Robust Predictive Models for Addressing Aggregate-Level Uncertainties within Data

Anonymous Authors

ABSTRACT

Though deep learning has fundamentally changed how we perform many tasks, its applications are somehow limited by the amount of data needed for effective task-specific generalization. For example, consider the case of a disease classification model or drug recommender system where observations for training may only be available at a cohort-level (rather than at individual levels) for reasons attributed to the preservation of privacy. Meanwhile, it is important to make predictions on particular instances - the ultimate goal is to correctly classify the symptoms of a *particular* individual. In this paper, we conduct an extensive study of this relatively under-studied problem of training on aggregates and inference on individual data. We present a formal description of this problem with a novel method that seeks to robustly learn from aggregated information during the training phase which we name Individual Predictions on Aggregated Training (IPAT). Improving generalization from aggregate data has implications in low information applications of machine learning such as modelling of medical random control trials and other population sample statistics. We hypothesize that learning from different parameterized representations corresponding to different perturbations of aggregated data better generalizes to inference on true individual instances. We propose a new architecture utilising probabilistic methods to regularize learning from aggregated data using perturbation at both the input and latent stage of modelling. Using simulated aggregations over partitions of instances via K-means clustering from several standard benchmarks from different domains and data types, we observe substantial improvements over baseline approaches (> Δ60% Accuracy).

CCS CONCEPTS

• **Computing methodologies** → *Neural networks*; **Machine learning approaches**.

KEYWORDS

Machine Learning, Instance Cohorts, Stochastic Sampling

ACM Reference Format:
Anonymous Authors. 2022. Robust Predictive Models for Addressing Aggregate-Level Uncertainties within Data. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Within the field of machine learning we observe trends in the increasing size of both models and training datasets, the former dictating a need for the latter. This is because deep neural models require a large number of samples from a population to capture non-linear relationships in data. In benchmark datasets used in research, clean examples are often abundant. However in real applications such as medical classification or population modelling where an individual's privacy can be infringed upon, we observe that data is often released as an aggregate statistic [2; 23]. In some cases it may not be feasible to present individual instances, for example in a country's census taking, and in others aggregation is performed out of an abundance of caution against membership attacks or other forms of deanonymization [7; 21; 33]. This being the identification of subjects data used in training neural models. A clear example of this is random control trials (RCTs) [2; 12; 23] where medical insights are presented as aggregations over groups of individuals. Many key insights regarding COVID-19 were derived from RCTs however the prevalence of significant side effects in some patients was lost due to the sole use of aggregate statistics [31]. By better generalizing from the information in RCTs we can improve the robustness of medical insights derived from these studies.

Due to the requirement of a high number of training examples to generalize to many downstream tasks, small datasets particularly those composed of aggregated examples may not lead to satisfactory outcomes when used to train a deep learning model. We propose that by improving the ability of a model to generalize to a task when trained on an aggregated dataset with a supervised learning approach could potentially have significant implications in any privacy conserving application of machine learning. For instance, in the context of RCTs, developing precision classification models trained on aggregate data mined from these trials. Another application could be the use of macroscopic trends derived from models train on data mined from censuses or other large scale surveys in aggregated form in general product recommendation, a large benefit being the privacy preserving nature of this process relative to accessing 3rd party data on particular individuals. This approach could potentially also provide robust generalization under realistic conditions where examples are limited and noisy such as census data or human trials. This problem is quite unique in the sense that the form of the training data and test data is different causing high epistemic error in a supervised training setting. We investigate methods of reducing the effect this problem, specifically we learn from aggregates by sampling pseudo-instances at both input and latent stages.

Within both traditional and neural modelling approaches we encounter uncertainty from multiple sources. Such uncertainties are broadly grouped into epistemic and aleatoric uncertainties [13; 16]. Epistemic uncertainty is caused by a lack of knowledge, for example if a model does not account for a variable in its prediction or has not observed enough samples from a given population. In deep

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

learning we can consider epistemic uncertainty as the uncertainty in the parameters of a model [18; 36]. Aleatoric uncertainty refers to the stochasticity of an input, in other words random noise that is not explained by uncertainty in the model itself and is inherent to the training data due to imperfect data collection or the inherent stochasticity of the observed entities. The noise captured within training data affects the posterior distribution over classes in a supervised task, which means that instances observed during inference may be out of distribution (OOD) for the learned decision boundaries of a model. We use aspects of variational inference when we taking realizations from latent logits as well as using regularization techniques. However, the problem investigated in this paper differs in the sense that we are training on data that sparsely represents the population distribution and must perform inference on individual inputs which will be of a different form to the training data.

Our Contributions. We propose a novel problem definition, Individual Predictions on Aggregated Training (IPAT) formally defined in Section 3.1. Within this space we consider relevant applications, an appropriate method to simulate aggregate level uncertainty and how to improve inference when training on aggregations of instance cohorts through the generation of pseudo-instances. We propose methods of simulating aggregate data across two modalities due to the lack of true evaluation benchmarks in literature allowing us to test our hypothesis regarding the ability to learn from aggregates through perturbation. Our generalization method is not specific to either the architecture of a neural model or the input modality of the data. We shall release artifacts, our codebase and evaluation suite upon acceptance of this work. A summary of our contributions is as follows.

- A formal definition of the IPAT problem.
- Development of synthetic aggregate data for IPAT evaluation.
- Novel architectures for IPAT training and inference.

2 RELATED WORK

Learning from Aggregate Data. Learning from aggregate is a somewhat understudied problem, the current literature is centred around the recovery of parameters of a probabilistic model, in other words training from aggregates to fit a model which most resembles a model trained on individual instances. Bhowmik et. al. [6] first proposed the recovery of a parameterized model from aggregate training data proposing upper bounds on the probability that parameters can be recovered from some aggregate data. Effectively they show that under different degrees of aggregation model parameters can still be recovered in a linear model. Gilotte et. al. [11] then proposed the use of a Markov Random Field to estimate gradient uncertainty during training. By using Gibbs sampling to estimate gradients, the optimization process is regularized such that parameters can be more effectively recovered. In our work the latent representation of a neural model is regularized as opposed to the optimization step itself. Though this work is partially motivated by these prior works, we expand our evaluation to a classification setting as opposed to optimizing for parameter recovery. By using neural methods, a plethora of downstream tasks become feasible on higher dimensional data that classical methods of machine learning are not capable of such as direct use of embeddings, segmentation and classification of complex instances.

Multiple Instance Learning. Our work in this paper relates to Multiple Instance Learning (MIL) which seeks to learn from weakly labelled data. MIL generally involves a binary classification model being exposed to a set of instances with a ground truth label which indicates the presence at least one member of a given class within the set [3]. MIL is useful in applications with weakly labelled training sets where confidence in the quality of a given dataset is low, weakly labelled data contains some indication of instance classes but is not explicitly human labelled, examples include unsupervised approaches to attach pseudo-labels to data instances or the use of a single label over a group of instances inferred by some majority decision. In our case we have a low-information training set in which aggregates are determined by class groupings giving a pseudo label over a group of training samples.

The concept within MIL that we seek to exploit is that an estimation of classification labels for individual instances can be learned from cohorts of instances, concretely speaking, our work differs in the form of training data, MIL exposes a model to multiple instances in a group under a single label, in our task we aggregate these groups and present the model with a label with absolute confidence.

Out-of-Distribution Generalization and Stochasticity. The problem with aggregated training can be considered a case of training with high epistemic error caused by a lack of training samples and the potential for high aleatoric error due to the corruption of distinct features by the aggregation process. The former problem is similar to problems in Out-Of-Distribution (OOD) generalization in that we compensate for realizations not captured by training data. Much of this research relies on sample generation during training using some conditional generator, which itself relies on variational inference that we also utilise. Nevertheless after sampling is performed, the classification network is a standard feed-forward network that does not use probabilistic methods to estimate a posterior.

Bai et. al [4] utilise the marginal likelihood of a synthetic dataset given a candidate set of model parameters to optimize parameters over a general task. Kaur et. al. [15] explicitly include priors over their random perturbations of image datasets such as conditional probabilities of a class given an image and its rotation. In terms of explicit regularization, Yi et. al. [34] add a penalty that is computed via penalizing the dependence of the models predictions on spurious features by measuring the variation in prediction for an input conditioned on multiple features.

Related to OOD generalization is the field of quantification first defined by Forman [10]. Quantification differs from classification in that you want to make predictions that align with some true population distribution as opposed to necessarily accurate individual predictions. Quantification has been investigated across multiple modalities including image and text data [5; 22]. This can be seen as the inverse of our work in that quantification fits to some population level features learning from individual instances whereas in this work IPAT fits to some individual instances from population level features.

3 PROPOSED METHODOLOGY

3.1 Formal Description of the Problem

We now formally define the task of training with aggregate-level information and predicting on individual instances. We use the name **IPAT (Individual Predictions on Aggregated Training)** as an abbreviation for this problem and from hereon continue to use this abbreviation to refer to the task.

3.1.1 *Standard Classification Task.* A standard supervised learning approach (e.g. classification) estimates a functional dependence of the form $\theta : X \mapsto Y$ between a set, $X = \{\mathbf{x} : \mathbf{x} \in \mathbb{R}^d\}$, of data instances (strictly speaking, a d-dimensional encoding of data instances), and a set, $Y = \{y : y \in \mathbb{Z}_c\}$ of c categorical value labels. This functional dependence is estimated from *training examples*, i.e. samples of pairs from the set $X \times Y$. With a set comprising M such pairs, $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{M}$, a parameterized classification model then estimates this function as a map from the training pairs to a c-class probability distribution function, $\hat{\theta} : \{(\mathbf{x}_i, \mathrm{y}_i)\}_{i=1}^{M} \mapsto \Delta_{c-1}$, where the notation Δ_{c-1} denotes a $(c-1)$-simplex.

The components of the parameter vector θ are usually learned via gradient descent based approaches. Obviously, the error in this parametric approximation $\hat{\theta}$, e.g., as learned via gradient descent, is expected to be decreasing with an increasing number of pairs $(\mathbf{x}_i, y_i)$ in the training set, M, i.e., $\hat{\theta} \to \theta$ as $M \to \infty$.

3.1.2 *IPAT.* Now to formally introduce the IPAT problem, let us denote a set of instances along with their class labels as the set $\mathcal{S} = \{(\mathbf{x}, \mathrm{y}) : \mathbf{x} \in \mathbb{R}^d,\ \mathrm{y} \in \mathbb{Z}^c\}$. An important difference from the standard supervised setup is that the set S is constituted of two different subsets - the training set $\mathcal{S}_{\text{train}}$ and the test set $\mathcal{S}_{\text{test}}$ with two different characteristics, which we explain next.

In contrast to standard supervised learning, in IPAT the training data $\mathcal{S}_{\text{train}}$ is not available in the form of a set of individual instances of the form $\{(\mathbf{x}, \mathrm{y})\}$. Instead, it is useful to think that the training set $\mathcal{S}_{\text{train}}$ is produced as a result of a transformation function that maps groups corresponding to the situation of learning a model of aggregated behaviour from RCTs, comprised of a variable number of individual instances into a single instance, thus resulting in a single collapsed or aggregated instance per group. More formally, the set $\mathcal{S}_{\text{train}}$ of individual instances in IPAT is a union of m mutually disjoint subsets $\mathcal{T}_1, \ldots, \mathcal{T}_m$, each $\mathcal{T}_i$ corresponding to a group or cohort of size n_i, i.e., $|\mathcal{T}_i| = n_i$ and $\sum_{i=1}^{m} n_i = |\mathcal{T}|$.

Each partition $\mathcal{T}_i$ produces an aggregated or collapsed instance $\mathbf{z}_i$, i.e., $\mathbf{z}_i = \phi_x(\mathbf{x}_1, \ldots, \mathbf{x}_{n_i})$ where ϕ_x denotes an *collapsing* function that transforms the n_i individual level instances into a single collapsed instance $\mathbf{z}_i$.

Similarly, the class label, i.e., the value of y associated with each aggregated instance $\mathbf{z}_i$ is also an aggregation function that maps the individual class labels associated with the instances $x_1, ..., x_i$ into one single label. Formally, $\mathrm{y} = \phi_y(\mathrm{y}_1, \ldots, \mathrm{y}_{n_i})$, where ϕ_y denotes a collapsing function of the labels.

3.2 General IPAT Training

The training phase in supervised learning has access only to aggregated instances, i.e., $\mathcal{S}_{\text{train}} = \{\mathbf{z}_i\}_{i=1}^{m}$, where each aggregated instance corresponds to the i^{th} group determined by some class label or other discrete indicator such as sex or age in human trials. Similar to the standard setup of a supervised learning, the objective is to learn a decision boundary of the form $\hat{\theta} : \{(\mathrm{z}_i, \mathrm{y})\}_{i=1}^{m} \mapsto \Delta_{c-1}$.

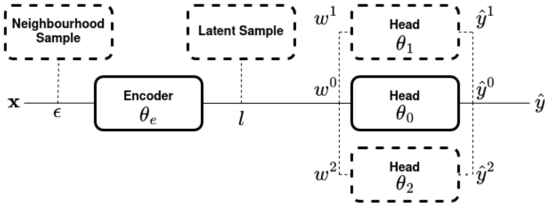


Figure 1: A schematic showing general components of our architecture, the essential components, the encoder and at least one classification head are connected with solid lines. Optional components are shown and connected with a dotted line, these include p ensemble classification heads (3 are shown for illustrative purposes), neighbourhood sampling (discussed in Section 3.3.1) and latent sampling (discussed in Section 3.3.2).

The main challenge during the training phase is to prepare the model for what it may encounter during the inference phase. Simply training on the aggregated instances poses two major concerns - first, the aggregated instances do not reflect the true properties of the data on which inference is to be carried out, and second, the number of aggregated instances is smaller than the total number of individual instances (as there is only aggregated instance each group), which means that the model requires to be trained with a much smaller amount of data, which is also challenging.

During the inference phase, the trained model $\hat{\theta}$ is used to predict the class labels of each $\mathbf{x} \in \mathcal{S}_{\text{test}}$. In contrast to the aggregated instances $\mathbf{z}_i$'s, each $\mathbf{x}$ is a non-aggregated (atomic) instance. This means that what the model has *seen* during training is characteristically different from what it *sees* during the inference.

3.3 Perturbation Methods

We propose two approaches to increase the number of samples a model observes during IPAT training. We propose to approximately reconstruct the true data instances and their associated labels that were aggregated to yield a collapsed data instance $\mathbf{z}$ and its associated label $\mathbf{y}$. Specifically we propose the application of noise at different stages of training to improve model generalization by relaxing its decision boundaries, ϵ-neighbourhood noise added to input representations of z to generate pseudo-instances and Gumbel noise added to the latent representation of each z perturbing latent logits before classification.

3.3.1 *ϵ-Reconstruction.* We investigate noise as a hyperparameter ϵ of the model controlling the range of noise added to input representations. We choose to define a parameter ϵ following ϵ-robustness in which a model should classify all instances within a given neighbourhood with the same class label [19; 37], this is a form of regularization. Instead of ensuring a model is ϵ-robust, we explicitly sample from this neighbourhood and assign each neighbour the label of the original instance.

When using neighbourhood sampling, we sample p instances, each denoted as $\mathbf{w}^{(i)}$, from the ϵ-neighborhood of $\mathbf{w} = \theta_n(\mathbf{z})$. More formally speaking, $\mathbf{w}^{(i)} \sim \mathcal{N}_\epsilon(\mathbf{z})$, where the notation $\mathcal{N}_\epsilon(\mathbf{z})$

denotes the ϵ neighborhood around the point $\mathbf{z}$ as measured by the l_∞ measure, i.e., a point $\mathbf{u} \in \mathcal{N}_\epsilon(\mathbf{z})$ if $||\mathbf{z} - \mathbf{u}||_\infty < \epsilon$. While we seek to reconstruct the individual data instances with the help of the stochastic step of ϵ-neighborhood sampling, it is more problematic to obtain the ground-truth class labels for each sampled instance $\mathbf{w}^{(i)}$. The majority label that was used in the aggregation is assigned to all the reconstructed instances. A task-specific network to extract features denoted θ_e then produces a vector for each sample that is classified by the ensemble of classifiers.

This sampling can be performed as an offline task by generating p samples per aggregate or at training time where different neighbourhood samples are realized in each batch.

3.3.2 Latent Perturbation. Similar to generative models such as generative adversarial networks or variational autoencoders, we investigate the perturbation of latent representations as a form of reconstruction. Reconstruction or in other words generation preferably uses a large number of unique training samples to learn a latent space which can be sampled from and reconstructed into a new instance. With aggregate datasets few instances exist [2; 23] such that common generative methods are inappropriate. As such, we approximately reconstruct through the addition of noise.

We investigate perturbation of latent categorical representations utilising the Gumbel-Softmax trick [?] to approximately sample from latent categorical distributions [14]. We use this latent form as it has been shown to outperform a standard latent representation in a variational autoencoder [14], though we do not reconstruct an instance from said latent sample.

To add noise to latent representations of samples we first use a task specific encoder with parameters θ_e to produce l, being a latent distribution over z which in this case is a set of categorical distributions. Here we sample p pseudo-representations $\mathbf{w} = \{\mathbf{w}_0, ..., \mathbf{w}_p\}$ where each $\mathbf{w}_i$ is a realization of l which can then be classified.

When using an explicitly stochastic component in a neural network, one must ensure that the component function is smooth or differentiable almost everywhere as this is required for back propagation to optimise the parameters of a neural network. In our use case, we sample at multiple points in the network architecture and as such make changes as necessary. When adding Gaussian noise to inputs no changes are necessary as the noise component occurs before any transformation by the network and is trivially reparameterised as $\mu + \sigma(\mathcal{N}(0, 1))$ where $\mathcal{N}(0, 1)$ is a single Gaussian random variable with mean 0 and variance 1.

When adding noise to the latent representation of an input, we must use a smooth sample function. To this end, as a first step we take an aggregated instance $\mathbf{z}$ and encode it to some latent representation $l \in \mathbb{R}^{d \times c}$ where d is the number of latent variables and c is the number of categories of a categorical distribution. Now, using the Gumbel-Softmax trick we approximately sample from the categorical distribution $\tilde{l}$ yielding p latent reconstructions $\{w_1, ..., w_p\}$. The Gumbel-Softmax distribution is re-parameterised as:

$$l_i = \text{softmax}((G + \log \alpha)/\tau), \tag{1}$$

Where G is Gumbel noise computed as $-\log(-\log(\mathcal{N}(0, 1)))$, $\log \alpha$ is the logits generated by the task-specific encoder structure and τ is the hyperparameter temperature [14].

This trick is an inverse transform, being a realization of a distribution via the inverse of its cumulative distribution function which is arranged in a form that is smooth. Therefore, the sample function can be differentiated unlike a true realization from a categorical distribution. This temperature over the logits affects how hard the distribution is, in other words as $\tau \rightarrow 0$, the distribution over $\tilde{l}$ which at high temperatures acts as a Softmax, becomes Argmax. In reality, this leads to a gradual increase in gradient variance as temperature decreases due to harder Argmax like samples.

To model the latent representation we minimize the Kullbach-Lieber divergence between the latent representation and a fixed uniform prior, this acts to regularize the latent representation when sampling from categorical logits. We are using a discrete distribution in the form of the relaxed categorical or Gumbel-Softmax distribution, the true KL divergence is poorly defined between discrete and continuous distributions [27]. As such, we use a Monte Carlo approximation of the relaxed KL divergence:

$$\mathbb{KL}(\tilde{\alpha}||p_0) = \mathbb{E}_{\tilde{\alpha}_\phi(w)}[\log \frac{\tilde{\alpha}_\phi(\tilde{w})}{p_0(\tilde{w})}] \text{ where } \tilde{\alpha} = \frac{\alpha_k}{\sum_{i=1}^{k} \alpha_i}. \tag{2}$$

In equation 2, $\tilde{\alpha}$ is a Monte Carlo approximation of q_0 being the latent distribution l, effectively a sum over each variable of our latent representation, which represents a continuous distribution we compare to p_0, being the distribution over the latent logits we want to approximate. The analytical computation of this approximation is the sum over the entropy of the logits minus the cross entropy of the logits with respect to a uniform categorical distribution. In our application as we have multiple latent categorical distributions, we take the mean of this expression. This computation is proportional to the logits at small values of τ [27] such that we can regularize the latent representation by minimizing this expression which henceforth we refer to as $\mathcal{L}_{\text{latent}}$.

With the advent of neural diffusion models [32] we feel it is appropriate to justify our approach with respect to generative diffusion models. When using diffusion, over a period of time-steps an input is perturbed with Gaussian noise and then the steps of this process are decoded to a new instance. This process requires significant computation expense and training data [28; 29], which is not available when training on aggregates. As such we propose that a VAE style encoder is more appropriate for this task.

3.4 Classification

Each $\mathbf{w}^{(i)}$, the i^{th} perturbation of an aggregate instance is transformed by a set of parameters θ_i $(i = 1, \ldots, p)$ being the parameters of the i^{th} classifier in an ensemble architecture, the simplest case being a sequential classifier. The output from each ensemble member is a c-class Softmax probability $\hat{y}^{(i)} = \theta_i(\mathbf{w}^{(i)})$. We parameterise the number of samples drawn from each aggregate instance neighbourhood by p and tie this value to the number of classifiers in the ensemble so that each classifier in ensemble observes different perturbations of an aggregate instances.

Informally speaking, our proposed method takes a cohort-level data instance, seeks to reconstruct the individual-level data instances by the sampling step, and then using a task specific classification method, learns a function θ_i from each individual classifier to map from a reconstructed input $\mathbf{w}^{(i)}$ to the class label $y = y_1 = \ldots = y_p$.

3.5 Architectures

Given these perturbation methods, the first being the addition of noise to the input representation before encoding, and the second being perturbation of the encoded representation, we propose the following architectures.

3.5.1 Ensemble Structure. To encourage generalization we propose the use of an ensemble of classifiers, this method has been shown in prior work to improve generalization [8; 20; 30]. Specifically we ensure that models do not just learn the noise applied to input or latent stages. Ensemble members are referred to in the explanation below though sequential classifiers can be seen as a reduced or base case of our architecture which we also investigate.

Our general structure is composed of a single encoder parameterized by θ_e and p classification heads $\{\theta_0, ..., \theta_p\}$. Each classification head observes a different perturbation of the input in both cases of perturbation.

For each classification head θ_i, we compute the cross entropy of the prediction of a pseudo-instance and ground truth of the aggregate instance.

$$\mathcal{L}_{\text{recons}} = -\sum_{j=1}^{n} P(y = j) log(P_{\theta_i}(y = j|\theta_e(z))) \tag{3}$$

Where in equation 3, n is the number of classes, $P(y = j)$ is the ground truth value for class j and $P_{\theta_i}(y = j|\theta_e(x))$ is the confidence level of the ith classification head in classifying the encoded aggregate instance with label j. We sum these losses to ensure the back-propagation of error through each classification head. At inference time these predictions are aggregated to return a single categorical prediction over the classes while utilising the regularizing effect of the ensemble architecture.

$$f(x; \hat{\theta}) = \frac{1}{p} \sum_{i=1}^{p} P_{\theta_i}(y|\theta_e(z)) \tag{4}$$

3.5.2 Input Reconstruction. We investigate two cases of input reconstruction, both expose models to neighbourhood-samples pseudo instances however they differ in how often the samples are realized. We first investigate reconstruction pre-training, in other words the dataset is reconstructed to Kp instances determined by K the number of total aggregate instances and p the number of ensemble members. This effectively means that the model is exposed to the same pseudo-instances in each epoch. The second method we investigate is in-training reconstruction where Kp instances are realized at the start of each epoch. The potential benefit of pre-training reconstruction is model stability as no explicitly probabilistic components are used within the model such that the error created by continuously sampling at each epoch cannot be explicitly compensated for. When using input reconstruction the loss function is defined in Equation 3 solely using $\mathcal{L}_{\text{recons}}$.

3.5.3 Latent Reconstruction. Latent reconstruction is performed in-training by the nature of this method. Realizing samples w from the latent parameterized distribution l, each epoch we expose the classifier heads to different latent representations of the original aggregate z. The hypothesized benefit of this approach is that the encoder structure can learn a strong representation of the mapping $z \longrightarrow l$ whilst the classifiers are exposed to diverse latent embeddings improving generalization at the inference stage. Each ensemble member observes a different realization from l. The loss function for latent reconstruction is computed as $\mathcal{L} = \mathcal{L}_{\text{recons}} + \mathcal{L}_{\text{latent}}$.

3.5.4 Modality Specific Components. In each case the classification head is a dense neural network that maps a latent representation w to some $y \in \mathbb{Z}^c$. The encoder is modality specific such that inductive bias can be exploited to best represent a given input, for example the use of convolutions over an image representation.

4 EXPERIMENT SETUP

We now describe the detailed experiment setup, the methods investigated and the results obtained for IPAT evaluation. To demonstrate that our proposed method is useful for effective IPAT evaluation, i.e., it is able to train effectively well with noisy data instances aggregated over cohorts but still perform adequately on the individual-level test data, we employ some simple baseline approaches.

4.1 Research Questions

We first look to address inference performance when a model is exposed to aggregate instances during training, we look to verify the negative effect of training on a small number of low information examples. Many human studies cannot release individual information and as such there is significant utility in learning from aggregates. To this end we will generate synthetic data with varying degrees of aggregation across dense representations of both image and tabular data, this method allows for the simulation of multiple realistic circumstances in a controlled environment.

- **RQ-1**: When exposed to aggregations of instance cohorts during training, how well can a model perform individual inference?

We next look to examine how sampling can be exploited to provide adequate performance when only an aggregate instance is provided in training. This would allow for the use of training data that is currently not considered valuable in a machine learning approach. We look to approximately reconstruct the original cohort from which the aggregate was computed and find a method which is consistent across multiple forms of data in allowing the reconstruction of a dataset.

- **RQ-2**: Can sampling from the parameterised neighbourhood of an aggregate instance improve inference?

4.2 Datasets

Image Datasets. We conduct the simulation operation over a number of standard base datasets for image classification. In particular, we use MNIST and CIFAR-10 for the IPAT evaluation experiments. These experiments allow us granular control of the level of aggregation due to the relative abundance of samples compared to feature-based datasets. The MNIST [9] dataset is comprised of 28x28 handwritten digits with a held-out train:test split of 60,000 training and 10,000 test images. The CIFAR-10 [17]is a labelled subset of the tiny images dataset. It contains a total of 60000 32x32 colour images labelled with 10 classes.

For each base dataset, the simulation operates only on the training fold. The test set is kept unchanged. We conduct the simulation of data aggregation across cohorts with different levels of granularity. A smaller value of K means the IPAT task with this setting is relatively challenging as there are very few training examples. We conduct experiments over a range of K values $\in [50, 100, 200, 500, 1000]$.

Feature-Based Datasets. In addition to the image datasets, we conduct experiments on feature-based datasets as well. These real datasets closely reflect the actual use-case which motivated us to explore the IPAT task in the first place, being health and census studeies where privacy must be considered. In particular, we use chart data paired with heart arrhythmia diagnosis from the UCI Cardiac Arrhythmia dataset [1], within this dataset there are 452 instances labelled with 17 labels, we use 271 of the 279 features due to missing data. This choice represents a complex classification problem due to the number of features and labels, we aggregate over sex (nominal), weight (grouped numerical) and grouping by sex and age. Our second dataset is the UCI Bank Marketing Dataset [25] containing personal information for 45211 interactions regarding contact with bank customers with the classification purpose of determining whether or not a person will take a loan from the bank, we use all 17 features. This is a binary classification task and we aggregate over job (nominal), current balance (grouped numerical) and a combination of the two. For the simulated aggregation process, instead of constructing groups with a K-means algorithm, for tabular data we make the grouping more realistic by defining the groups as unique combinations of values of demographic attributes of individuals, e.g., a combination of sex and age.

4.3 Synthetic Data Generation

Since real-life cohort level data from randomized control trials would require significant manual effort to procure and annotate from the published literature, for the sake of evaluating our approach in this paper we focus on the use of simulated data (we leave evaluating our approach on true cohort-level user data for future work). In this section, we describe the simulated setup to generate data for IPAT evaluation that operates on a number of standard datasets used in the supervised learning literature.

In all cases aggregates are unweighted within groups as would be realistic in a real study. We investigate image aggregation not because it represents a realistic situation but instead because we want to further investigate the use of extrapolation from a synthetic mean to improve training efficiency and robustness to attack.

Simulated Aggregated Cohorts. We propose a single method for the simulation of an aggregated dataset. Given a standard classification dataset of any modality, comprised of a set of pairs of instances and labels $\mathcal{T} = \{(\mathbf{x}, \mathrm{y}) : \mathbf{x} \in \mathbb{R}^d,\ \mathrm{y} \in \mathbb{Z}_c\}$, the simulation procedure first partitions the set into c groups by label. A value of K is chosen being the number of instances the aggregation procedure should produce. Depending on the modality, each partition is then sub-partitioned by features to yeild $\frac{K}{c}$ disjoint sets, for example a categorical indicator in a feature based representations or via unsupervised clustering in image representations. After sub-partitions are determined, the training set then consists of K mutually disjoint sets, i.e. $\mathcal{T} = \mathcal{T}^1 \cup \ldots \mathcal{T}^K$. For each $\mathcal{T}^i$, we create a set

$$\mathcal{T}_i = \{\mathbf{z} : \mathbf{z} = \frac{1}{|\mathcal{T}^i|} \sum_{\mathbf{x} \in \mathcal{T}^i} \mathbf{x}\}, \quad (5)$$

i.e., we create a single point that is the average of the constituent data points within that cluster. In other words, in our simulation we use the average operator as the collapsing function ϕ_x of Equation 5. This pre-processing results in a total of K points in our aggregated training set. Replicating the aggregation of real datasets, we use an unweighted average as we do not consider any sample to have greater importance than another. As the label collapsing function, we use the majority over the class labels of the constituent data instances within a cluster (cohort) as the induced label of the collapsed data instance.

Aggregations from Image Data. In this work due to the absence of discrete features in an image representation we use K-Means over each class partitions to yield aggregated instances. Both datasets used in this investigation have 10 class labels such that K can be set as round numbers offering a clear illustration of aggregate inference in low information applications.

Aggregations from Tabular Data. Given a classification dataset comprised of tabular features and class labels, within each class partition we perform aggregations with respect to features chosen for two reasons. Firstly, to represent a practical use case and secondly, for their form being nominal or numerical to investigate the effects of discrete bins over numerical data compared to simple groupings over nominal variables. For example data aggregated over sex or both sex and weight. This sub-partition stage can be repeatedly applied to have subgroups conditioned on multiples features before aggregation. Due to dataset size, the notion of K instances is less applicable such that we instead adopt the pragmatic approach of ensuring that when aggregating over categorical indicators no single instance set is created such that privacy is realistically preserved.

Data Preparation. All image representation values are 0-scaled between [-1, 1]. For tabular data, MinMax scaling is used across numerical features. The features are converted into a dense tensor using feature columns. To aggregate over nominal features we first group by class label, this represents a similar format to anonymized aggregate medical trials in which there is an overwhelming majority of healthy people and a few examples of each diagnosis. We then group by binning over a numerical feature and or grouping by nominal features. Reconstruction is identical to images as the processed features are a normalised dense tensor, as such no distinction is made between features columns when adding noise.

4.4 Methods Investigated

We present architectures to measure the effectiveness of our proposed IPAT training across multiple modalities.

4.4.1 General Architecture.

4.4.1 General Architecture. In this experiment each architecture contains two components, a task specific encoder and a set of at least one multi-layer perceptron (MLP) classification heads as shown in Figure 1. For IPAT tests there is also a stochastic component, this component either perturbs input before it enters the network (component: neighbourhood sample in Figure 1) or samples from a latent distribution formed over encoder logits, effectively perturbing the latent representation (component: latent sample in Figure 1). We investigate two methods for neighbourhood classification, that being p pre-generated neighbourhood samples per aggregate which are then input to a standard task specific model or sampling at run time where neighbourhood sampling is applied at training time before input to the model.

4.4.2 Baselines. We investigate the performance of simple architectures in predicting on atomic instances when trained on aggregate instances. Alongside these baselines we present oracle versions of each of these baseline architectures, that being a model that has been trained on the entire original dataset prior to aggregation. We use these models as an upper bound on the performance achievable in IPAT training. In each case identical classification heads with a set of [512, 256, 128] layers each.

- **Convnet (C):** A Resnet-18 model with randomly initialized weights as a standard image classification baseline with classification head
- **Multi-Layer Perceptron (MLP):** 2 layer MLP [512, 256] as a standard tabular classification baseline with classification head

4.4.3 Image Architectures.

- **Ensemble of Classifiers with Input Reconstruction (Recons-EC):** Ensemble of image classifiers with dataset reconstruction, parameterised by ϵ and samples per aggregate tied with ensemble members parameterised by p. Encoder is identical to C.
- **Ensemble of Classifiers with Latent Perturbation (Latent-EC):** Ensemble of image classifiers with latent perturbation, samples per aggregate tied with ensemble members parameterised by p. Encoder is identical to C.

4.4.4 Tabular Architectures.

- **Ensemble of Classifiers with Input Reconstruction (Recons-EMLP):** Ensemble of tabular classifiers with dataset reconstruction, parameterised by ϵ and samples per aggregate tied with ensemble members parameterised by p. Encoder stage is identical to MLP.
- **Ensemble of Classifiers with Latent Perturbation (Latent-EMLP):** Ensemble of tabular classifiers with latent perturbation, samples per aggregate tied with ensemble members parameterised by p. Encoder stage is identical to MLP.

The case $p = 0$ represents a standard classification task where the model is only exposed to the aggregate instances. Here we sample directly from the aggregate instance neighbourhood. This baseline with $p = 0$ tests the hypothesis of RQ-1 that inference quality is negatively affected by aggregate training. This is already clear as the training set is not only small but noisy due to aggregation. By increasing p we assess the hypothesis of research question 2 that reconstructing a dataset from the neighbourhood of aggregate instances can improve inference in standard classification networks.

Table 1: IPAT results on MNIST and CIFAR-10 image datasets for different granularity of aggregation (different values of K). The best results across different K are reported (the optimal value of p shown as the p^* column). C is a standard network without IPAT sampling used as a baseline. Models denoted with * are trained on individual instances representing the upper bound on performance. Highest performing methods by accuracy are marked bold.

				MNIST				CIFAR-10			
Model	K	p^*	ϵ	Acc	F1	Prec	Rec	Acc	F1	Prec	Rec
C*	N/A	N/A	N/A	0.9943	0.9943	0.9943	0.9943	0.7765	0.7733	0.7784	0.7765
C	50	N/A	N/A	0.2623	0.2623	0.2623	0.2623	0.2092	0.2092	0.2092	0.2092
Latent-EC	50	20	N/A	0.1189	0.1189	0.0398	0.0240	0.1015	0.0590	0.1015	0.0416
Recons-EC	50	20	0.1	**0.8027**	0.7999	0.8027	0.8027	**0.2983**	0.2861	0.2983	0.2983
C	100	N/A	N/A	0.7682	0.7618	0.7682	0.7682	0.2116	0.2116	0.2116	0.2116
Latent-EC	100	20	N/A	0.1736	0.1520	0.1645	0.1413	0.1077	0.1077	0.1077	0.0487
Recons-EC	100	20	0.1	**0.8576**	0.8549	0.8576	0.8576	**0.3298**	0.3245	0.3298	0.3298
C	200	N/A	N/A	0.8276	0.8237	0.8276	0.8276	0.2839	0.2573	0.2839	0.2839
Latent-EC	200	5	N/A	0.1490	0.1522	0.1518	0.1526	0.1149	0.1149	0.1149	0.0746
Recons-EC	200	20	0.1	**0.8966**	0.8948	0.8966	0.8966	**0.3495**	0.3430	0.3495	0.3495
C	500	N/A	N/A	0.8842	0.8820	0.8842	0.3412	0.3412	0.3412	0.3412	0.3412
Latent-EC	500	10	N/A	0.3579	0.3622	0.3509	0.3579	0.1168	0.1168	0.1168	0.0526
Recons-EC	500	20	0.1	**0.9327**	0.9317	0.9327	0.9327	**0.3970**	0.3893	0.3970	0.3970
C	1000	N/A	N/A	0.9099	0.9083	0.9099	0.9099	0.3490	0.3330	0.3490	0.3490
Latent-EC	1000	3	N/A	0.5619	0.5619	0.5688	0.5392	0.1174	0.1174	0.1174	0.0752
Recons-EC	1000	20	0.1	**0.9481**	0.9473	0.9481	0.9481	**0.4105**	0.4038	0.4105	0.4105

4.5 Parameter Settings

We investigate IPAT across both image and tabular inputs. On both image and tabular inputs, empirical testing is performed over $p \in [0, 3, 5, 10, 20]$ being the number of classifiers in ensemble which is tied to the number of perturbations generated per aggregate instance, and in the case of neighbourhood sampling, $\epsilon \in [0.001, 0.005, 0.01, 0.05, 0.1]$ being the variance in the noise added to the instances. By testing our method under varying levels of aggregation we assess how this method can generalize not only to different forms of data but to varying levels of information within training data.

Hyperparameters. We utilise PyTorch [26] throughout this investigation. Reported hyper-parameter configurations are run 5 times and averaged to account for the stochastic nature of this approach, this provides similar effect to cross-validation which is less appropriate due to the distinct differences in the test and training data, to use part of the training data to validate would be to predict on aggregated data which is not useful in this case. In all cases we use a batch size of 32. Parameter tuning is performed on ϵ and p with both image and tabular instances. We present the highest performing model with p sampling for each aggregated variant. In all cases learning rate at $1e^{-4}$ and 50 training epochs are constant. We use the Adam optimizer with default torch settings for decay betas. For the temperature of the Gumbel-Softmax distribution we use an anneal rate of $3e^{-5}$ to gradually reduce the temperature from 0.5 to 0.2 throughout training.

5 RESULTS

In this section, we present and discuss results from empirical testing of IPAT training. In all cases we tune p, being the number of samples and ensemble members, presenting the best tuned value of p, being the optimal number of members in the ensemble of classifers and when applying neighbourhood sampling we tune ϵ, being the size of the neighbourhood from which we sample pseudo-instances.

Table 2: IPAT results on aggregated UCI Cardiac Arrhythmia and UCI Bank Marketing conditioned on different features. The best results across different p (number of samples) are reported (the optimal value of p shown as the p^* column). Models denoted with * are trained on individual instances representing the upper bound on performance. Highest performing methods by accuracy are marked bold.

			Cardia Arrythmia				Bank Marketing			
Model	p^*	ϵ	Acc	F1	Prec	Rec	Acc	F1	Prec	Rec
MLP*	N/A	N/A	0.9451	0.9451	0.9451	0.9451	0.8900	0.8900	0.8900	0.8900
					Nominal					
MLP	N/A	N/A	0.0433	0.0433	0.0635	0.0700	0.0970	0.0833	0.0833	0.0833
Latent-EMLP	5	N/A	**0.3787**	0.0605	0.0625	0.0586	**0.5591**	0.5591	0.5591	0.5591
Recons-EMLP	5	0.1	0.1758	0.0796	0.0588	0.0110	0.1770	0.1770	0.1770	0.1770
					Nominal and Numerical					
MLP	N/A	N/A	0.0378	0.0378	0.0634	0.0691	0.1771	0.1771	0.1771	0.1771
Latent-EMLP	10	N/A	**0.8658**	0.6120	0.6120	0.6120	**0.8789**	0.8789	0.8789	0.8789
Recons-EMLP	5	0.001	0.3077	0.1403	0.2800	0.0769	0.6591	0.6591	0.6591	0.6591
					Numerical					
MLP	N/A	N/A	0.0206	0.0206	0.0630	0.0711	0.1010	0.1010	0.1010	0.1010
Latent-EMLP	3	N/A	**0.8151**	0.0.6050	0.6250	0.5860	**0.8709**	0.8709	0.8709	0.8709
Recons-EMLP	5	0.05	0.2527	0.0966	0.1000	0.0220	0.4369	0.4369	0.4369	0.4369

5.1 IPAT on Images

Results for reconstruction on MNIST and CIFAR-10 are presented in Table 4.5. There is a clear trend in accuracy reducing as problem complexity increases. This decrease in effectiveness is more pronounced in CIFAR-10 tests which could be attributed to the perturbation of all RGB values, meaning colours are disturbed with the ϵ-neighbourhood across 3 values per pixel. Regardless of this drop off a significant improvement is still observed in smaller values of K as seen for Recons-EC relative to C. We observed variance in accuracy across a given value of K was negligible across multiple reconstructions ($\sigma < 0.001$), in training with a fixed value of epsilon reconstructing an aggregate set multiple times we observed low variance ($\sigma < 0.0001$) in accuracy. In our experiments larger values of ϵ yielded better results but across different ensemble sizes and input modalities there is no consistent trend as observed across Tables 3 and 5.2 as such this value should be tuned. Larger values of p led to lower variance in classification performance (as shown in Figures 2). Furthermore, the regularizing effect of the ensemble yields a small improvement in test accuracy with less variance when using greater aggregation which can be observed in Table 4.5 and Figure 2.

Latent perturbation of aggregate images yields extremely poor results as which consistently under perform a standard classification baseline as observed in Table 4.5 under Latent-EC. This would suggest that the model does not capture salient features when exposed to a small set of instances. We discuss this effect further in Section 5.3.

5.2 IPAT on Tabular Data

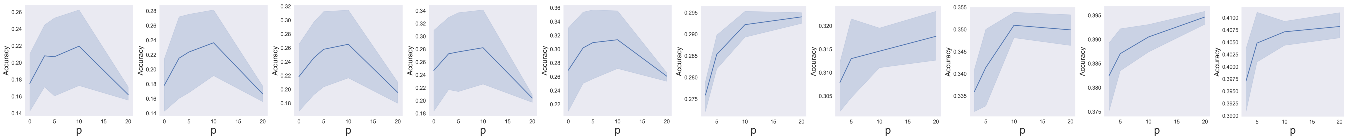
Results for classification of the UCI cardiac arrhythmia and Bank Marketing datasets are presented in Table 5.2. Strong results are achieved by using an encoder structure with latent perturbation (Latent-EMLP). The performance is still correlated with the number of instances observed as accuracy decreases depending on the level of aggregation. This method shows strong generalization properties over two out of three aggregate forms with the third being aggregation over nominal data (Sex, Job) which has the heaviest aggregation applied with only 24 instances in each dataset being shown to the model during training still achieving an accuracy improvement of 0.33 and 0.46 respectively. Furthermore, these experiments show that on tabular data we can generalize well from aggregates as Latent-EMLP achieves large increases in accuracy across each aggregated variation. We observe that in a binary classification setting (Bank Marketing) the majority of model performance is recovered relative to the baseline oracle model. In future work, we look to investigate if a membership attack can still be performed given these constant perturbations to the classification head input. Training time is significantly reduced in Latent-EMLP relative to both the oracle MLP and offline Recons-EMLP as the model does not explicitly generate new samples so is efficient during training.

Results using neighbourhood sampling are significantly weaker than latent perturbation as seen under Recons-EMLP. We believe the random noise across each input dimension is less appropriate for tabular data as each dimension represents a specific feature as opposed to images where there is a sense of locality exploited by convolutional layers and as such neighbourhood sampling across each dimension does not yield particularly useful samples for tabular data. Variance in accuracy under ϵ is low when using appropriate values ($\Delta\text{Accuracy} < 0.05$). To maximise performance a machine learning practitioner could use an MLP trained under IPAT in ensemble with a forest method, these neural-forest ensembles have been shown to outperform either method alone [24; 35; 38].

5.3 Ablation Study for Input Perturbation

We ablate components of our architecture to assess their necessity and effect on model performance. We investigate both the offline generation of neighbourhood samples for training (Recons-* variations) and the generation of samples during training (Perturb-* variations) in Tables 3 and 4 to assess two questions. Firstly, does an ensemble better generalize when using data reconstruction and secondly, do these reconstructions need to be computed offline such that the model is repeatedly exposed to the same pseudo-instances. We consider offline generation to be the pre-computation of a set of pseudo-instances from aggregates that are then passed into a classic task-specific neural network. The benefit of sampling during training is a significant speed increase similar to latent perturbation as for n aggregates, each member of the ensemble of classifiers is exposed to n samples compared to $n \times p$ samples when using ad-hoc generation. Furthermore we explicitly compare a sequential classifier (Recons-C / Recons-MLP) versus an ensemble of classifiers under both modalities. We present results for all tabular aggregate variations but only aggregated MNIST variations for brevity though similar trends were observed in CIFAR-10 experiments.

As observed in Table 3 comparing Latent-EC and Recons-EC, we observe that image classifiers perform significantly better using pre-generated samples across all classifiers in the ensemble relative to sampling at training time comparing Recons-EC to Latent-EC. Furthermore ensembles are largely insensitive to a chosen value of ϵ as shown in the contrast in variance shown across figure 2 though due to the small time taken to perform random perturbations it is worthwhile to tune ϵ. The poor performance of the constant



(a) $K = 50$ **(b)** $K = 100$ **(c)** $K = 200$ **(d)** $K = 500$ **(e)** $K = 1000$ **(f)** $K = 50$ **(g)** $K = 100$ **(h)** $K = 200$ **(i)** $K = 500$ **(j)** $K = 1000$

Figure 2: Comparison of Test Accuracy on a sequential classifier - Recons-C (a-e), and an ensemble of image classifiers - Recons-EC (f-j) for different granularity of aggregation (K) on CIFAR-10. The x-axis, p, denotes the number of neighbourhood samples which also the number of ensemble members. The shaded area represents the change in performance using different values of ϵ.

Table 3: Ablation across aggregated MNIST. Recons-C represents a sequential classifier with dataset reconstruction and Perturb-EC represents an ensemble of classifiers with input sampling occurring at train time.

Model	K	p^*	ϵ	Acc	F1	Prec	Rec
Recons-C	50	20	0.1	0.7993	0.7962	0.7993	0.7993
Perturb-EC	50	5	0.01	0.1862	0.1862	0.1773	0.0983
Recons-EC	50	20	0.1	**0.8027**	0.7999	0.8027	0.8027
Recons-C	100	10	0.001	0.8416	0.8381	0.8416	0.8416
Perturb-EC	100	5	0.01	0.2121	0.2121	0.2025	0.0891
Recons-EC	100	20	0.1	**0.8576**	0.8549	0.8576	0.8576
Recons-C	200	20	0.05	0.8906	0.8885	0.8905	0.8905
Perturb-EC	200	3	0.1	0.2706	0.3228	0.2692	0.4029
Recons-EC	200	20	0.1	**0.8966**	0.8948	0.8966	0.8966
Recons-C	500	20	0.01	0.9291	0.9282	0.9291	0.9291
Perturb-EC	500	3	0.001	0.3962	0.4225	0.3888	0.4647
Recons-EC	500	20	0.1	**0.9327**	0.9317	0.9327	0.9327
Recons-C	1000	20	0.1	0.9428	0.9420	0.9428	0.9428
Perturb-EC	1000	3	0.001	0.7003	0.7003	0.6958	0.7758
Recons-EC	1000	20	0.1	**0.9481**	0.9473	0.9481	0.9481

Table 4: Ablation across aggregated UCI Cardiac Arrhythmia and UCI Bank Marketing. Recons-MLP represents a sequential classifier with dataset reconstruction and Perturb-EMLP represents an ensemble of classifiers with input sampling occurring at train time.

			Cardia Arrythmia				Bank Marketing			
					Nominal					
Model	p^*	ϵ	Acc	F1	Prec	Rec	Acc	F1	Prec	Rec
Recons-MLP	1	0.05	**0.1758**	0.0953	0.1519	0.1319	0.1751	0.1751	0.1751	0.1751
Perturb-EMLP	5	0.01	0.0169	0.0605	0.0625	0.0586	0.1201	0.1201	0.1201	0.1201
Recons-EMLP	5	0.1	0.1758	0.0796	0.0588	0.0110	**0.1770**	0.1770	0.1770	0.1770
					Nominal and Grouped Numerical					
Recons-MLP	1	0.05	**0.3077**	0.1563	0.3091	0.1868	0.5671	0.5671	0.5671	0.5671
Perturb-EMLP	3	0.05	0.0209	0.0209	0.0627	0.0774	0.2871	0.2871	0.2871	0.2871
Recons-EMLP	5	0.001	0.3077	0.1403	0.2800	0.0769	**0.6591**	0.6591	0.6591	0.6591
					Numerical					
Recons-MLP	1	0.1	0.2308	0.1524	0.1538	0.0659	0.3964	0.3964	0.3964	0.3964
Perturb-EMLP	3	0.1	0.0302	0.0302	0.0628	0.0646	0.1422	0.1422	0.1422	0.1422
Recons-EMLP	5	0.05	**0.2527**	0.0966	0.1000	0.0220	**0.4369**	0.4369	0.4369	0.4369

perturbation of input further strengthens our explanation that when exposed to a small number of examples, convolutional layers cannot extract salient features from aggregated instances.

We observe small performance changes between sequential and ensemble classifiers using neighbourhood sampling with tabular data. The ensemble of classifiers generalize on fewer samples per aggregate (p) however, in two ablation cases a sequential classifier with reconstruction outperforms an ensemble by a small margin. We infer that the use of an ensemble architecture is less significant to performance when using tabular neighbourhood sampling. We observed that using latent perturbation an ensemble of classifier consistently outperformed a sequential classifier.

5.4 Limitations and Applications

The loss of performance in experiments with lower sample counts due to heavier aggregation, shows that much work can still be done in this area. In the case of tabular data only the encoder can exploit inductive biases e.g use of a tabular transformer that can process different types of features, forcing the classification head currently to be a feed-forward network, with a decoder this could be improved though one could have issues creating realistic data due to the lack of samples, we leave this problem for future work.

RQ-1: We have shown that models trained on aggregate instances have substantially lower accuracy than those trained on individual instances. This represents a common situation in applications of deep learning such as empirical population studies and as such warrants new methods to improve performance in low information environments.

RQ-2: Reconstruction outperforms a standard training process with ensemble networks generalizing on a small number of samples. This method is computationally inexpensive and as such we are optimistic that even with further developments in IPAT specific architectures to improve performance, this method will scale well.

Concluding Remarks. We have presented a formal definition of the problem of learning from aggregated data which we call IPAT. By improving inference performance on atomic instances when training on aggregates, machine learning practitioners can exploit data that would otherwise not be considered when using neural approaches. Our experiment looked to improve individual instance inference performance of machine learning methods when trained on aggregate instances which we describe as IPAT. Our initial methods using parameterized noise to extrapolate from aggregated cohorts of instances improve accuracy across multiple levels of aggregation in both image and tabular data. It is interesting to observe how using aggregate instances that are by representative of a true set of samples, we can reconstruct a small number of pseudo-instances and see improvements in accuracy. We aim to investigate architectures that exploit this observation to potentially reduce training time significantly with a small reduction in accuracy. We plan to investigate the robustness of both input and latent perturbation to adversarial and membership attacks in future on real RCT data.

REFERENCES

[1] Altay Guvenir, Burak Acar, H. M. UCI Cardiac Arrythmia Database, 2014.

[2] Aus, G., Abrahamsson, P. A., Ahlgren, G., Hugosson, J., Lundberg, S., Schain, M., Schelin, S., and Pedersen, K. Three-month neoadjuvant hormonal therapy before radical prostatectomy: a 7-year follow-up of a randomized controlled trial. *BJU international 90*, 6 (2002), 561–566. Publisher: BJU Int.

[3] Babenko, B. Multiple Instance Learning: Algorithms and Applications.

[4] Bai, H., Zhou, F., Hong, L., Ye, N., Chan, S.-H. G., and Li, Z. NAS-OoD: Neural Architecture Search for Out-of-Distribution Generalization.

[5] Beijbom, O., Hoffman, J., Yao, E., Darrell, T., Rodriguez-Ramirez, A., Gonzalez-Rivero, M., and Guldberg, O. H. Quantification in-the-wild: datasets and baselines, Nov. 2015. arXiv:1510.04811 [cs].

[6] Bhowmik, A. *Learning from aggregated data.* Thesis, Feb. 2019. Accepted: 2019-04-04T15:18:49Z.

[7] Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramer, F. Membership Inference Attacks From First Principles. *Proceedings - IEEE Symposium on Security and Privacy 2022-May* (Dec. 2021), 1897–1914. ISBN: 9781665413169 Publisher: Institute of Electrical and Electronics Engineers Inc.

[8] Chen, Y., and Vorobeychik, Y. Regularized Ensembles and Transferability in Adversarial Learning.

[9] Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine 29*, 6 (2012), 141–142. Publisher: IEEE.

[10] Forman, G. Counting positives accurately despite inaccurate classification. In *Proceedings of the 16th European conference on Machine Learning* (Berlin, Heidelberg, Oct. 2005), ECML'05, Springer-Verlag, pp. 564–575.

[11] Gilotte, A., Yahmed, A. B., and Rohde, D. Learning from aggregated data with a maximum entropy model, Oct. 2022. arXiv:2210.02450 [cs].

[12] Green, H. J., Pakenham, K. I., Headley, B. C., Yaxley, J., Nicol, D. L., Mactaggart, P. N., Swanson, C., Watson, R. B., and Gardiner, R. A. Altered cognitive function in men treated for prostate cancer with luteinizing hormone-releasing hormone analogues and cyproterone acetate: a randomized controlled trial. *BJU international 90*, 4 (2002), 427–432. Publisher: BJU Int.

[13] Hora, S. C. Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability Engineering and System Safety 54*, 2-3 (Nov. 1996), 217–223.

[14] Jang, E., Gu, S., and Poole, B. Categorical Reparameterization with Gumbel-Softmax. In *ICLR (Poster)* (2017), OpenReview.net.

[15] Kaur, J. N., Kiciman, E., and Sharma, A. Modeling the Data-Generating Process is Necessary for Out-of-Distribution Generalization.

[16] Kiureghian, A. D., and Ditlevsen, O. Aleatory or epistemic? Does it matter? *Structural Safety 31*, 2 (Mar. 2009), 105–112. Publisher: Elsevier.

[17] Krizhevsky, Vinod Nair, G. H. CIFAR-10 (Canadian Institute for Advanced Research).

[18] Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *Advances in Neural Information Processing Systems 2017-December* (Dec. 2016), 6403–6414. Publisher: Neural information processing systems foundation.

[19] Leino, K., Wang, Z., and Fredrikson, M. Globally-Robust Neural Networks. *CoRR abs/2102.08452* (2021). arXiv: 2102.08452.

[20] Liang, C., He, P., Shen, Y., Chen, W., and Zhao, T. CAMERO: Consistency Regularized Ensemble of Perturbed Language Models with Weight Sharing. 7162–7175. Publisher: Association for Computational Linguistics (ACL).

[21] Liu, Y., Zhao, Z., Backes, M., and Zhang, Y. Membership Inference Attacks by Exploiting Loss Trajectory. 2085–2098. ISBN: 9781450394505 Publisher: Association for Computing Machinery (ACM).

[22] MARTINO, G. D. S., GAO, W., and SEBASTIANI, F. Ordinal text quantification. *Proceedings of the 38th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016)* (July 2016), 937–940.

[23] McNiven, P. S., Williams, J. I., Hodnett, E., Kaufman, K., and Hannah, M. E. An early labor assessment program: a randomized, controlled trial. *Birth (Berkeley, Calif.) 25*, 1 (1998), 5–10. Publisher: Birth.

[24] Mejri, M., and Mejri, A. RandomForestMLP: An Ensemble-Based Multi-Layer Perceptron Against Curse of Dimensionality.

[25] Moro, S., Cortez, P., and Rita, P. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems 62* (2014), 22–31.

[26] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[27] Potapczynski, A., Loaiza-Ganem, G., and Cunningham, J. P. Invertible Gaussian Reparameterization: Revisiting the Gumbel-Softmax. *Advances in Neural Information Processing Systems 2020-December* (Dec. 2019). Publisher: Neural information processing systems foundation.

[28] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-Shot Text-to-Image Generation, Feb. 2021. arXiv:2102.12092 [cs].

[29] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models, Apr. 2022. arXiv:2112.10752 [cs].

[30] Shui, C., Mozafari, A. S., Marek, J., Hedhli, I., and Gagné, C. Diversity regularization in deep ensembles.

[31] Simpson, C. R., Shi, T., Vasileiou, E., Katikireddi, S. V., Kerr, S., Moore, E., McCowan, C., Agrawal, U., Shah, S. A., Ritchie, L. D., Murray, J., Pan, J., Bradley, D. T., Stock, S. J., Wood, R., Chuter, A., Beggs, J., Stagg, H. R., Joy, M., Tsang, R. S. M., Lusignan, S. d., Hobbs, R., Lyons, R. A., Torabi, F., Bedston, S., O'Leary, M., Akbari, A., McMenamin, J., Robertson, C., and Sheikh, A. First-dose ChAdOx1 and BNT162b2 COVID-19 vaccines and thrombocytopenic, thromboembolic and hemorrhagic events in Scotland. *Nature medicine 27*, 7 (July 2021), 1290–1297. Publisher: Nat Med.

[32] Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep Unsupervised Learning using Nonequilibrium Thermodynamics, Nov. 2015. arXiv:1503.03585 [cond-mat, q-bio, stat].

[33] Suri, A., Kanani, P., Marathe, V. J., and Peterson, D. W. Subject Membership Inference Attacks in Federated Learning.

[34] Yi, M., Wang, R., Sun, J., Li, Z., and Ma, Z.-M. Improved OOD Generalization via Conditional Invariant Regularizer.

[35] Young, S., Abdou, T., and Bener, A. Deep Super Learner: A Deep Ensemble for Classification Problems.

[36] Zeng, J., Lesnikowski, A., and Alvarez, J. M. The Relevance of Bayesian Layer Positioning to Model Uncertainty in Deep Bayesian Active Learning.

[37] Zhong, Z., Tian, Y., and Ray, B. Understanding Spatial Robustness of Deep Neural Networks. *CoRR abs/2010.04821* (2020). arXiv: 2010.04821.

[38] Zhou, Z.-H., and Feng, J. Deep Forest.

Received 3 February 2023