Domain Complexity Estimation for Distributed AI Systems in Open-World

Shafkat Islam*, KMA Solaiman*, Ruy de Oliveira*, Bharat Bhargava*
*Purdue University, West Lafayette, USA

Abstract—**Although Artificial intelligence (AI) systems have been widely deployed in many applications, they still face difficulties in quantitatively specifying how distinct datasets or environments differ, i.e., if one environment is faster (or more complex) at learning than another. As the frequency of rare or unexpected events increases in open-world, understanding the inherent characteristics of the task domain is essential to model the domain effectively and is needed for accurate prediction. This work proposes a framework for measuring an application-independent complexity metric for the AI systems corresponding to the perception domain. The target environment space is characterized by distributed datasets where the overall complexity cannot be computed by existing approaches dealing with singular local datasets. We propose a complexity measure for distributed AI environments using inherent dataset properties such as dimensionality, heterogeneity, and sparsity. We use federated learning as the reference paradigm to handle distributed dataset operations since federated learning has been widely used in edge AI due to data privacy concerns. We propose a hierarchical architecture to model the federated training phase. We define the relationships between intrinsic properties and the environment features in a distributed setting with the proposed metric. We conduct experiments on three variants of the MNIST dataset with increasing complexity and measure the domain complexities independent of any classifiers. Empirical evaluations show the correlation of** $(R^2 =)$ **0.85 of the proposed metric with the reference federated model. The evaluation results imply that effectively choosing a distributed learning model (or distributed dataset) can fasten federated learning.**

Impact Statement—**Training an AI system in a closed world (or lab environment) is different than training in an open world since multiple unexpected environmental factors, such as rare or unusual events, can occur more frequently in the open world. However, the problem intensifies if the training occurs in a distributed manner (i.e., federated learning) since the training data (or environment) remains distributed without central supervision. Hence, quantifying the complexity (or quality) of training data or environment is essential since it can assist the training phase by appropriately choosing a distributed training policy. To address the issue, we propose a complexity measurement technique for distributed AI environments, which shows a correlation of** $(R^2 =)$ 0.81 **and** 0.85 **with the maximum and average distributed (or federated) accuracy. This metric can measure the complexity of multiple federated training paths beforehand, thus enabling the choosing of the low complex path for the training if required.**

Index Terms—**Open-world AI, Domain complexity, Federated learning complexity, Edge-centric AI, Distributed AI, Perception Domain**

I. Introduction

While a wide amount of research and progress has been made on Artificial Intelligence (AI) agents, there is still a lot to be done to make such intelligent agents capable of effectively dealing with the uncertainties in the Open world [1], [2]. The inherent relationship between the agent and its perceived environment is crucial in designing an adaptable agent. To some extent, this issue has recently been addressed in [3], prompting a full complexity investigation of the perception domain. AI agents in simulated environments encounter much fewer possible states and accomplish reasoning on smaller sets of possible state-action sets than they would face in the natural environment. As stated in [3], it is essential to understand the complexity level of a domain as it helps characterizing and defining uncertainties in that domain, and this is a crucial prerequisite for a robust transition from restricted domain to the open world. We need to understand the impact of domain complexity on algorithm choice to avoid building suboptimal decision-making systems, which could have fatal consequences in critical applications such as self-driving cars [4].

Our objective in this work is to compare the unconditioned distribution of data between any two datasets and, in turn, use that to compare different training settings for distributed learning. In distributed learning, the training phase can choose one of the multiple possible learning paths, which provides an advantage but complicates the learning process. Our interest is in analyzing the comparative complexity of different distributed settings based on the data complexity and the training environment variables. We want to analyze whether this aspect of complexity impacts the learning ability of a distributed system. We have used concepts from information theory, statistics, collaborative learning, and open-world AI to capture the essence of domain complexity without the classifier information.

In this work, we focus on the benchmark datasets widely used in training various computer vision tasks, where finding the appropriate class label of images in a relatively sparse dataset is of high interest. This can easily be extended to autonomous driving datasets, where image semantic segmentation is more important. Towards the former problem, a number of variants have been released with increasing complexity. A good classification performance on image datasets informs the feasibility of building perception algorithms, distinguishing novel elements in perceived environments, and characterizing different groups of features in the environment. Therefore, by analyzing the inherent complexity in these types of datasets, we can realize how hard it is to learn from a dataset compared to others and hence provides insights into building generalizable agents for the problem of perception in open-world.

While there have been some efforts to evaluate the complexity of various datasets from the agent perspective [5]–[7], we need ways to assess the intrinsic features of individual datasets toward a full complexity realization since this will render AI systems much more efficient. Intrinsic complexity metrics refer to complexities that arise from the data distribution, environment, or properties of the data. Approaches proposed in recent years [7]–[11] focused on local datasets used by single agents. To complement these research efforts, environments relying on distributed datasets (used by multiple agents) must be addressed, as their complexity computation is not straightforward. In this paper, we show that in such a distributed environment, it is also necessary to consider the entities in the environment.

We use federated learning (FL) [12] for training on a distributed dataset. The main reason for using federated learning is its rising popularity as a robust architecture for securely, distributed, and efficiently training AI systems. In other words, we are addressing what is emerging in the open world today regarding dataset structure for training modern AI systems. Our final objective is to build, test, and demonstrate complexity metrics for the data distribution in large volume, high dimensional image datasets in singular and distributed environment settings.

In practice, different entities of federated learning [13] environment engage in the training phase depending on its availability. At each communication round, the federated server can choose which entity to incorporate in training while leaving others. Hence, depending on the federated entity or client's availability and the server's choice, the training path of an identical federated task can be different. However, different learning paths of federated training may not be equally difficult or achieve similar generalizations. Therefore, the federated server needs to realize the complexity of a learning path before training. To address the issue, we propose a domain complexity metric for measuring the difficulty of each federated learning path and, thus, the complexity of the overall federated learning task.

Our proposed approach to measure the domain complexity is dataset-independent. First, we proposed different metrics for the complexity measurements separated into three inherent aspects of the domain: dimensionality, sparsity, and heterogeneity. Then, we proposed an effective complexity metric in distributed settings using the intrinsic complexity of a dataset and the properties of the distributed environment. We formulate the distributed complexity function as an augmentation of the L_2 norm of intrinsic properties in the domain space and inverse of the entity number in the distributed environment. The FL learning in place relies on a shallow CNN [14] to perform its training, which renders the approach mostly agent-independent. During the computation of the proposed complexity metric, intrinsic properties of the dataset are used from the first step as a part of the augmented complexity in the distributed paradigm. Specifically, we are interested in establishing a procedure for perception domain complexity evaluations in singular and distributed settings since this is an essential open issue to address in AI systems development.

Following are the main contributions of our work:

- We propose an application-independent framework for the intrinsic domain complexity measurements of the perception domain, where we considered upper and lower bounds for dimensionality and linear vs non-linear methods for sparsity and heterogeneity.
- We propose a complexity measurement metric for distributed federated environment in perception domain while combining the intrinsic components and distributed environment features of federated learning. We also propose a hierarchical architecture for modeling the available federated training paths.
- We conducted extensive experiments on MNIST, Fashion-MNIST, and EMNIST-digits in distinct distributed settings and performed ablation study to measure the impact of each components of our proposed metric on the distributed domain complexity.

II. Related Works

Domain complexity has been evaluated in various contexts. The AI field typically involves an environment from which we need to get information about how organized the data are in such a structure. The effort to get effective information impacts agents' adaptability to assigned tasks in the open-world environment. There has been a myriad of work for defining, detecting, measuring novelty in AI-based open-world systems [7], [10], [11], [15]–[22]. Most of these approaches deal with open-world novelties as something abnormal that the intelligent system must manage. A lot of work on the open world has been devoted to game applications, where the environment space is limited. Still, even so, such a strategy is interesting as it permits playing with a diverse range of potentially adaptable solutions in a controlled environment [15], [20], [23]–[27]. There are also other directions focused on information theory, algorithmic information theory, uncertainty to measure dataset complexity [24], [28]–[36]. These works typically aim at singular or single agents' perspectives in the open world.

A formalization for defining open-world uncertainties to unify novelty concepts was proposed in [19] by a framework that provides functions to evaluate if a given input is novel. Using the proposed framework, they formally define multiple types of novelty an agent can encounter. The formalism relies on dissimilarity and regret measures and considers novelty in the world, observed space, and agent space. That work was expanded in [10], which introduced enhanced dissimilarity measures by using extreme value theory, allowing for multiple sub-types of novelty, and this was performed in the agent space. [11], [37] introduced an information theoretic approach by using representation edit distance (RED) to measure the editing needed to represent skill programs in an agent's model effectively. This aimed at estimating the difficulty of learning and adapting to novelties. The approach relies on the algorithmic information theory (AIT) [8] and the minimal description length (MDL) [9] principle. Agents are built with a mental model consisting of representation and prediction portions, by which novelty is determined as a mismatch between an agent's mental model expectations and observations.

These frameworks focused heavily on defining novelty and its different characteristics, mostly on planning domains such as CartPole, Monopoly, and self-driving cars. Our work is significantly different but complementary to these works, as we proposed a quantifiable approach to understanding the perception domain characteristics used by learning models or agents in AI-based domains and is a component in defining the novelty characteristics.

In [6], authors have proposed a theory to measure the complexity between distinct domains in AI. The theory is evaluated using approximations by various neural network-based AI systems. The approximations are compared to well-known standards (entropy, cluster distribution, increased number of dataset classes, etc.), and the outcome shows it meets intuitions of complexity. Dataset complexity was estimated in [5], in which generative adversarial networks (GANs) are used to evaluate (un)interpretability of natural image distributions. Their approach considers methods to infer probability density estimates from GANs. The GAN-based algorithms are trained and tested on both MNIST and CIFAR datasets, to compute the probability densities. These results may help detect outliers, domain shifts, and novelties, but they have a high computational requirement. Classification difficulty was estimated in [7], where thirteen distinct datasets were considered, and three different strategies were employed, including Silhouette score, K-means, and a small neural network-based approach, *ProbeNets*. ProbeNets performs best, up to 27 times faster than training state-of-the-art deep neural networks. Although these methods attempted to estimate complexity in perception domains, they relied on classification labels and generalized datasets on classifiers in a singular environment. Our target is distinct in that we do not want to estimate classification difficulty, but estimate the intrinsic properties of the data and the training environment in a distributed setting.

In federated learning [13], [38], multiple entities possess a particular portion of training data, and the central server leads the training process by utilizing a model combining method, i.e., FedAvg [13], FedAdaGrad [38], FedYogi [38], etc. However, the existing training mechanisms must consider intrinsic property, i.e., data quality and distribution attributes, while training the federated model. Hence, the central server does not have any notion of domain complexity while choosing entities during training, which hinders the server from determining the optimal training algorithm or policy. Depending on the domain complexity, the training mechanism requires a considerable number of iterations (in iterative learning methods) to learn or utilize complex models or converge at lower task accuracy. Our proposal addresses the complexity of distributed learning irrespective of the dataset or environment type in the perception domain. Classical concepts related to data complexity, such as heterogeneity, sparsity, and dimensionality combined with distributed environment variables, are used to compute a single value corresponding to the difficulty level of the distributed environment.

III. Background and Preliminaries

A. *Perception domain*

Perceiving the complexity of the domains where an AI agent may navigate through is key for the development of AI systems really adaptable to novelties in the open world. There are a large number domain features to be considered toward robust solutions, highlighting the need of an in depth investigation into this area. The investigation in [3] has addressed this to some extent. Domain complexity comprises of both agent-dependent and agent-independent factors. Agent-dependent domain complexity components may change for different agents. On the other hand, agent-independent complexity focus on the inherent features of the environment itself, and does not change with different agents. The agent-independent complexity is named *intrinsic domain complexity* and the dependent components are called *extrinsic domain complexity*. A full domain complexity measure should consider both intrinsic and extrinsic components, as neither one of them can independently calculate the difficulty for adapting to novelties in open world.

Intrinsic domain complexity can be further divided into *environment space* and *task solution space*. The former encompasses all elements of the task environment, while the latter is concerned with only the elements relevant to accomplishing a given task inside that environment. Both complexities are further explained, as follows.

a) Environment space: The environment features may include objects, states, data scheme, parameter, variables, scale size, observations, or agents internal to the system. It is intuitive that the environment complexity increases with the number of the elements in each category and the distinct attributes and representations for each element, as defined in the open-world novelty hierarchy levels in [3]. In perception domain, the environment space accounts for all features, relationships and eventual phenomena resulting from either single entities or multiple entities.

b) Task solution space:: The task solution space is related to both the number and diversity of possible paths to complete a task. The task solution space grows in complexity as the set of allowed state transitions increases and as the possible paths for success get more complex. In perception domains, the task solution space also would include the set of data-classification classes [3]. The main causes of complexity in the task solution space are the number of possible paths, the set of possible agent interactions, and the restrictions on successful paths to reach a goal.

B. *Federated learning*

In federated learning, multiple distinct entities trains a model based on each ones local dataset. In each communication rounds a central server combines the model parameters and returns the global parameters to each entities for next phase of training. Federated learning objective function [13] as following,

$$\underset{\mathbf{w}\in R^d}{min} \; f(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} F_i(\mathbf{w}) \tag{1}$$

where N is the number of local entities and $F_i(w)$ is the local model parameter whereas $f(w)$ is the global parameter. The parameter in $F_i(w)$ are trained on local dataset possessed by each entity in the learning phase.

IV. Methodology

[3] defined three groups of measures in their proposed framework for computing the domain complexity, focused heavily on planning domains. Here we propose intrinsic complexity metrics to compute dimensionality, sparsity, and heterogeneity in the perception domain. Finally, we define a complexity metric for distributed perception domain combining the proposed intrinsic metrics and the distributed property of the learning environment.

A. *Intrinsic perception domain complexity*

1) Dimensionality: ***i) Environment complexity.*** For the general perception domain, the environment consists of the features in the dataset and the class or output labels (N_C). For a supervised setting, both are present, whereas, for an unsupervised setting, only the feature space is available. From an intuitive and classifier-independent point of view, we can estimate an upper bound for the feature space by considering two factors -

- Number of samples or size of the dataset (N_s).
- Number of features or the dimension of the dataset (N_d).

Any classifier, regardless of its architecture, would have to traverse this space in the worst case. So the environment complexity for any perception domain dataset would be,

$$EC_{upper} = N_s * N_d + N_C$$

We can further reduce the upper bound by considering the features that have zero variance over the data set. These features have little to no impact on the classifier's complexity or prediction rate. By variance, we try to see how a particular feature varies over its population. Features with the same value in all samples are said to have zero variance. So we can further reduce N_d after dropping zero variance variables all over the dataset. After reducing the feature space to N_d^r, the upper bound estimation for the environment complexity is:

$$EC_{upper} = N_s * N_d^r + N_C \tag{2}$$

ii) Intrinsic dimensionality. Intrinsic Dimensionality (ID) represents the minimal representation of the underlying manifold possible for a dataset [39] without losing any information. The manifold hypothesis refers to the fact that many high-dimensional data sets in the real world lie along low-dimensional latent manifolds inside that high-dimensional space [40]. So ID is the smallest dimension required for a deep learning model to exactly represent a dataset. Manifold representation is also sensitive to the non-linear structure of data and learns the high-dimensional structure of the data from the data itself without using predetermined classifications. This makes ID an ideal candidate to estimate a lower bound for the intrinsic metric for dimensionality. For example, in [41], they estimated the ID for the hand image data (real video sequence of a hand rotating along a 1-d curve) to be 3, referring to the different poses that are required to represent a hand image.

We used the maximum likelihood estimator of intrinsic dimensionality defined in [41]. Consider a dataset with n images and let X_i represent the individual sample (image) in the dataset. Let $X_1, \ldots, X_n \in \mathbb{R}^p$ be i.i.d. observations with an embedding of a lower dimensional sample. Let each sample X_i be mapped to a point x_i in the manifold. Specifically, $X_i = g(Y_i)$, where Y_i are sampled from an unknown smooth density f on $\mathbb{R}^p$, with some $m \leq p$, and $g(.)$ is a continuous and sufficiently smooth mapping function. Then m is the intrinsic dimensionality of the data set. Using the derivation from [41] and following the notations, the maximum likelihood estimation for m, given k nearest neighbors, is

$$m_k(x) = \left[\frac{1}{k-1}\sum_{j=1}^{k-1}\log\frac{T_k(x)}{T_j(x)}\right]^{-1} \tag{3}$$

where $T(x)$ is the average distance from each point to its k-th nearest neighbor. This estimator is shown to balance bias and variance better than all other existing solutions [41].

2) Sparsity: In image domain, for calculating the intrinsic sparsity of the single image, a common method [42] includes:

- create a low resolution (I^{LR}) image of a given high resolution (I^{HR}) version by downsampling it,
- calculate the absolute difference of the super-resolved version of I^{LR} (I^{SR}) and I^{HR}, $|I^{HR} - I^{SR}|$
- sparsity is the percentage of pixels in the $|I^{HR} - I^{SR}|$ with comparable performance.

In general, **sparsity**, or parsimony, is defined as the remaining components after a representation of some phenomenon with as few variables as possible [43]. For the perception domain, it translates into defining an environment with as few components as possible while retaining as much information as possible.

Using Principal Component Analysis (PCA) [44], it is possible to identify patterns in data on the basis of the correlation between features, which allows for reducing the dataset dimension. PCA achieves this by linearly transforming the data into a new coordinate system where (most of) the variation in the data can be described with fewer dimensions than the initial data. Isomap (Isometric Feature Mapping), unlike Principle Component Analysis, is a non-linear feature reduction method [45]. It is better than linear methods when dealing with almost all types of real image and motion tracking. Neither PCA nor Isomap relies on class labels and deals with intrinsic properties of data set, hence is a good indicator of intrinsic complexity.

3) Heterogeneity: In information theory [1], the entropy of a random variable is the average level of ”information”, ”surprise”, or ”uncertainty” inherent to the variable's possible outcomes.

[1] https://en.wikipedia.org/wiki/Entropy_(information_theory)

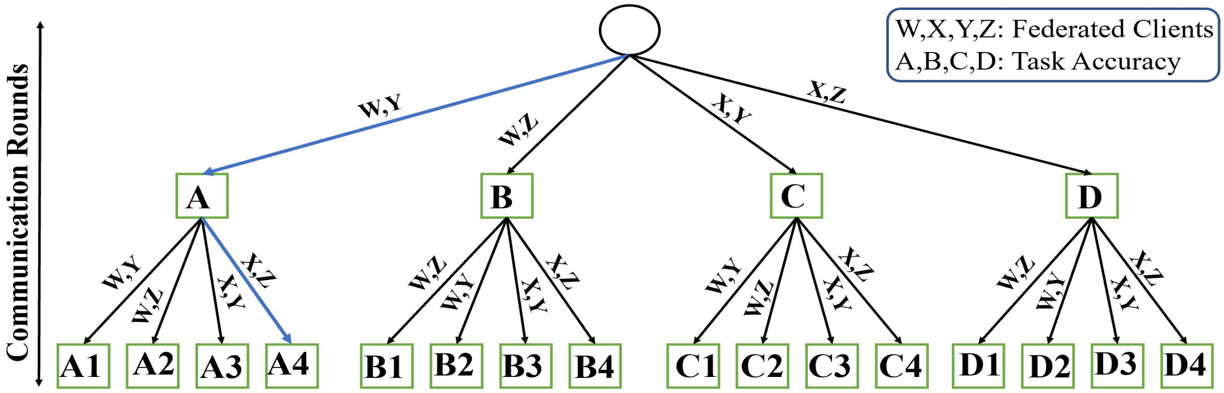

Fig. 1: Federated Learning Tree

Heterogeneity is defined as the diversity in the data set that contributes the most to the outcome. The **heterogeneity** of space can be measured by the Shannon entropy [46]. It gives a measurement of how diverse the data set is. Higher entropy translates into a more heteregeneous data set. The standard Shannon entropy of a grayscale image is defined as:

$$H = -\sum_{i=0}^{n-1} p_i \log p_i \tag{4}$$

where n is the number of gray levels and p_i is the probability of a pixel having a given gray level i.

B. Federated learning complexity

While the intrinsic complexity measure the difficulty level of a single, local domain, it does not account for additional complexities found in the open world, such as the one imposed by distributed environments. Because of that, we designed a framework based on the federated learning (FL) approach, which has been extensively used for training machine learning in a distributed way.

We consider FL training environment as a tree based solution, where we have distinct complexity for each non-identical paths in the tree. Figure 1 illustrates the FL tree, in which the height of the tree is the communication rounds (CR) between the federated master and the participating entities in the environment, and in this case CR is equal to two. In Figure 1 we can observe that there exists four entities in the environment which we denote as W, X, Y, Z, and the task accuracy is given by A, B, C, D and $A1, A2, \ldots, D4$. Data distribution of the entities is independent of the learning environment.

We assume that in our setting, at each communication round only two entities participate in the training. In practise, all entities are not available at each communication round for training. Here, by following each single path we achieve a distinct accuracy. For instance, in the first CR if W and Y participate in the learning, the accuracy achieved for the classification task is A. Likewise, if in the second round X and Z are engaged in the learning, the classification accuracy achieved is $A4$. This is true for each of the paths.

We define the complexity of a federated learning tree (G) as a composition of two distinct functions such as,

$$F(d, X) = f(X) + f(d) \tag{5}$$

where $f(X)$ perceives the intrinsic property of data and $f(d)$ penetrates the complexity of federated learning environment by the nature imposed by open-world settings. In this paper, federated environment complexity $f(d)$ is distinct from environment complexity in section 4.1.1. Environment complexity is defined for singular dataset whereas $f(d)$ is defined for distributed federated environment. We define the intrinsic property estimation function as follows,

$$f(X) = \beta \|X\|_2 = \beta \sqrt{x_1^2 + x_2^2 + ... + x_n^2} \tag{6}$$

where β is a normalizing hyper-parameter and n is the total number of actively participating entities in the federated learning environment. In open-world federated learning, data can be distributed to multiple distinct entities, but only certain available entities may become interested in taking active participation in the federated training. Here, x_i is the intrinsic property of the locally available data for participant i. We define the federated environment complexity as follows,

$$f(d) = (\frac{1}{m_1} + \frac{1}{m_2} + ... + \frac{1}{m_d}) \tag{7}$$

where d is the total number of distinct entities in the federated environment. In federated classification tasks, we consider each class of non-repeating entity as a distinct entity. For instance, in a federated binary classification task, if there exists two federated learning entities and each of the entities contain data samples from both of the classes, we consider $d = 1$. However, if each of the entities contains samples only from one class, then $d = 2$. Here, m_j is the frequency of the distinct entity j in the federated environment. The value of $f(d)$ can range from 0 to d, i.e., $0 < f(d) \leq d$. Hence, the complexity of a single path of the federated learning tree can be defined as,

$$\begin{split} F(d, X) = \beta(\sqrt{x_1^2 + x_2^2 + ... + x_n^2}) + (\frac{1}{m_1} + \\ \frac{1}{m_2} + ... + \frac{1}{m_d}) \end{split} \tag{8}$$

In federated learning, the federated master selects from multiple different paths in the learning tree based on the availability of the entities. Therefore, we define the learning tree complexity of the federated environment as the complexity of the perceived path (π). For instance, in figure 1 we can compute the complexity of the blue path using equation 8. This complexity can be the federated learning complexity, as long as the federated training follows this path. Here, π is a path in G, i.e., $\pi \in P(G)$. The federated master's goal is to estimate the complexity of each available paths and choose the one which can lessen the complexity in comparison to the complexity of the other paths while keeping the federated generalization accuracy as maximum as possible. Mathematically, with the assumption that all the federated generalization accuracy is equal, we can denote the objective of federated master as follows,

$$min\ F_{\pi}(d, X); \quad \forall \pi \in P(G) \tag{9}$$

The federated learning complexity will enable the federated server to decide on efficient training path as well as will provide a notion of learning task difficulty beforehand. Thus, the federated server can fine tune the training process by selecting effective path and learning model.

V. Experimental Setup and Evaluations

In this section we evaluate the efficacy of multiple domain complexity metrics in perception domain, i.e., (i) Shannon entropy as a metric of heterogeneity measurement, (ii) PCA-based sparsity measurement, (iii) environment complexity and intrinsic dimensionality, (iv) shallow convolutional neural network (ProbeNet) based complexity ranking, and (v) implicit data distribution based federated learning complexity.

We conduct the experiments using Python 3.8.10 in a windows machine of Intel core $i7$-8^{th} generation with 16 GB of memory. In the following we present and describe the experimental results with appropriate discretion.

A. Datasets

As the focus of our work is to evaluate perception domain, we chose to work with the classical MNIST dataset, as it is simple to start with and has some variations that are appropriate for evaluations of different complexity levels. Here we use three variants of this dataset: MNIST handwritten digit [47], Fashion-MNIST [48] and EMNIST-digits [49]. Both MNIST-handwritten and Fashion-MNIST contains $70,000$ gray-scale images, with $60,000$ training samples and $10,000$ testing dataset. EMNIST-digits contain $280,000$ characters. For all of them, the dimension of each image is 28×28 pixels, and the value of each pixel can be within 0-255.

B. Heterogeneity, sparsity, EC and ID

To show the inherent complexity of the three variants of MNIST, we have conducted experiments to measure the heterogeneity (entropy), the sparsity (number of sparse components for explaining variance, r^2 of 80% or 95%), environment complexity (with a variance threshold v_θ of 0 and 90), and intrinsic dimensionality for each variant. We use the Scikit-dimension [50] package for ID estimation, Scikit-image [51] for Shannon entropy estimation, Scikit-learn [52] for sparsity and EC estimation, and keras [53] for implementing the federated ProbeNet models.

The results are illustrated in Table-I, where we can observe that by the obtained values of sparsity, heterogeneity, environment complexity, and intrinsic dimensionality, Handwritten-MNIST is the least complex dataset and Fashion-MNIST is the most complex one, which is explained by their variation, and EMNIST-digits is between both. Though in cases, i.e., sparsity at $r^2 = 95\%$, the complexity order changes. This implies that Fashion-MNIST requires a lot of sparse components for explaining variances in between 80% and 95%. When we consider only the zero values as variance threshold, we can see Fashion-MNIST does not include many zeros over the data set, whereas if we consider pixel value 90 as threshold, it still has the largest environment complexity. We can infer the geometric complexity of our datasets from ID and we can see from MNIST to Fashion-MNIST the variance increases due to the increasing size and computational complexity. The nuanced increase also indicates that the datasets are all derived from MNIST.

C. ProbeNet as a complexity benchmark

We used a shallow ProbeNet [14] to measure the effectiveness of our proposed domain complexity metric. To determine the relationship between Probenet and the benchmarks on MNIST variants, we have evaluated the accuracy of Probenet over the three variants of MNIST, as a measure of complexity. We used a shallow convolution neural network as ProbeNet, and extrapolated the relationship between ProbeNet and the benchmark accuracy [54]–[56]. The results are depicted in Figure $2a$. There is a positive linear relationship between the two models. This is important since it allows us to use ProbeNet as a benchmark model to measure the accuracy of our proposed metric. Throughout the rest of the paper, we assume that the accuracy of ProbeNet ranks domain complexity in reverse order.

D. Federated Learning Complexity

In this section we assess the effectiveness of our proposed complexity metric, $F(d, X)$. We assume five distinct federated clients in the learning environment, where each client contains non-identical local data. Each client uses identical shallow ProbeNet model and the federated server relies on the FedAvg [13] algorithm for each global update. The accuracy results presented here reflects the accuracy of the federated ProbeNet on the test data. *Effort* represents the number of communication rounds in the federated learning context.

In these experiments we set the values of local iteration for each client to 1, and each client contains similar amount of data, allowing us to ignore possible data amount disparity. For each respective experiments, we run 100 communication rounds. In the following sections, we describe the impact of different components of our proposed complexity measurement, $F(d, X)$.

TABLE I: Heterogeneity, Sparsity, Environment Complexity, and Intrinsic Dimensionality Measurement

Dataset	Heterogeneity	Sparsity (r^2 = 80%)	Sparsity (r^2 = 95%)	$EC_{upper}(v_\theta = 0)$	$EC_{upper}(v_\theta = 90)$	ID
Handwritten-MNIST	1.60	740	629	717	530	13.368
EMNIST-digits	2.86	751	685	697	557	14.095
Fashion-MNIST	4.11	760	594	784	745	14.547

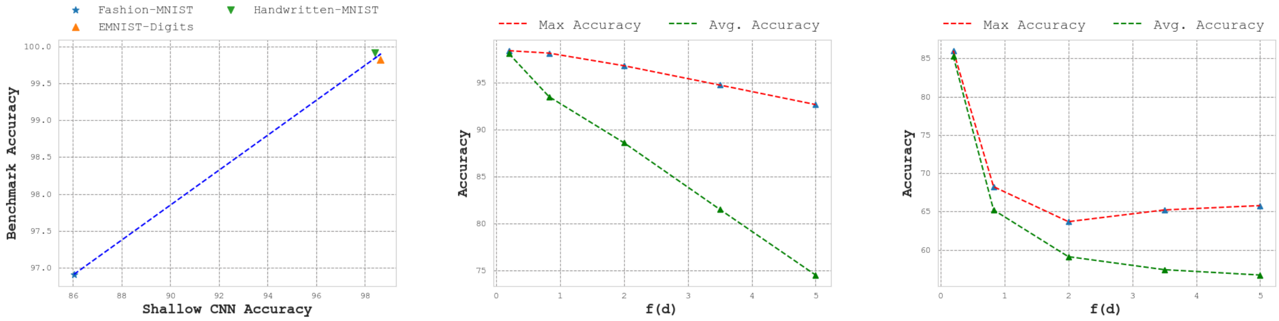

Fig. 2: (a) Shallow CNN accuracy vs. benchmark accuracy for Fashion-MNIST, Handwritten-MNIST and EMNIST-Digits, (b) Federated environment complexity $f(d)$ vs. shallow federated learning accuracy for Handwritten-MNIST when $f(X)$ is fixed, (c) Federated environment complexity $f(d)$ vs. shallow federated learning accuracy for Fashion-MNIST when $f(X)$ is fixed.

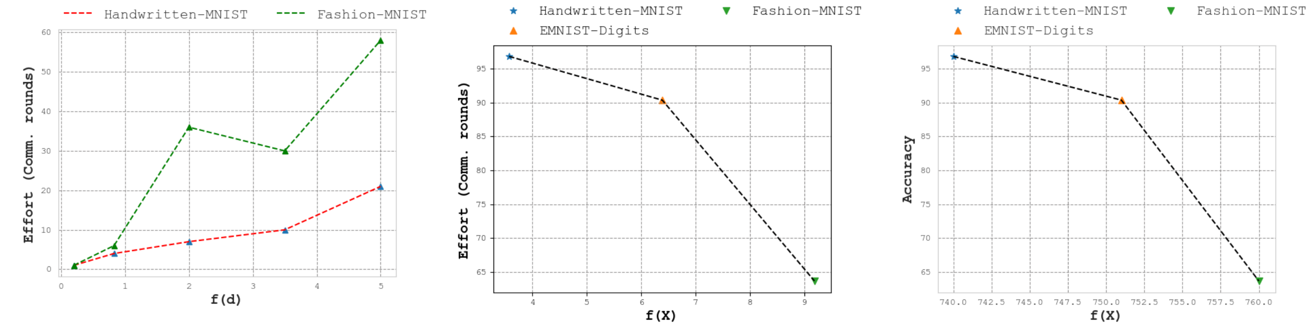

Fig. 3: (a) Federated environment complexity $f(d)$ vs. effort (communication rounds) for MNIST and Fashion-MNIST, (b) Federated accuracy vs. Federated intrinsic function $f(X)$, reflecting heterogeneity (entropy), for Fashion-MNIST, Handwritten-MNIST and EMNIST-Digits, with $f(d)$ fixed at 2, (c) Federated accuracy vs. Federated intrinsic function $f(X)$, reflecting sparsity (number of sparse components for explaining 80% of variance), for Fashion-MNIST, Handwritten-MNIST and EMNIST-Digits with $f(d)$ fixed at 2.

1) Federated learning complexity vs Federated environment complexity: To measure the impact of the federated environment complexity $f(d)$ on federated learning complexity $f(d, X)$, we experiment on five different values of d, ranging from 1 to 5. Figure $2b$ and $2c$ depict the relationship between maximum (or average) accuracy and $f(d)$ for Handwritten-MNIST (the easiest) and Fashion-MNIST (the hardest), respectively.

The maximum accuracy is computed here by taking the maximum test accuracy among all the communication rounds, and the average accuracy is the average test accuracy in the communication rounds. The results show us that as the value of $f(d)$ increases, the value of both accuracy decreases. This behavior is in line with the benchmark classifier.

We also evaluate the amount of effort needed by the federated learning complexity as a function of $f(d)$. The results are illustrated in Figure $3a$, where we observe that as $f(d)$ increases, the complexity of the federated learning increases too, while the intrinsic features (i.e., heterogeneity, sparsity, EC and ID) remains identical. In this case effort is defined as the earliest communication round in which a certain amount of federated generalization (or test) accuracy is achieved. We set this threshold to 60%. Since federated learning can achieve at least 60% accuracy in different distributed settings for all the three variants of MNIST.

2) Federated learning complexity vs Intrinsic complexity: Since the measured values for the intrinsic complexities in Section IV-A directly impact the computation of $f(X)$, we have evaluated the accuracy of the federated learning against $f(X)$, for all the variants of MNIST. While calculating $f(X)$ we set the value of β at 1 for entropy and for sparsity we set $\beta = \frac{1}{\sqrt{n}}$.

The results in figures. $3b$ and $3c$ show the generalization accuracy of the federated learning against $f(X)$. We can conclude that $f(X)$ has a inverse correlation with accuracy, which translates into positive correlation with complexity. Inversely, from figures $4a$ and $4b$, we can observe that with the increase of $f(X)$, the effort for threshold accuracy of 60% also increases. Overall, the complexity ranking of Fashion-MNIST, EMNIST-digits, and Handwritten-MNIST goes from high to low.

3) Federated complexity metric vs. Federated test accuracy: In order to evaluate the effectiveness of our proposed complexity metric, we compared the federated accuracy with the proposed metric $F(d, X)$. The results, considering maximum and average accuracy, are depicted in Figures. $4c$ and 5. We can observe that the average federated accuracy is higher correlation (R^2) with the federated complexity $F(d, X)$ than the maximum accuracy.

In these evaluations, R^2 value for maximum accuracy and $F(d, X)$ is 0.81, with standard error of approximately 7%, whereas R^2 value for average accuracy and $F(d, X)$ is 0.85 with standard error of approximately 6%. So, we can conclude that the federated complexity is indeed correlated with both the average and maximum accuracy, which implies that increase in $F(d, X)$ will decrease the accuracy of the federated learning. Thus, $F(d, X)$ can be considered as a standard metric for evaluating federated learning complexity.

4) Discussions on federated learning complexity: The evaluations confirm that the federated complexity function $F(d, X)$ encapsulates both the intrinsic properties of the data and the features of the federated environment, which makes it an ideal candidate to measure the rational complexity of the perception domain in the distributed environment.

For instance, in figures $2b$ and $2c$, the intrinsic properties of the domain is fixed, but the accuracy is changing with the change in federated environment complexity. On the other-hand, in figures $3b$ and $3c$ the accuracy changes with the intrinsic property change. In both cases, the federated environment variable is fixed. Though in the figures we only consider heterogeneity and sparsity as intrinsic properties, the accuracy trend also holds true for other intrinsic properties, i.e., environment complexity, EC, and intrinsic dimensionality, ID. We can observe that considering only one part of the $F(d, X)$ does not reflect the complete complexity of the perception domain in distributed environment. Based on this, it is reasonable to conclude that our proposed approach for measuring perception domain complexity in federated distributed environment is effective.

VI. Conclusion and Future Works

We have proposed a methodology to compute domain complexity of distributed datasets aiming at improving performance of AI systems in the open world. The proposed approach combines well-known complexity metrics such as heterogeneity, sparsity, environment complexity and intrinsic dimensionality with the Federated Learning framework as a robust distributed training technique. This is aimed towards a single metric to measure the complexity of distributed environments focused on perception domain. The performance evaluations were conducted on the classical MNIST dataset and its variants, and the outcome showed that our proposed metric achieves same complexity rankings perceived by the literature. The experiments show a correlation of 0.85 to *federated learning generalization accuracy* with the proposed domain complexity metric. We believe that our work can be very beneficial for future AI systems in dealing with uncertainties in the open world, in terms of quantifying the domain complexity for different environments. This is going to be very useful to determine agent performance bounds in any given environment, preventing wastage of computation effort for the agents. We are publishing our work with hopes of a widespread conversation within the open world AI community.

As future work, we will evaluate different strategies for the federated learning in our metric such as, variable distributed data size, distinct environments, autonomous driving datasets, etc. These additional steps will certainly render the proposed approach more general and robust.

References

[1] T. Shu, A. Bhandwaldar, C. Gan, K. Smith, S. Liu, D. Gutfreund, E. Spelke, J. Tenenbaum, and T. Ullman, "Agent: A benchmark for core psychological reasoning," in *Proceedings of the 38th International Conference on Machine Learning* (M. Meila and T. Zhang, eds.), vol. 139 of *Proceedings of Machine Learning Research*, pp. 9614–9625, PMLR, 18–24 Jul 2021.

[2] Š. Balogh, "Knowledge and datasets as a resource for improving artificial intelligence," in *Lecture Notes in Networks and Systems*, Lecture notes in networks and systems, pp. 828–837, Cham: Springer International Publishing, 2021.

[3] K. Doctor, C. Task, E. Kildebeck, M. Kejriwal, L. Holder, and R. Leong, "Toward defining a domain complexity measure across domains," in *In Proc. AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, March 2022.

[4] L. Claussmann, M. Revilloud, S. Glaser, and D. Gruyer, "A study on al-based approaches for high-level decision making in highway autonomous driving," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3671–3676, 2017.

[5] R. Krusinga, S. Shah, M. Zwicker, T. Goldstein, and D. W. Jacobs, "Understanding the (un)interpretability of natural image distributions using generative models," *ArXiv*, vol. abs/1901.01499, 2019.

[6] C. Pereyda and L. B. Holder, "Measuring the complexity of domains used to evaluate ai systems," *ArXiv*, vol. abs/2010.01985, 2020.

[7] F. Scheidegger, R. Istrate, G. Mariani, L. Benini, C. Bekas, and C. Malossi, "Efficient image dataset classification difficulty estimation for predicting deep-learning accuracy," *Vis. Comput.*, vol. 37, pp. 1593–1610, June 2021.

[8] P. D. Grünwald and P. M. B. Vitányi, "Algorithmic information theory," *CoRR*, vol. abs/0809.2754, 2008.

[9] P. D. Grunwald, *The minimum description length principle*. Adaptive Computation and Machine Learning Series, London, England: MIT Press, June 2019.

[10] T. E. Boult, N. M. Windesheim, S. Zhou, C. Pereyda, and L. B. Holder, "Weibull-open-world (wow) multi-type novelty detection in cartpole3d," *Algorithms*, vol. 15, no. 10, 2022.

[11] J. Alspector, "Representation edit distance as a measure of novelty," *CoRR*, vol. abs/2111.02770, 2021.

[12] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[13] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[14] F. Scheidegger, R. Istrate, G. Mariani, L. Benini, C. Bekas, and C. Malossi, "Efficient image dataset classification difficulty estimation for predicting deep-learning accuracy," *The Visual Computer*, vol. 37, 2021.

[15] S. W. Kim, Y. Zhou, J. Philion, A. Torralba, and S. Fidler, "Learning to simulate dynamic environments with gamegan," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1228–1237, 2020.

[16] R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak, "Planning to explore via self-supervised world models," 2020.

[17] M. Jafarzadeh, A. R. Dhamija, S. Cruz, C. Li, T. Ahmad, and T. E. Boult, "Open-world learning without labels," *ArXiv*, vol. abs/2011.12906, 2020.

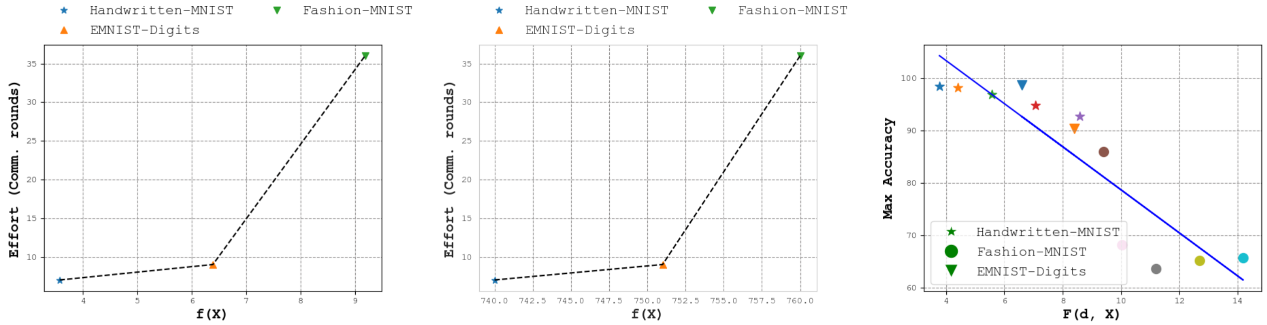

Fig. 4: (a) Federated intrinsic function $f(X)$ (reflecting heterogeneity) vs. effort (communication rounds), (b) Effort (communication rounds) vs. Federated intrinsic function $f(X)$, reflecting sparsity, for Fashion-MNIST, Handwritten-MNIST and EMNIST-Digits with $f(d)$ fixed at 2, (c) Federated complexity $F(d, X)$ vs. shallow Federated learning accuracy, considering maximum accuracy ($R^2 = 0.81$).

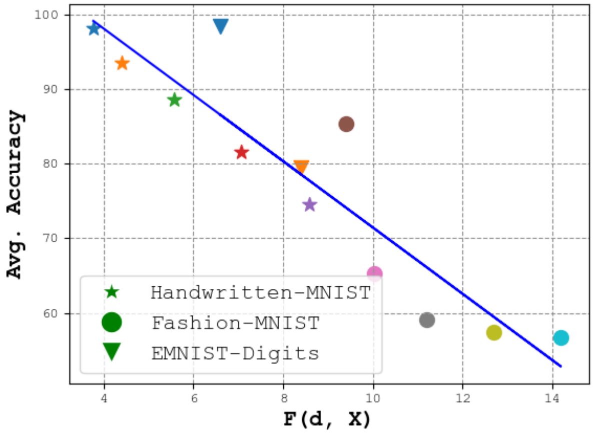

Fig. 5: Federated complexity $F(d, X)$ vs. shallow Federated learning accuracy, considering average accuracy ($R^2 = 0.85$).

[18] W. M. Piotrowski and S. Mohan, "Model-based novelty adaptation for open-world ai," 2020.

[19] T. E. Boult, P. A. Grabowicz, D. S. Prijatelj, R. Stern, L. B. Holder, J. Alspector, M. Jafarzadeh, T. Ahmad, A. R. Dhamija, C. Li, S. Cruz, A. Shrivastava, C. Vondrick, and W. J. Scheirer, "Towards a unifying framework for formal theories of novelty," in *AAAI*, 2021.

[20] T. Thai, M. Shen, N. Varshney, S. Gopalakrishnan, U. Soni, C. Baral, M. Scheutz, and J. Sinapov, "An architecture for novelty handling in a multi-agent stochastic environment: Case study in open-world monopoly," in *the 2022 AAAI Spring Symposium 'Designing Artificial Intelligence for Open Worlds*, 2022.

[21] K. Solaiman and B. Bhargava, "Open-learning framework for multi-modal information retrieval with weakly supervised joint embedding," in *Proceedings of the AAAI Spring Symposium on Designing Artificial Intelligence for Open Worlds, Palo Alto, CA, USA*, March 2022.

[22] A. Nesen, K. Solaiman, and B. Bhargava, "Dataset augmentation with generated novelties," in *2021 Third International Conference on Transdisciplinary AI (TransAI)*, pp. 41–44, IEEE, 2021.

[23] N. Tomasev, U. Paquet, D. Hassabis, and V. Kramnik, "Assessing game balance with alphazero: Exploring alternative rule sets in chess," *CoRR*, vol. abs/2009.04374, 2020.

[24] R. Domingues, P. Michiardi, J. Barlet, and M. Filippone, "A comparative evaluation of novelty detection algorithms for discrete sequences," *Artif. Intell. Rev.*, vol. 53, pp. 3787–3812, June 2020.

[25] X. Peng, J. C. Balloch, and M. O. Riedl, "Detecting and adapting to novelty in games," *CoRR*, vol. abs/2106.02204, 2021.

[26] M. Kejriwal and S. Thomas, "A multi-agent simulator for generating novelty in monopoly," *Simulation Modelling Practice and Theory*, vol. 112, p. 102364, 2021.

[27] P. Feeney, S. Schneider, P. Lymperopoulos, L. Liu, M. Scheutz, and M. C. Hughes, "NovelCraft: A dataset for novelty detection and discovery in open worlds," 2022.

[28] M. Batty, R. Morphet, P. Masucci, and K. Stanilov, "Entropy, complexity, and spatial information," *Journal of Geographical Systems*, vol. 16, pp. 363–385, Sept. 2014.

[29] E. Estevez-Rams, A. Mesa-Rodriguez, and D. Estevez-Moya, "Complexity-entropy analysis at different levels of organisation in written language," *PLOS ONE*, vol. 14, p. e0214863, May 2019.

[30] G. Dohnal and I. Bukovský, "Novelty detection based on learning entropy," *Appl. Stoch. Models Bus. Ind.*, vol. 36, pp. 178–183, Jan. 2020.

[31] J. Vrba and J. Mareš, "Introduction to extreme seeking entropy," *Entropy (Basel)*, vol. 22, p. 93, Jan. 2020.

[32] J. Wurst, A. F. Fernández, M. Botsch, and W. Utschick, "An entropy based outlier score and its application to novelty detection for road infrastructure images," *CoRR*, vol. abs/2005.13288, 2020.

[33] P. Langley, "Open-world learning for radically autonomous agents," *Proc. Conf. AAAI Artif. Intell.*, vol. 34, pp. 13539–13543, Apr. 2020.

[34] M. Tian, D. Guo, Y. Cui, X. Pan, and S. Chen, "Improving auto-encoder novelty detection using channel attention and entropy minimization," in *Proceedings of the 2nd ACM International Conference on Multimedia in Asia*, (New York, NY, USA), ACM, Mar. 2021.

[35] R. P. Cardoso, E. Hart, D. B. Kurka, and J. V. Pitt, "Using novelty search to explicitly create diversity in ensembles of classifiers," in *Proceedings of the Genetic and Evolutionary Computation Conference*, (New York, NY, USA), ACM, June 2021.

[36] P. Juszczuk, J. Kozak, G. Dziczkowski, S. Głowania, T. Jach, and B. Probierz, "Real-world data difficulty estimation with the use of entropy," *Entropy (Basel)*, vol. 23, p. 1621, Dec. 2021.

[37] K. Solaiman and B. Bhargava, "Measurement of novelty difficulty in monopoly," in *Proceedings of the AAAI Spring Symposium on Designing Artificial Intelligence for Open Worlds, Palo Alto, CA, USA*, March 2022.

[38] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," *arXiv preprint arXiv:2003.00295*, 2020.

[39] A. A. Rahane and A. Subramanian, "Measures of complexity for large scale image datasets," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, IEEE, feb 2020.

[40] C. Fefferman, S. Mitter, and H. Narayanan, "Testing the manifold hypothesis," *Journal of the American Mathematical Society*, vol. 29, no. 4, pp. 983–1049, 2016.

[41] E. Levina and P. Bickel, "Maximum likelihood estimation of intrinsic dimension," *Advances in neural information processing systems*, vol. 17, 2004.

[42] L. Wang, X. Dong, Y. Wang, X. Ying, Z. Lin, W. An, and Y. Guo, "Exploring sparsity in image super-resolution for efficient inference," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4917–4926, 2021.

[43] J. Mairal, F. Bach, and J. Ponce, "Sparse modeling for image and vision processing," 2014.

[44] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.

[45] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[46] C. E. Shannon, “A mathematical theory of communication,” *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.

[47] L. Deng, “The mnist database of handwritten digit images for machine learning research,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[48] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” 2017. cite arxiv:1708.07747Comment: Dataset is freely available at https://github.com/zalandoresearch/fashion-mnist Benchmark is available at http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/.

[49] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, “Emnist: an extension of mnist to handwritten letters,” 2017.

[50] J. Bac, E. M. Mirkes, A. N. Gorban, I. Tyukin, and A. Zinovyev, “Scikit-dimension: a python package for intrinsic dimension estimation,” *Entropy*, vol. 23, no. 10, p. 1368, 2021.

[51] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, “scikit-image: Image processing in python,” *Peerj*, 2014.

[52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[53] F. Chollet *et al.*, “Keras,” 2015.

[54] S. An, M. Lee, S. Park, H. Yang, and J. So, “An ensemble of simple convolutional neural network models for mnist digit recognition,” *arXiv preprint arXiv:2008.10400*, 2020.

[55] M. S. Tanveer, M. U. K. Khan, and C.-M. Kyung, “Fine-tuning darts for image classification,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 4789–4796, IEEE, 2021.

[56] K. Viswanathan, A. Sethi, *et al.*, “Wavemix-lite: A resource-efficient neural network for image analysis,”