathbf{y} = \sum_{i=1}^{N} \alpha_i \mathbf{v}_i$ where $\sum_{i=1}^{N} \alpha_i = 1$
- **Conic combination:** $\mathbf{y} = \sum_{i=1}^{N} \alpha_i \mathbf{v}_i$ where $\alpha_i \geq 0$
- **Convex combination:** $\mathbf{y} = \sum_{i=1}^{N} \alpha_i \mathbf{v}_i$ where $\sum_{i=1}^{N} \alpha_i = 1$ and $\alpha_i \geq 0$

Transform your data

- Let $\mathbf{x} \in \mathbb{R}^d$ be an input vector and $\mathbf{W} \in \mathbb{R}^{d \times m}$ be a transformation matrix, a simple linear transform is:

$$\mathbf{y} = \mathbf{W}^T \mathbf{x} \tag{1}$$

- **Random Projections:** Project data to low dimensional space using Random Gaussian matrix

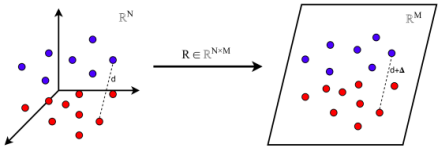


Figure: Random projections for dimensional reduction.

Types of Transforms

- Transforms can be categorised based on the nature of transformation matrix
  - **Analytical transform: W** is analytically determined
  - **Learned transform: W** is learned from the data itself
- Analytical transforms:
  - Data independent and universal
  - E.g. Fourier and cosine transforms where signal is decomposed onto sinusoidal waves of different frequencies
- Learned transforms:
  - Data dependent
  - **W** contains basis of a subspace or exemplars
  - E.g. PCA as principle components are learned from the data
  - **Dictionary learning:** Learn transformation matrices or dictionaries from data and decompose data onto the atoms of these dictionaries

Types of Transforms: Deep Learning

- Deep neural networks can be seen as cascades of learned transforms:

$$\mathbf{y} = \mathbf{W}_3\mathbf{W}_2\mathbf{W}_1\mathbf{x} \tag{2}$$

But this is linear! (Deep dictionary learning[1])

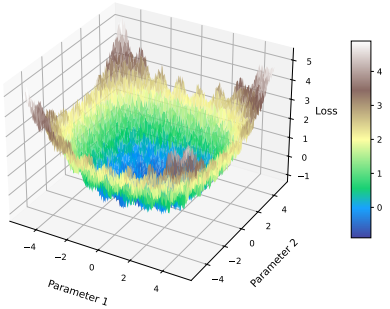
$$\mathbf{y} = \text{ReLU}(\mathbf{W}_3\text{ReLU}(\mathbf{W}_2\text{ReLU}(\mathbf{W}_1\mathbf{x}))) \tag{3}$$

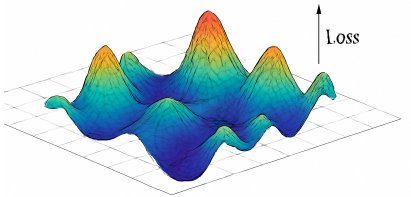
- DNNs learn these transforms simultaneously using gradient descent
- Purpose is to learn embedding or $\mathbf{y}$-space where classes are separable

[1] Thakur & Rajan, Deep archetypal analysis based intermediate matching kernel for bioacoustic classification, IEEE JSTSP 2019.

Loss Landscape

- A loss landscape represents how the loss (or error) of a neural network changes as we adjust its weights.





Loss Landscape of Neural Networks

Loss Landscape

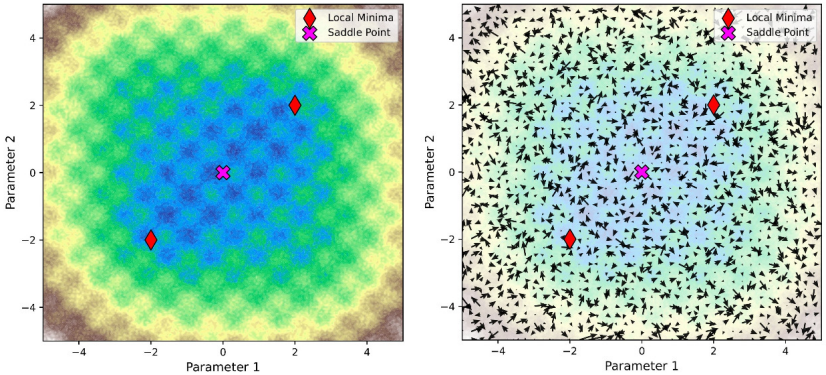
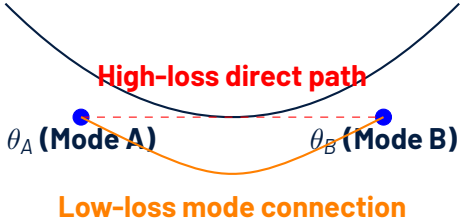


Figure: Relation between loss landscape and gradient field.

What is a Mode Connection?

- A **Mode Connection** is a **low-loss path** between two local optima (or **modes**) of a neural network.
- This path satisfies:
  - **Smooth Transition:** Every point along the path remains a viable local minimum.
  - **Loss Constraint:** The entire path maintains loss values **no worse** than the endpoints.



What is a Mode Connection?

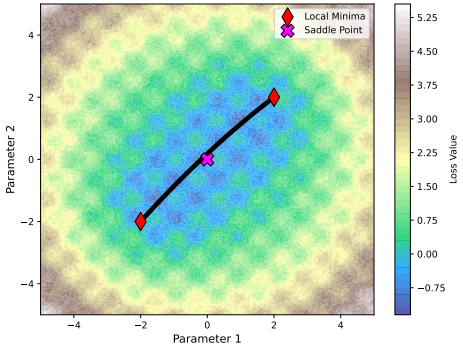


Figure: Mode connection in loss landscape.

Why is Mode Connectivity Important?

- **Understanding the Loss Landscape:**
  - Challenges the assumption that local minima are isolated.
  - Reveals the underlying structure of neural network loss surfaces.
- **Finding Flatter Minima for Better Generalization:**
  - Improves **robustness** by making models less sensitive to **small input changes**.
  - Enhances stability by reducing sensitivity to **weight perturbations**.
  - Leads to **smoother decision boundaries**, improving test-time performance.
- **Providing Insights for Optimization:**
  - Guides new training strategies for selecting better minima.
  - Helps understand interactions in **joint optimization and model merging**.

Linear Mode Connectivity

- **Interpolation** between two optima using a direct linear path in parameter space.

$$\theta(t) = (1 - t)\theta_A + t\theta_B, \quad t \in [0, 1]$$

- Often passes through **loss barriers**
- Observed when loss landscape is **approximately convex** and models are trained similarly

Non-Linear Mode Connectivity

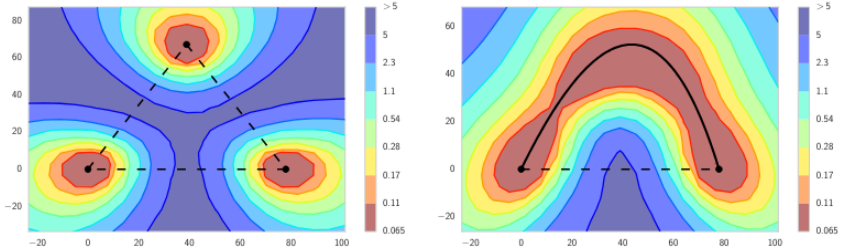


Figure: Loss Landscape of Resnet-164 trained on CIFAR-100.

Garipov et al., *Loss surfaces, mode connectivity, and fast ensembling of DNNs*, NeurIPS (2018).

Parametric Bézier Curves in Mode Connectivity

- **Bezier Curves:** Trainable Parametric curves that can act as mode connections

$$\theta_\phi(t) = (1-t)^2\theta_A + 2(1-t)t\phi + t^2\theta_B, \quad t \in [0,1]$$

  - ϕ is the learnable **control point**, optimized to minimize loss along the curve.
- **Training the Bézier Curve**
  - Instead of direct weight interpolation, we **optimize** ϕ to ensure a **low-loss path**.
  - Training objective:

$$\mathcal{L}_{\text{Bézier}} = \mathbb{E}_t\Big[\mathcal{L}(\theta_\phi(t))\Big]$$

  - We **sample values of** t, compute loss at those points, and optimize ϕ to minimize the loss along the curve.

Non-Linear Mode Connectivity in Action

- Respiratory deterioration prediction model

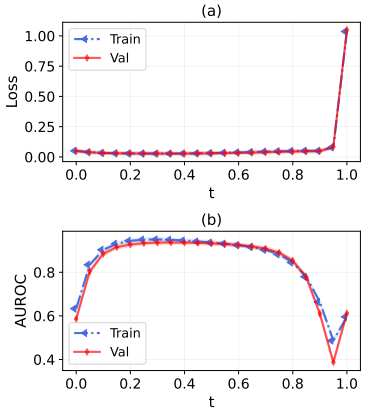


Figure: Mode connection between an optimum and a random point in parameter space.

Table of Contents

Multi-tasking Architectures

- Multi-tasking architectures are characterised by:
  - A shared feature extractor "trunk"
  - Tasks-specific layers working on same feature representation

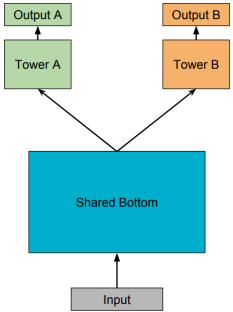


Figure: Illustration of a typical multi-tasking architecture.

Multi-tasking: Under the hood

- $\mathbf{x}$: input examples. $(y_1, y_2, \ldots, y_n)$: corresponding task labels
- Model $f_{\theta,\phi_i}()$ outputs task-specific predictions: $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n$
- θ: shared parameters, ϕ_i: task-specific parameters for ith task
- Loss function is an aggregate of task-specific losses:

$$\mathcal{L} = \sum_{i=1}^{n} \alpha_i \mathcal{L}_i(y_i, \mathbf{p}_i), \tag{4}$$

where α_i is the coefficient for task i.

Multi-tasking: Under the hood

- Computing gradients for shared parameters:

$$\nabla_\theta \mathcal{L} = \sum_{i=1}^{n} \alpha_i \nabla_\theta \mathcal{L}_i(y_i, \mathbf{p}) \tag{5}$$

- α_i are usually constrained to be the convex coefficients

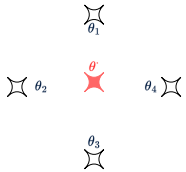
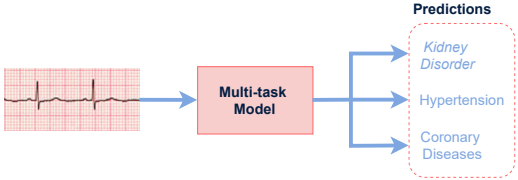


Figure: Deviation between optimal parameters and shared parameters learned by MTL.

Why do we need multi-tasking?



- Predicting multiple outcomes for a patient from a single input:
  - Avoids firing multiple models
  - Lesser storage complexity
- Regularises the training
  - Avoids over-fitting
  - Root cause of this regularisation is learning common representation
- Another way to look at it:
  - MTL adds noise to noisy gradients
  - Further increasing the chances of arriving at wider local minima

Challenges in Multi-tasking: Shared Representation

- Is it always possible to learn a good shared representation?
  - Of-course not!
- A good shared representation could be obtained if tasks are similar
  - Our and machine's perception of task similarity could be different!
- Less similar tasks result in shared parameters that are not be helpful to any task

Challenges in Multi-tasking: Optimisation Issues

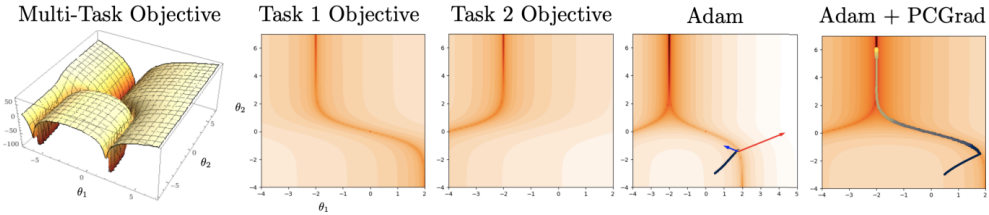


Figure: Gradient conflicts

Source: Yu et al., Gradient Surgery for Multi-Task Learning, Neurips 2020.

Challenges in Multi-tasking: Optimisation Issues

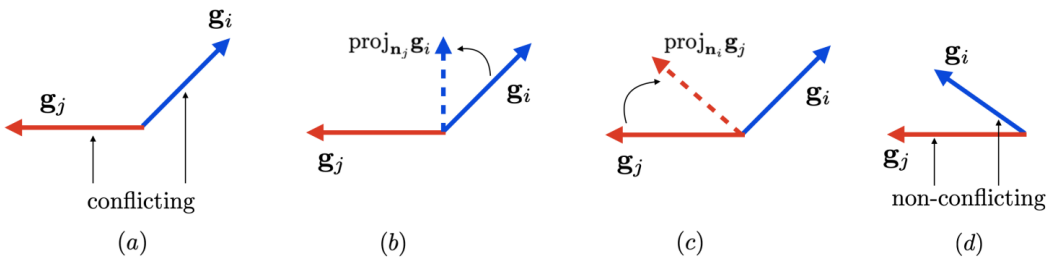


Figure: Gradient conflicts

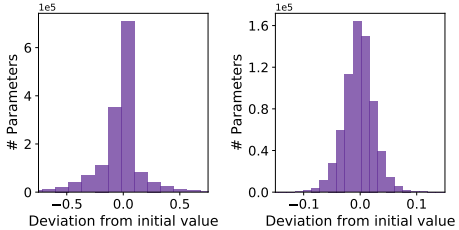
Source: Yu et al., Gradient Surgery for Multi-Task Learning, Neurips 2020.

Challenges in Multi-tasking: Label Availability

- Availability of all labels for an example is rare
- In healthcare informatics, it's possible that a patient may not exhibit all the outcomes that we are modelling
- MTL can only use a subset of all the available data

Lazy Training[2]

- Minimal or no deviation in majority of parameters



(a) DNN (b) LSTM

Figure: Histograms depicting deviation of trained parameters from their initial values in (a) 5-layered DNN trained on FashionMNIST dataset and (b) LSTM based model trained on MIMIC-III dataset for in-hospital mortality prediction.

[2] Chizat et al., On lazy training in differentiable programming, Neurips 2019.

Adaptive Parameter Optimisation: An Overview

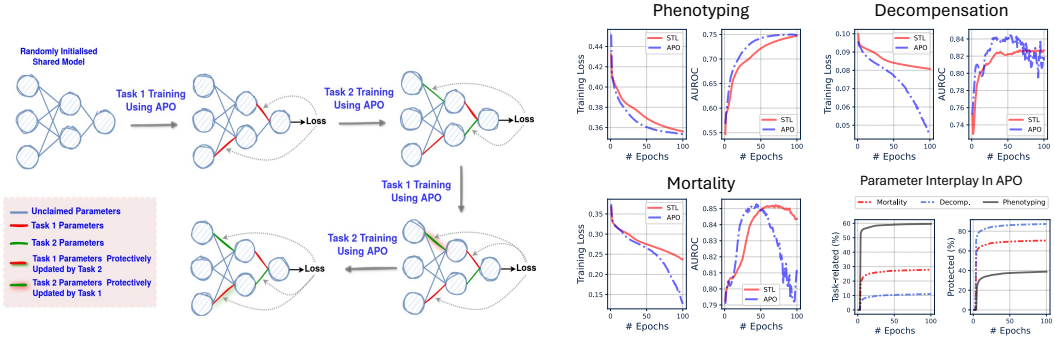


Figure: Illustration of APO[3] in action.

[3] Thakur et al., Information Transfer Across Clinical Tasks via Adaptive Parameter Optimisation, AISTATS 2025.

Task Groupings to Alleviate Negative Transfer

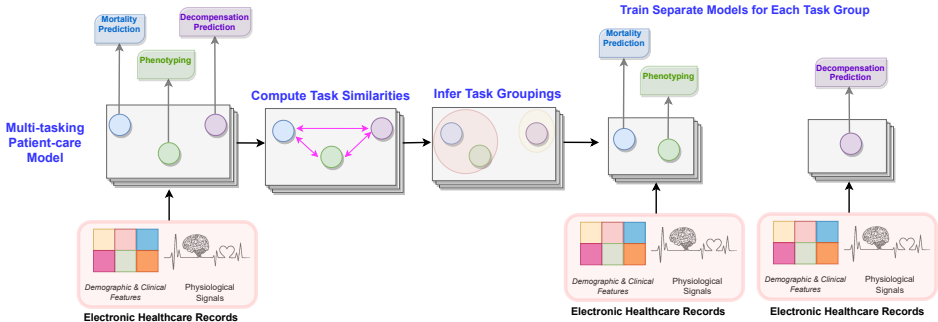


Figure: A typical task grouping framework[4].

[4] Thakur et al., Efficient Task Grouping Through Sample-wise Optimisation Landscape Analysis, IEEE TPAMI 2025.

Sample-wise optima

- $f_\theta(\cdot)$: neural network parameterised by θ, and θ_0 is the initial random state of the model
- i-th sample: $(\mathbf{x}_i, y_i^1, y_i^2, \ldots y_i^t)$ in dataset $\mathcal{D}$
- Task-specific sample-wise optima $\theta_i^{t\star}$:

$$\theta_i^{t\star} = \theta_0 - \nabla_\theta \ell(f_\theta(\mathbf{x}_i), y_i^t)$$

- Sample-wise optima $\theta_i^\star$:

$$\theta_i^\star = \frac{1}{T}\sum_{t=1}^{T} \theta_i^{t\star}$$

Sample-wise Convergence For Task Groupings

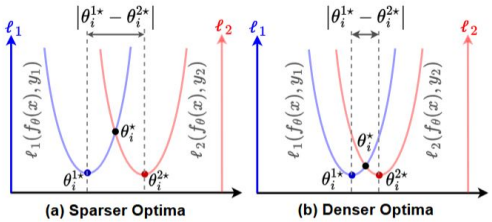


Figure: Illustration of the nature of shared global optima $\theta^\star$ as a function of the density of task-specific sample-wise local optima, $\theta_1^\star$ and $\theta_2^\star$ for tasks 1 and 2, respectively. In comparison to sparser local optima **(a)**, denser local optima **(b)** results in $\theta^\star$ that leads to better average loss across both tasks, $\mathcal{J} = (\ell_1 + \ell_2)/2$, as desired in multi-tasking and other information sharing frameworks.

Sample-wise Convergence For MTL

- Average loss bound:

$$\mathcal{J} \leq \frac{T^3}{2n} \sum_{i=1}^{n} \psi_{\theta_i}^2 + \frac{n^3}{2} \Psi_{\theta^\star}^2,$$

$$\psi_{\theta_i} = \frac{\sqrt{\mathcal{H}}}{T^2} \sum_{j=1}^{T} \sum_{k=1}^{T} \|\theta_i^{j\star} - \theta_i^{k\star}\|_1,$$

$$\Psi_{\theta^\star} = \frac{\sqrt{\mathcal{H}}}{n^2} \sum_{j=1}^{n} \sum_{k=1}^{n} \|\theta_j^\star - \theta_k^\star\|_1.$$

- SCA based task affinity between task τ_1 and τ_2:

$$a_{\tau_i,\tau_j} = \frac{1}{n} \sum_{i=1}^{n} \|\theta_i^{\tau_i\star} - \theta_i^{\tau_j\star}\|_1,$$

Results: Celeb-A dataset

Table: Performance of different task grouping methods on Celeb-A dataset in terms of (a) total absolute error and (b) average training time (RTX A6000 GPU hours).

(a) Total Absolute Error

Method ↓	Splits			
	2	3	4	5
Random	50.1 ± 0.085	49.97 ± 0.13	49.75 ± 0.08	49.73 ± 0.11
TAG	49.66 ± 0.095	49.55 ± 0.08	49.48 ± 0.066	49.46 ± 0.07
HOA	49.76 ± 0.08	49.74 ± 0.43	49.72 ± 0.1	49.69 ± 0.09
SCA (Prop.)	49.6 ± 0.24	49.58 ± 0.07	49.48 ± 0.05	49.44 ± 0.09

(b) Average Training Time

Method ↓	Splits			
	2	3	4	5
Random	3.52 ± 0.02	5.24 ± 0.02	6.98 ± 0.01	8.62 ± 0.02
TAG	5.4 ± 0.03	7.17 ± 0.04	8.61 ± 0.03	10.45 ± 0.02
HOA	57.6 ± 0.08	59.5 ± 0.06	61.2 ± 0.04	62.7 ± 0.06
SCA (Prop.)	3.61 ± 0.02	5.31 ± 0.03	7.06 ± 0.02	8.71 ± 0.01

Results: Celeb-A dataset

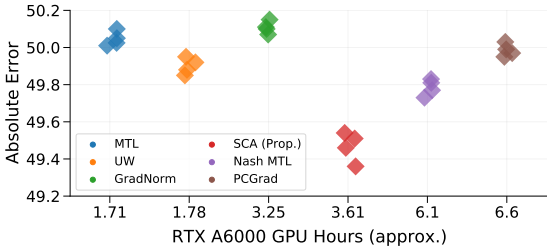


Figure: Performance and computational efficiency of the proposed *SCA*-based framework in comparison to prominent MTL methods on Celeb-A dataset.

Table of Contents

Meta-learning: Learning To Learn

- **Interpretation:** Training a model using gradients computed by other models
- **Purpose:** Learning a common model across multiple tasks
  - Global/common model can be quickly adapted to new unseen tasks
  - Proposed to mimic the learning in humans
- **Why are we interested in meta-learning?:**
  - Provides common/shared model across multiple tasks
  - Alleviate example-labels requirement of MTL
  - Information sharing across tasks in an example-independent manner

First Order Meta-learning: REPTILE[5]

- $f_\theta()$: Global model parameterised by θ
- $\mathcal{D}_t = \{\mathbf{x}_i, y_i\}_{i=1}^n$: Task t dataset

---

for $t \leftarrow 1 : T$ **do**

$\theta_t = \theta$

$\mathcal{B} \leftarrow$ SAMPLE-BATCHES($\mathcal{D}_t$)

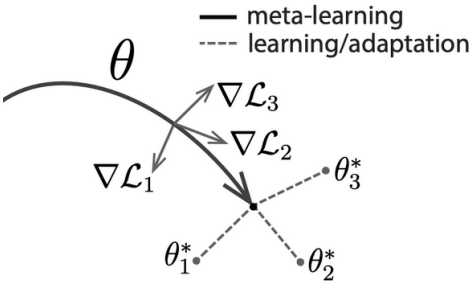
for any $(\mathbf{b}, \mathbf{l}) \in \mathcal{B}$ **do**

$L = \mathcal{L}(f_{\theta_t}(\mathbf{b}, \mathbf{l})$

$\theta_t = \theta_t - \alpha \nabla_{\theta_t} L$

$\phi = \sum_{t=1}^{T} (\theta_t - \theta)$

$\theta = \theta + \alpha\phi$



Multi-tasking to Meta-learning

- All inner-loop updates can be summaries as one single gradient update
- Meta-grad for task t, $\phi_t = -(\theta - \theta_t) = \nabla_{\theta_t} L$
- REPTILE update can be written as:

$$\nabla \mathcal{L}_\theta = \sum_{t=1}^{T} \nabla_{\theta_t} L = \sum_{t=1}^{T} \alpha_i \nabla_{\theta_t} L \tag{6}$$

- REPTILE/MAML, MTL and FedAvg are identical w.r.t. optimisation

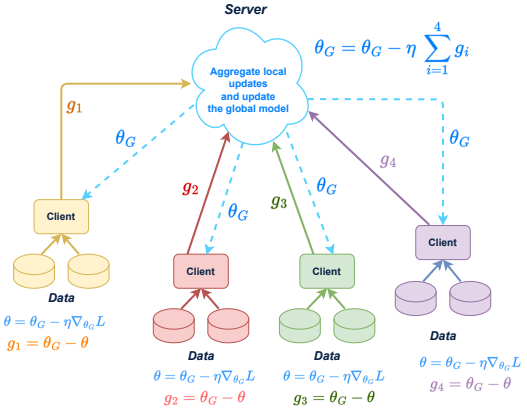
Table of Contents

Data Privacy Constraints in Healthcare

- Healthcare data is sensitive, contains patients' private information
- Protected by legislation such as GDPR and DATA PRIVACY ACTS
- Data is distributed across multiple sites such as the NHS Trusts, cannot be moved across sites
- Global trends may not be captured without the analysis of complete data
- Generalised predictive models should be a function of datasets distributed across sites

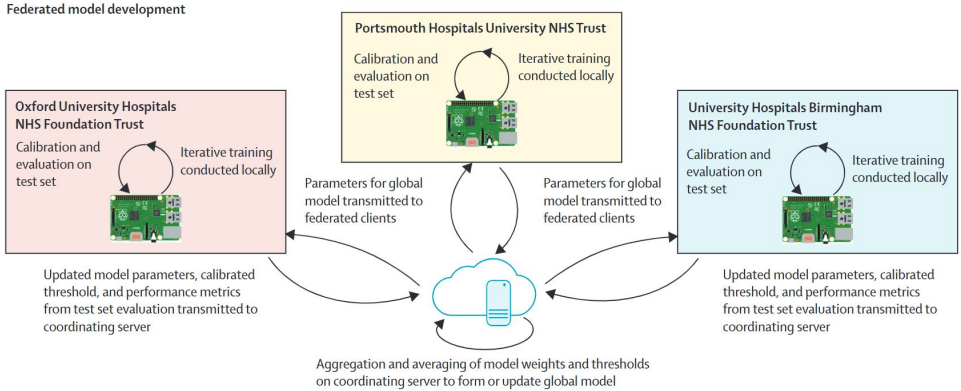
Federated Learning

- Distributed training of deep models using **FedSGD** or **FedAvg**
- **SGD:** $\theta' = \theta - \eta \nabla_\theta \mathcal{L}(f_\theta(\mathbf{x}), \mathbf{y})$
  - $(\mathbf{x}, \mathbf{y})$: Data, $f_\theta()$: Neural network parameterised by θ



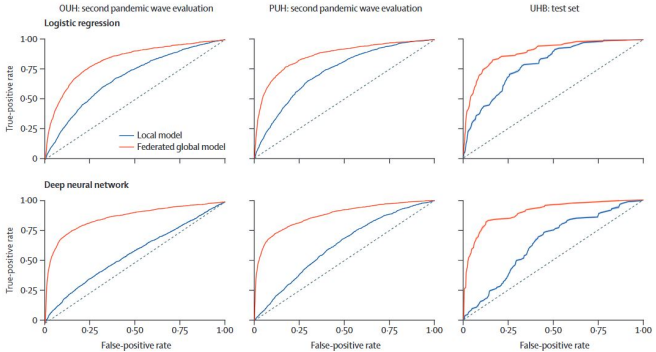
Thakur et al., *Dynamic neural graphs based federated reptile for semi-supervised multi-tasking in healthcare applications*, IEEE JBHI (2021)

CURIAL-FL[6]



[6] Soltan et al., A scalable federated learning solution for secondary care using low-cost microcomputing: privacy-preserving development and evaluation of a COVID-19 screening test in UK hospitals, The Lancet Digital Health 2023.

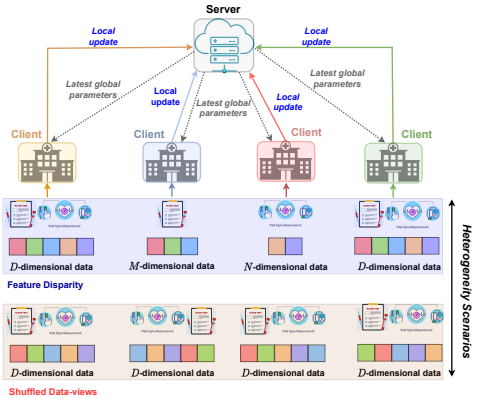
CURIAL-FL[7]



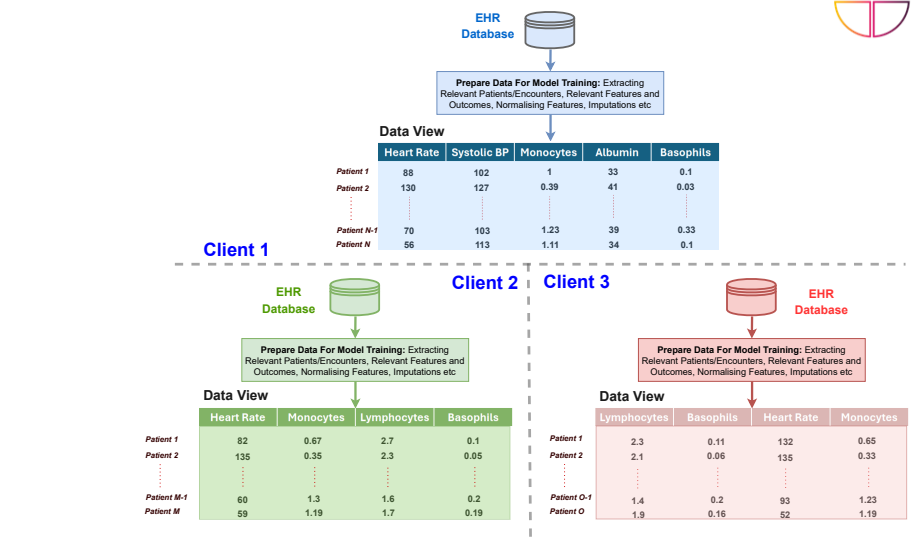
[7] Soltan et al., A scalable federated learning solution for secondary care using low-cost microcomputing: privacy-preserving development and evaluation of a COVID-19 screening test in UK hospitals, The Lancet Digital Health 2023.

Federated Learning: A Perfect Solution?

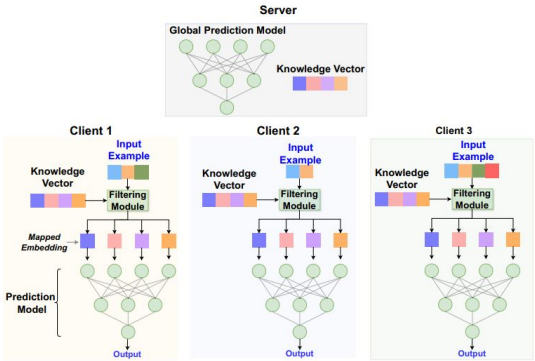
- Data-view heterogeneity: Massive pre-processing is required to align data-views across clients

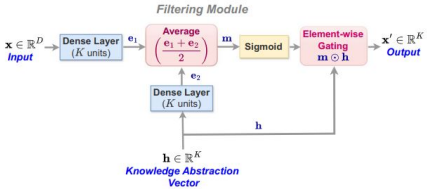


Federated Learning under Data-view Heterogeneity



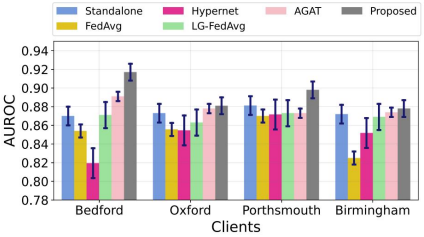
Federated Learning under Data-view Heterogeneity[8]

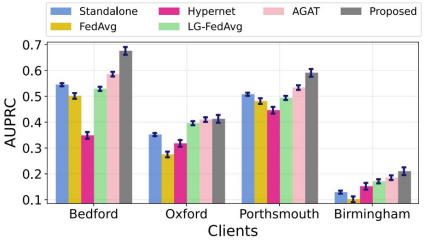




[8] Thakur et al., Knowledge abstraction and filtering based federated learning over heterogeneous data views in healthcare, npj Digital Medicine 2024.

Federated Learning under Data-view Heterogeneity





Federated Learning under Data Heterogeneity

- FedAvg or FedSGD may struggle under non-IID settings
- Client heterogeneity may force local models to deviate far away from each other

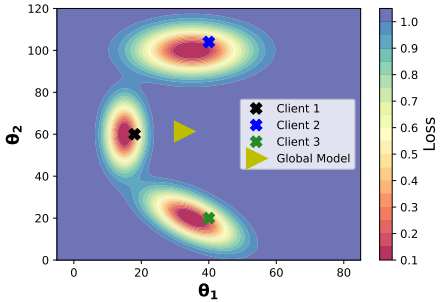


Figure: Global model drifts to high loss regions under client heterogeneity.

Optimising FL with Mode Connectivity

Key Idea: Improving Global Model Aggregation[9]

- **Aligning Client Models:** Leverage mode connections between the global model and clients' local models
- **Global Model Selection:** Choose the new global model **near the intersection** of these mode connections
- **Improved Fairness:** Ensures the global model resides in **a shared low-loss region** that benefits all clients

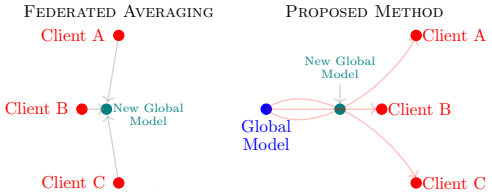


Figure: Fundamental difference between FedAvg and FedMode.

[9] Thakur et al., Optimising Clinical Federated Learning through Mode Connectivity-based Model Aggregation, AISTATS 2025.

Optimising FL with Mode Connectivity

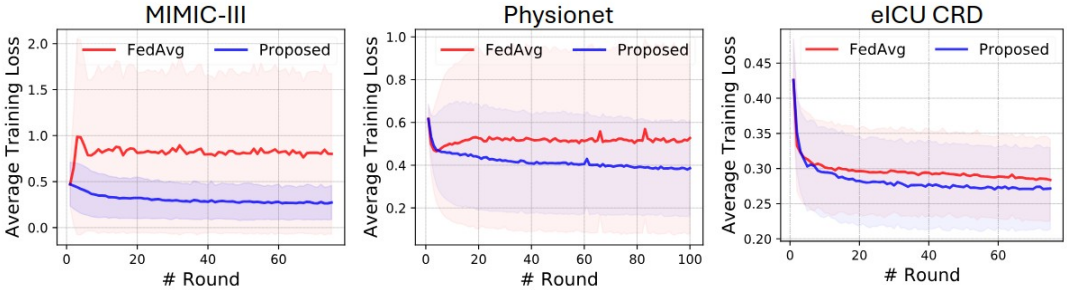


Figure: Training dynamics on 15 clients (MIMIC-III, Physionet) and 100 clients (eICU-CRD). MIMIC-III and Physionet exhibit higher heterogeneity.

Optimising FL with Mode Connectivity: Client Fairness

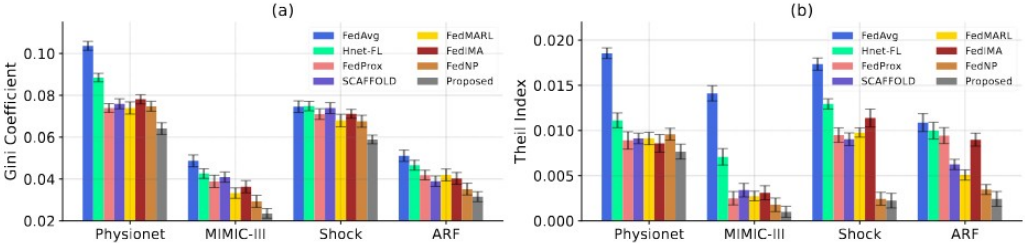
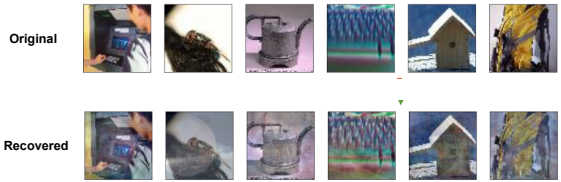


Figure: Client fairness across different FL methods.

Federated Learning: A Perfect Solution?

- Gradient Inversion:[10] Original samples can be reconstructed from the gradient updates
- Some success but can have large implications



[10] Huang, Yangsibo, et al. "Evaluating gradient inversion attacks and defences in federated learning." NeurIPS, 2021.

Federated Learning: A Perfect Solution?

- No true data democratisation
- All clients are forced to work towards common goal
- Can be difficult to convince institutions to participate

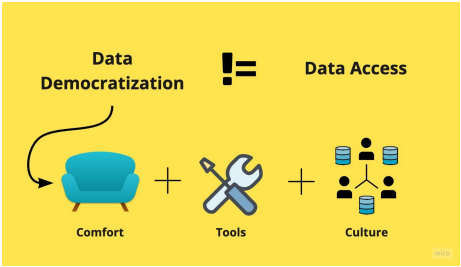


Table of Contents

Data Democratisation in Clinical Machine Learning

- **Data Democratisation:** improving data access for medical researchers
- In the context of **Clinical ML:** data access for AI/ML researchers
- Most ML tools for healthcare informatics are neither robust nor validated
- Data privacy laws understandably inhibit data sharing
- Healthcare data democratisation could stimulate
  - novel clinical ML algorithms
  - well-validated models
  - generic models that are trained on multiple populations

Data Democratisation in Clinical Machine Learning

- Data Democratisation is possible without violating privacy constraints
- Key is to find a way to train models without physical data access
  - Extracting imperceptible semantic information from sensitive data
  - Securely transferring this information to the intended researchers
  - Researchers then use this information to train models

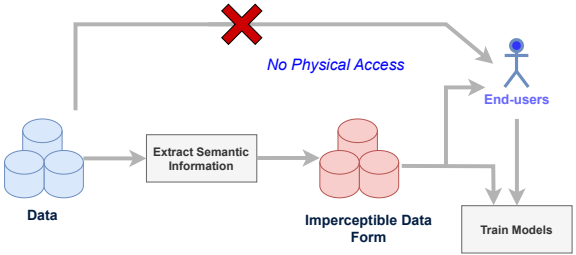
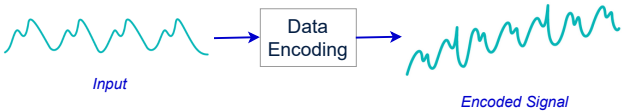


Table of Contents

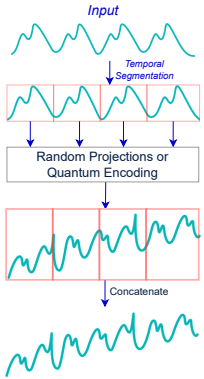
Data Encoding

- **FL:** Very restrictive, clients don't have independence
- **Data Encoding:** Transformation to make data "imperceptible" while preserving semantics
- Imperceptibility preserves data privacy
- Semantic preservation allows subsequent pattern analysis



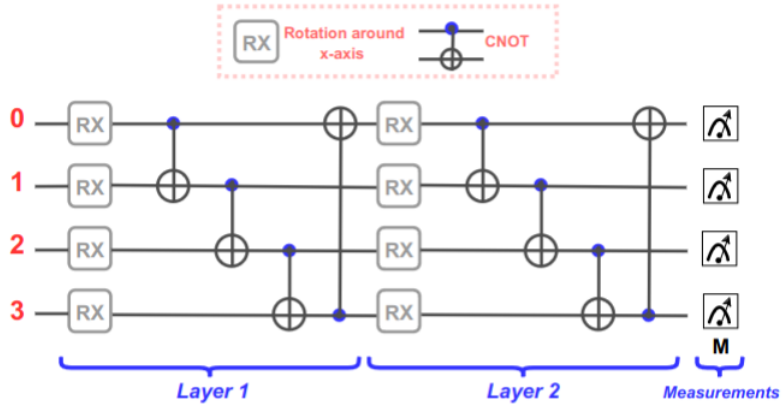
Data Encoding for Time-series Data

- Encoding framework:
  - Temporal segmentation of input signal
  - Apply data transformation on each segment
  - Concatenate the resulting segments
- Random Quantum Encoding and Random Projection as required transformations
- Each sample is a hash of its neighbours
- Loosing local structure to preserve global semantics



Thakur et al., *Data Encoding For Healthcare Data Democratisation and Information Leakage Prevention*, 2023.

Data Encoding for Time-series Data



Thakur et al., *Data Encoding For Healthcare Data Democratisation and Information Leakage Prevention*,2023.

Encoded Signals

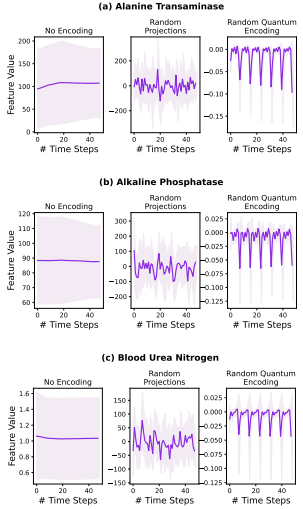
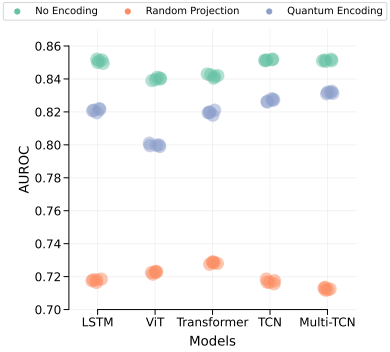
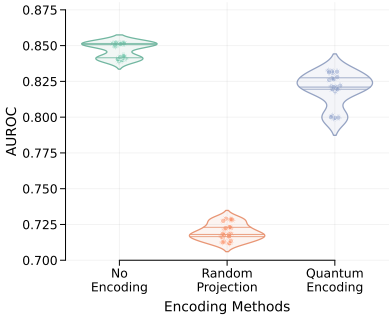


Figure: Average of 50 time series representing patients in the PhysioNet dataset.

Predictive Modelling on Encoded Data

- **MIMIC-III** dataset for mortality prediction: $21{,}156$ ICU stays
- **Models:** LSTM, Transformers, Vision Transformers (ViT), TCN & Multi-branch TCN

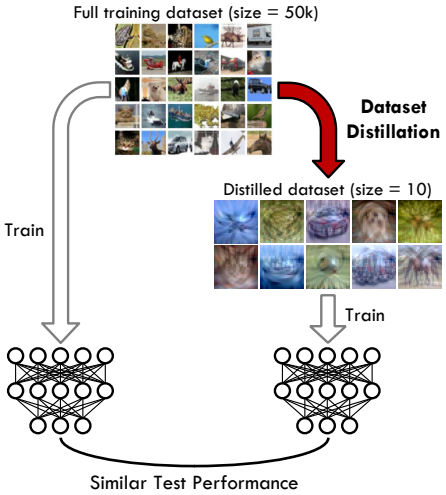




Mortality prediction on encoded MIMIC-III data.

Practicality outside academic circles?

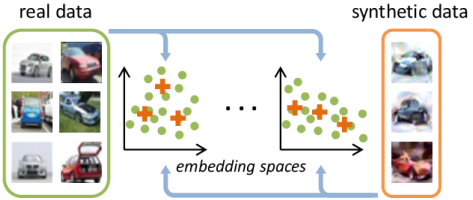
Dataset Condensation



- **Privacy-preserving Nature:** Real Samples ⇄ Condensed Samples
  - No direct one-to-one alignment
  - Obfuscation via aggregation
  - Reduced memorisation risks
- PRIVACY IS NO FREE LUNCH
- Differential Privacy provides theoretical privacy guarantees

Dataset Condensation using Distribution Matching

- Aligning outputs of a function $f()$ generated for real $\mathcal{D}$ and synthetic $\mathcal{S}$ data
- $\mathbf{e}_1 = \frac{1}{N}\sum_{\forall \mathbf{x} \in \mathcal{D}} \mathbf{x}$, $\mathbf{e}_2 = \frac{1}{N}\sum_{\forall \mathbf{x} \in \mathcal{S}} \mathbf{x}$
- $\mathcal{L} = \|\mathbf{e}_1 - \mathbf{e}_2\|_2^2$
- $\mathbf{S} = \mathbf{S} - \eta \nabla_{\mathcal{S}} \mathcal{L}(\mathbf{e}_1, \mathbf{e}_2)$



Dataset Condensation: Time-series Data

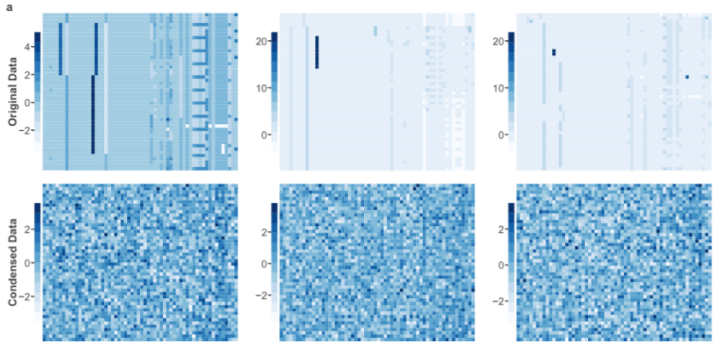
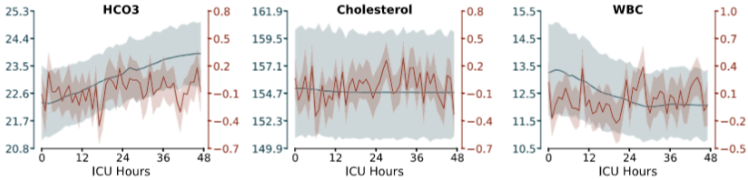


Figure: Heatmaps exhibiting changes in magnitude and trends of real and condensed time-series examples from MIMIC-III dataset.

Dataset Condensation: Time-series Data



Dataset Condensation: Time-series Data

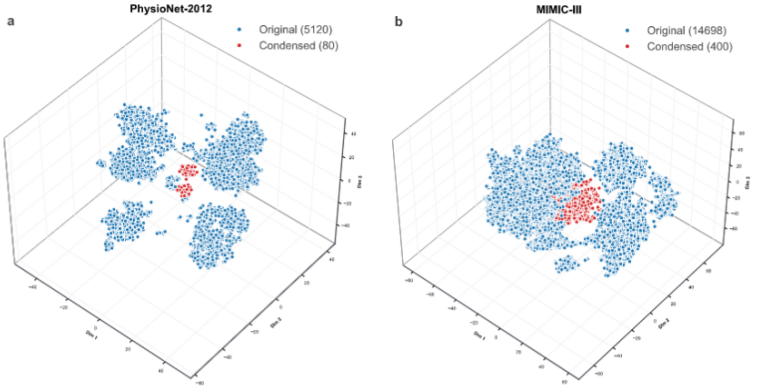


Figure: t-SNE representation of the average time-series examples.

Dataset Condensation: Predictive Performance

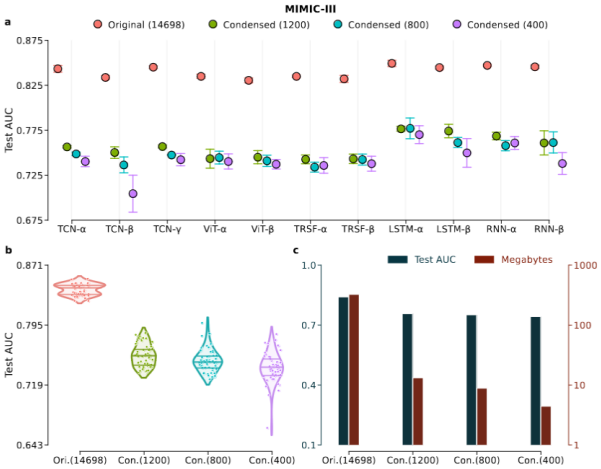


Figure: Performance on real and condensed data.

What is Trajectory Matching?

- State-of-the-art condensation methods that align synthetic data with real training trajectories
- Provide comparable performance to the real training data

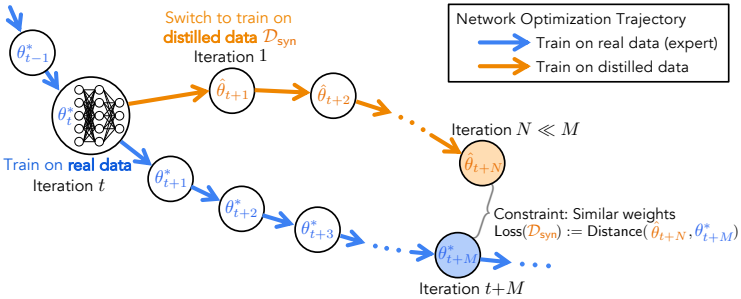
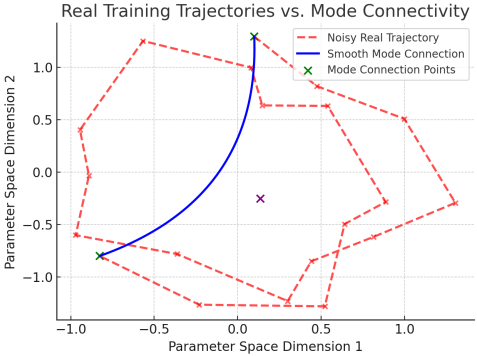


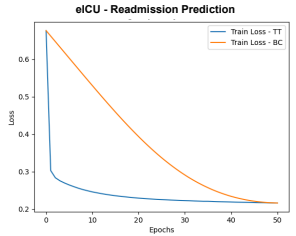
Figure: Illustration of trajectory matching strategy

Trajectory Matching: Drawbacks



- **High memory footprint:** Requires storing full training trajectories across multiple models
- **Noisy and unstable paths:** SGD trajectories are highly stochastic and non-smooth, reducing reliability for matching
- **Ambiguous temporal structure:** No clear notion of “early” or “late” stages in training, limiting interpretability and supervision quality

Mode Connections as Trajectory Surrogates [11]



Theorem

Let $\mathcal{L} : \Theta \to \mathbb{R}$ be a β-smooth, lower-bounded loss function, $\boldsymbol{\theta}_0 \in \Theta$ be a random initialization with loss $\ell_0 = \mathcal{L}(\boldsymbol{\theta}_0)$, and $\boldsymbol{\theta}_T$ be an SGD endpoint after K steps such that $\|\nabla \mathcal{L}(\boldsymbol{\theta}_T)\| \leq \varepsilon$, where $\mathcal{L}(\boldsymbol{\theta}_T) = \ell_T \ll \ell_0$. Assume $\gamma(t)$ denote the piecewise-linear interpolation of the SGD iterates $\{\boldsymbol{\theta}_k\}_{k=0}^{K}$. The optimized Bézier curve and its curvature as $\boldsymbol{\theta}^\star(t) := \boldsymbol{\theta}(t; \boldsymbol{\phi}^\star), \quad \kappa := 2\|\boldsymbol{\theta}_0 - 2\boldsymbol{\phi}^\star + \boldsymbol{\theta}_T\|$. Then,

(i) Average loss along the Bézier path is near-optimal:

$$\int_0^1 \mathcal{L}(\boldsymbol{\theta}^\star(t))\, dt \leq \int_0^1 \mathcal{L}(\gamma(t))\, dt + \frac{\beta \kappa^2}{240}. \tag{7}$$

(ii) Bézier path has lower and noise-free curvature compared to SGD trajectory:

$$\sup_{t \in [0,1]} \|\boldsymbol{\theta}^{\star\prime\prime}(t)\| = \kappa, \quad \mathbb{E}\left[\sup_t \|\gamma''(t)\|\right] \geq \kappa + c \cdot \sigma_{\text{sgd}} \tag{8}$$

(iii) Model predictions along Bézier path remain close to those along SGD:

$$\sup_{x \in \mathcal{X},\ t \in [0,1]} \|f_{\boldsymbol{\theta}^\star(t)}(x) - f_{\gamma(t)}(x)\| \leq \frac{L_f \kappa}{8}. \tag{9}$$

[11] Nganjimi et al., *Improving Clinical Dataset Condensation with Mode Connectivity-based Trajectory Surrogates*, arXiv 2025.

Mode Connections as Trajectory Surrogates: Results

DEPARTMENT OF ENGINEERING SCIENCE
UNIVERSITY OF OXFORD

(a) Oxford University Hospitals (OUH)

	AUROC				AUPRC			
Method	**50**	**100**	**200**	**500**	**50**	**100**	**200**	**500**
RANDOM	$0.835_{+0.021}$	$0.855_{+0.007}$	$0.869_{+0.006}$	$0.888_{+0.004}$	$0.158_{+0.036}$	$0.176_{+0.029}$	$0.219_{+0.024}$	$0.276_{+0.031}$
M3D	$0.840_{+0.004}$	$0.862_{+0.003}$	$0.872_{+0.003}$	$0.893_{+0.002}$	$0.162_{+0.015}$	$0.190_{+0.010}$	$0.266_{+0.010}$	$0.290_{+0.012}$
MTT	$0.824_{+0.016}$	$0.849_{+0.011}$	$0.855_{+0.008}$	$0.870_{+0.005}$	$0.356_{+0.019}$	$0.381_{+0.019}$	$0.405_{+0.012}$	$0.407_{+0.007}$
TESLA	$0.839_{+0.007}$	$0.875_{+0.002}$	$0.874_{+0.003}$	$0.880_{+0.002}$	$0.169_{+0.012}$	$0.205_{+0.009}$	$0.202_{+0.010}$	$0.217_{+0.011}$
FTD	$0.831_{+0.014}$	$0.847_{+0.006}$	$0.852_{+0.005}$	$0.860_{+0.006}$	$0.321_{+0.031}$	$0.382_{+0.012}$	$0.400_{+0.009}$	$0.400_{+0.023}$
DATM	$0.829_{+0.010}$	$0.844_{+0.007}$	$0.851_{+0.009}$	$0.872_{+0.009}$	$0.338_{+0.020}$	$0.394_{+0.009}$	$0.414_{+0.009}$	$0.409_{+0.004}$
BTM (OURS)	$0.854_{+0.012}$	$0.863_{+0.009}$	$0.876_{+0.00}$	$0.888_{+0.003}$	$0.394_{+0.014}$	$0.396_{+0.010}$	$0.427_{+0.006}$	$0.436_{+0.003}$
Full Dataset			$0.901_{+0.001}$				$0.445_{+0.004}$	

(b) Portsmouth University Hospitals (PUH)

	AUROC				AUPRC			
Method	**50**	**100**	**200**	**500**	**50**	**100**	**200**	**500**
RANDOM	$0.858_{+0.006}$	$0.867_{+0.004}$	$0.879_{+0.004}$	$0.893_{+0.004}$	$0.322_{+0.025}$	$0.347_{+0.011}$	$0.410_{+0.029}$	$0.473_{+0.018}$
M3D	$0.863_{+0.004}$	$0.883_{+0.003}$	$0.888_{+0.002}$	$0.900_{+0.001}$	$0.348_{+0.013}$	$0.405_{+0.011}$	$0.416_{+0.011}$	$0.527_{+0.011}$
MTT	$0.845_{+0.009}$	$0.871_{+0.005}$	$0.887_{+0.005}$	$0.895_{+0.004}$	$0.495_{+0.013}$	$0.517_{+0.012}$	$0.529_{+0.013}$	$0.544_{+0.018}$
TESLA	$0.861_{+0.006}$	$0.861_{+0.006}$	$0.885_{+0.004}$	$0.893_{+0.002}$	$0.412_{+0.019}$	$0.385_{+0.016}$	$0.472_{+0.009}$	$0.483_{+0.009}$
FTD	$0.845_{+0.007}$	$0.870_{+0.007}$	$0.885_{+0.006}$	$0.899_{+0.003}$	$0.499_{+0.013}$	$0.536_{+0.010}$	$0.550_{+0.010}$	$0.584_{+0.007}$
DATM	$0.871_{+0.005}$	$0.876_{+0.004}$	$0.887_{+0.004}$	$0.889_{+0.003}$	$0.540_{+0.007}$	$0.555_{+0.008}$	$0.555_{+0.008}$	$0.579_{+0.005}$
BTM (OURS)	$0.881_{+0.007}$	$0.895_{+0.007}$	$0.902_{+0.003}$	$0.905_{+0.002}$	$0.564_{+0.011}$	$0.575_{+0.008}$	$0.593_{+0.009}$	$0.603_{+0.007}$
Full Dataset			$0.906_{+0.002}$				$0.610_{+0.004}$	

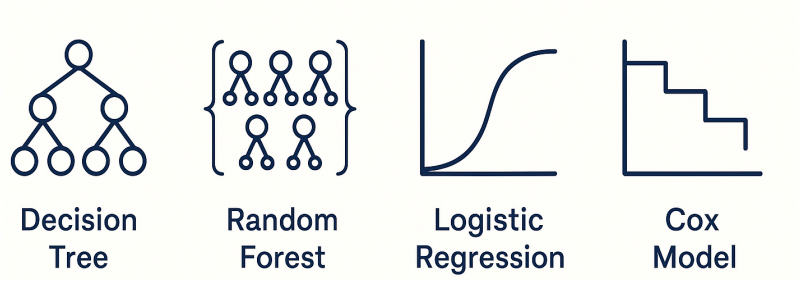
Mode Connections as Trajectory Surrogates: Results

(c) University Hospitals Birmingham (UHB)

	AUROC				AUPRC			
Method	**50**	**100**	**200**	**500**	**50**	**100**	**200**	**500**
RANDOM	$0.847_{\pm 0.015}$	$0.856_{\pm 0.010}$	$0.876_{\pm 0.007}$	$0.891_{\pm 0.005}$	$0.089_{\pm 0.024}$	$0.108_{\pm 0.018}$	$0.126_{\pm 0.019}$	$0.153_{\pm 0.015}$
M3D	$0.863_{\pm 0.005}$	$0.853_{\pm 0.003}$	$0.872_{\pm 0.004}$	$0.891_{\pm 0.003}$	$0.107_{\pm 0.005}$	$0.070_{\pm 0.006}$	$0.107_{\pm 0.008}$	$0.153_{\pm 0.006}$
MTT	$0.802_{\pm 0.008}$	$0.847_{\pm 0.013}$	$0.871_{\pm 0.013}$	$0.884_{\pm 0.007}$	$0.092_{\pm 0.065}$	$0.228_{\pm 0.021}$	$0.242_{\pm 0.016}$	$0.234_{\pm 0.017}$
TESLA	$0.820_{\pm 0.001}$	$0.850_{\pm 0.004}$	$0.859_{\pm 0.004}$	$0.873_{\pm 0.006}$	$0.099_{\pm 0.001}$	$0.079_{\pm 0.006}$	$0.125_{\pm 0.008}$	$0.131_{\pm 0.014}$
FTD	$0.830_{\pm 0.005}$	$0.839_{\pm 0.010}$	$0.847_{\pm 0.010}$	$0.872_{\pm 0.010}$	$0.122_{\pm 0.056}$	$0.218_{\pm 0.013}$	$0.216_{\pm 0.018}$	$0.233_{\pm 0.012}$
DATM	$0.828_{\pm 0.028}$	$0.832_{\pm 0.022}$	$0.843_{\pm 0.015}$	$0.870_{\pm 0.005}$	$0.152_{\pm 0.019}$	$0.176_{\pm 0.022}$	$0.221_{\pm 0.017}$	$0.237_{\pm 0.017}$
BTM (OURS)	$0.842_{\pm 0.011}$	$0.836_{\pm 0.023}$	$0.884_{\pm 0.007}$	$0.891_{\pm 0.006}$	$0.258_{\pm 0.006}$	$0.246_{\pm 0.013}$	$0.253_{\pm 0.011}$	$0.272_{\pm 0.006}$
Full Dataset	$0.895_{\pm 0.005}$				$0.284_{\pm 0.005}$			

Dataset Condensation for Classical Clinical Models

- **Classical AI models:** State-of-the-art (still!), regulatory familiarity and interpretability
- **Non-differentiable**: Existing DC methods require models to be differentiable
- Model-specific inductive biases can hinder condensed data portability



Zero-order DC

- Real clinical data $\mathcal{X}_{\text{real}} = (\mathbf{X}, \mathbf{y})$, Synthetic dataset $\mathcal{X}_{\text{syn}} = (\mathbf{X}_{\text{syn}}, \mathbf{y}_{\text{syn}})$
- **Train a model on real data:** $f_{\text{real}}(\cdot)$ (e.g., XGBoost) trained on $\mathcal{X}_{\text{real}}$
- **Initialise synthetic data:** $\mathbf{X}_{\text{syn}} \in \mathbb{R}^{m \times d}, \quad \mathbf{y}_{\text{syn}} \in \{0,1\}^m$
- **Composite loss:** $\ell = \ell_{\text{pred}} + \alpha \cdot \ell_{\text{match}}$

$$\ell_{\text{pred}} = \frac{1}{m} \sum_{i=1}^{m} \left[-y_i \log f_{\text{real}}(\mathbf{x}_i) - (1 - y_i) \log \left(1 - f_{\text{real}}(\mathbf{x}_i)\right) \right]$$

$$\ell_{\text{match}} = \sum_{c \in \{0,1\}} \left| \frac{1}{|\mathcal{I}_c^{\text{syn}}|} \sum_{i \in \mathcal{I}_c^{\text{syn}}} f_{\text{real}}(\mathbf{x}_i^{\text{syn}}) - \frac{1}{|\mathcal{I}_c^{\text{real}}|} \sum_{j \in \mathcal{I}_c^{\text{real}}} f_{\text{real}}(\mathbf{x}_j^{\text{real}}) \right|$$

- **Zero-order gradient:**

$$\nabla_{\mathbf{X}_{\text{syn}}} \ell = \left(\frac{\partial \ell}{\partial f_{\text{real}}(\mathbf{X}_{\text{syn}})} \right) \cdot \left(\frac{\partial f_{\text{real}}(\mathbf{X}_{\text{syn}})}{\partial \mathbf{X}_{\text{syn}}} \right)$$

- *Finite-diff approximation:*

$$\frac{\partial f_{\text{real}}(\mathbf{X}_{\text{syn}})}{\partial \mathbf{X}_{\text{syn},j}} \approx \frac{f_{\text{real}}(\mathbf{X}_{\text{syn}} + \epsilon_j \mathbf{E}_j) - f_{\text{real}}(\mathbf{X}_{\text{syn}} - \epsilon_j \mathbf{E}_j)}{2\epsilon_j}$$

- **Update rule:**

$$\mathbf{X}_{\text{syn}}^{(t+1)} = \mathbf{X}_{\text{syn}}^{(t)} - \eta \nabla_{\mathbf{X}_{\text{syn}}} \ell$$

Zero-order DC: Results

- **Datasets:** CURIAL datasets (PUH, OUH) and Proteomics
- **Baseline:** Real dataset

Portsmouth

Instances	Metrics					Privacy Budget (ϵ)
	AUROC	Sensitivity	Specificity	PPV	NPV	
50	0.862 ± 0.023	0.733 ± 0.042	0.839 ± 0.009	0.205 ± 0.013	0.982 ± 0.003	2.5
100	0.894 ± 0.021	0.756 ± 0.044	0.887 ± 0.008	0.286 ± 0.018	0.985 ± 0.003	2.6
500	0.888 ± 0.022	**0.782** ± 0.042	0.871 ± 0.008	0.271 ± 0.017	**0.986** ± 0.003	1.8
1000	0.884 ± 0.022	0.741 ± 0.041	0.868 ± 0.007	0.262 ± 0.016	0.983 ± 0.003	3.1
Full Dataset	**0.901** ± 0.022	0.751 ± 0.045	**0.905** ± 0.006	**0.316** ± 0.026	0.985 ± 0.003	–

Oxford

Instances	Metrics					Privacy Budget (ϵ)
	AUROC	Sensitivity	Specificity	PPV	NPV	
50	0.855 ± 0.019	0.752 ± 0.037	0.819 ± 0.007	0.067 ± 0.004	0.995 ± 0.001	2.2
100	0.873 ± 0.019	0.744 ± 0.038	0.847 ± 0.003	0.077 ± 0.004	0.995 ± 0.001	2.7
500	0.870 ± 0.019	0.725 ± 0.039	0.843 ± 0.004	0.073 ± 0.004	0.994 ± 0.001	2.1
1000	0.891 ± 0.018	**0.760** ± 0.038	0.887 ± 0.005	0.095 ± 0.005	0.995 ± 0.001	1.9
Full Dataset	**0.911** ± 0.018	0.756 ± 0.035	**0.898** ± 0.003	**0.126** ± 0.008	**0.996** ± 0.001	–

Proteomics

Instances	Metrics					Privacy Budget (ϵ)
	AUROC	Sensitivity	Specificity	PPV	NPV	
50	0.856 ± 0.053	0.788 ± 0.106	0.817 ± 0.003	0.014 ± 0.002	0.999 ± 0.001	2.1
100	0.905 ± 0.048	0.846 ± 0.093	0.785 ± 0.006	0.013 ± 0.001	0.999 ± 0.001	2.6
500	**0.913** ± 0.044	**0.865** ± 0.092	0.834 ± 0.006	0.017 ± 0.002	0.999 ± 0.001	1.9
1000	0.912 ± 0.052	0.808 ± 0.096	**0.879** ± 0.005	**0.022** ± 0.003	0.999 ± 0.001	2.3
Full Dataset	0.898 ± 0.050	0.846 ± 0.091	0.757 ± 0.006	0.011 ± 0.001	**0.999** ± 0.001	–

Zero-order DC: External Validation using CURIAL

Model	PUH → UHB					OUH → UHB				
	AUROC	Sensitivity	Specificity	PPV	NPV	AUROC	Sensitivity	Specificity	PPV	NPV
Full	0.830 ± 0.037	0.631 ± 0.074	**0.866** ± 0.005	0.038 ± 0.005	0.996 ± 0.001	0.842 ± 0.040	0.611 ± 0.077	**0.917** ± 0.004	**0.058** ± 0.008	0.996 ± 0.001
IPC 50	0.855 ± 0.033	0.758 ± 0.064	0.847 ± 0.006	0.040 ± 0.004	**0.998** ± 0.001	0.825 ± 0.037	0.711 ± 0.074	0.814 ± 0.006	0.031 ± 0.003	0.997 ± 0.001
IPC 100	0.860 ± 0.035	0.765 ± 0.070	0.840 ± 0.004	0.039 ± 0.003	**0.998** ± 0.001	0.831 ± 0.035	0.691 ± 0.074	0.863 ± 0.005	0.041 ± 0.004	0.997 ± 0.001
IPC 500	0.847 ± 0.038	0.698 ± 0.074	0.836 ± 0.003	0.035 ± 0.004	0.997 ± 0.001	**0.864** ± 0.033	**0.772** ± 0.064	0.833 ± 0.006	0.037 ± 0.003	**0.998** ± 0.001
IPC 1000	**0.861** ± 0.032	**0.805** ± 0.064	0.845 ± 0.005	**0.042** ± 0.003	**0.998** ± 0.001	0.844 ± 0.035	0.745 ± 0.070	0.862 ± 0.005	**0.043** ± 0.004	**0.998** ± 0.001

Model	PUH → OUH					UHB → OUH				
	AUROC	Sensitivity	Specificity	PPV	NPV	AUROC	Sensitivity	Specificity	PPV	NPV
Full	0.863 ± 0.021	0.705 ± 0.041	**0.885** ± 0.003	**0.095** ± 0.006	0.994 ± 0.001	0.778 ± 0.020	0.326 ± 0.041	**0.953** ± 0.002	**0.085** ± 0.013	0.988 ± 0.001
IPC 60	0.862 ± 0.019	0.723 ± 0.039	0.844 ± 0.004	0.074 ± 0.004	0.994 ± 0.001	0.806 ± 0.022	0.502 ± 0.043	0.891 ± 0.003	0.070 ± 0.006	0.991 ± 0.001
IPC 100	**0.873** ± 0.019	0.721 ± 0.039	0.874 ± 0.004	0.089 ± 0.005	**0.995** ± 0.001	**0.821** ± 0.021	**0.613** ± 0.040	0.870 ± 0.004	0.072 ± 0.005	**0.993** ± 0.001
IPC 500	0.848 ± 0.020	0.727 ± 0.039	0.823 ± 0.004	0.066 ± 0.004	0.994 ± 0.001	0.806 ± 0.022	0.525 ± 0.041	0.886 ± 0.003	0.070 ± 0.005	0.991 ± 0.001
IPC 1000	0.851 ± 0.020	**0.729** ± 0.039	0.848 ± 0.004	0.076 ± 0.004	**0.995** ± 0.001	0.803 ± 0.022	0.517 ± 0.042	0.871 ± 0.004	0.062 ± 0.005	**0.991** ± 0.001

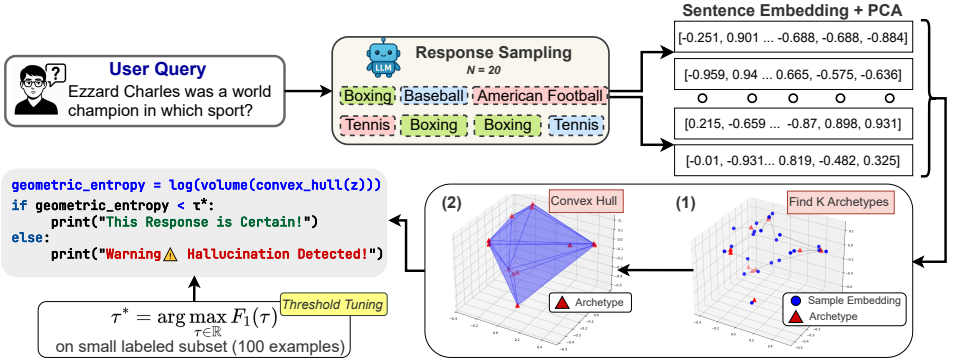
Table of Contents

Uncertainty in LLMs and Its Link to Hallucinations

- LLMs are probabilistic systems that generate outputs based on learned distributions
- **Uncertainty** arises when the model has low confidence in its output or faces ambiguous input
- Understanding uncertainty is crucial to detecting hallucinations — responses that are incorrect, fabricated, or misleading.

Type	Description	Example	Link to Hallucination
External Uncertainty	Ambiguity or underspecified user intent	`Who played Spiderman?`	Model guesses instead of clarifying intent
Internal Uncertainty	Missing, outdated, or conflicting model knowledge	`What's the latest COVID variant?`	Model fills gap with hallucinated facts

Simplical Modelling-Based Uncertainty Estimation [12]



[12] Philips et al., *Geometric Uncertainty for Detecting and Correcting Hallucinations in LLMs*, arXiv 2025.

Benchmarks and Results: Global Uncertainty

DEPARTMENT OF ENGINEERING SCIENCE
UNIVERSITY OF OXFORD

Experimental Setup

Benchmark	LLM(s)
CLAMBER	GPT-4o Mini
TriviaQA	LLaMA-3.2B Instruct
K-QA	GPT-3.5 Turbo, GPT-4o Mini
Med-QA	GPT-4o Mini

Sentence Encoder: `gte-Qwen2-1.5B-Instruct`

AUROC and F1 Score Comparison

AUROC

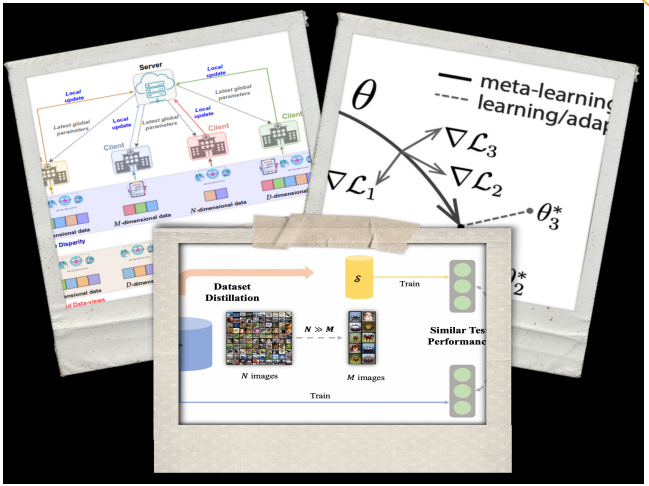
Dataset	p(true)	Semantic Entropy	Semantic Volume	Ours
CLAMBER	$58.5_{0.4}$	$48.1_{0.8}$	$55.8_{0.9}$	$\mathbf{61.7_{0.7}}$
TriviaQA	$73.3_{5.6}$	$\mathbf{79.3_{0.2}}$	$75.5_{0.3}$	$75.6_{0.4}$
K-QA (3.5)	$60.5_{1.2}$	$66.4_{1.2}$	$65.3_{0.7}$	$\mathbf{67.6_{1.2}}$
K-QA (4o)	$63.4_{1.5}$	$50.5_{5.7}$	$65.2_{1.1}$	$\mathbf{69.7_{2.3}}$
Med-QA (4o)	$59.4_{0.9}$	$59.3_{1.1}$	$59.4_{1.2}$	$\mathbf{62.3_{1.3}}$

F1 Score

Dataset	p(true)	Semantic Entropy	Semantic Volume	Ours
CLAMBER	$47.0_{1.6}$	$63.9_{0.3}$	$\mathbf{68.5_{0.4}}$	$66.1_{0.3}$
TriviaQA	$69.4_{3.0}$	$\mathbf{74.9_{0.3}}$	$73.4_{1.6}$	$74.7_{0.3}$
K-QA (3.5)	$71.8_{1.4}$	$40.3_{6.9}$	$74.8_{1.0}$	$\mathbf{75.6_{0.9}}$
K-QA (4o)	$61.5_{4.7}$	$70.4_{4.8}$	$78.2_{1.6}$	$\mathbf{79.2_{1.4}}$
Med-QA (4o)	$\mathbf{73.6_{1.2}}$	$70.9_{1.1}$	$72.6_{1.2}$	$73.1_{1.4}$

Note: Bold = highest per row. Subscripts = standard deviation.

Thank You!


Server
Local update
Latest global parameters
Client
M-dimensional data
N-dimensional data
Dataset Distillation
N images
N ≫ M
M images
Train
Similar Tes
Performanc
meta-learnin
learning/adap