Data Augmentation for Compositional Data: Advancing Predictive Models of the Microbiome

Anonymous Author(s)
Affiliation
Address
email

Abstract

Data augmentation plays a key role in modern machine learning pipelines. While numerous augmentation strategies have been studied in the context of computer vision and natural language processing, less is known for other data modalities. Our work extends the success of data augmentation to *compositional data*, i.e., simplex-valued data, which is of particular interest in the context of the human microbiome. Drawing on key principles from compositional data analysis, such as the *Aitchison geometry of the simplex* and *subcompositions*, we define novel augmentation strategies for this data modality. Incorporating our data augmentations into standard supervised learning pipelines results in consistent performance gains across a wide range of standard benchmark datasets. In particular, we set a new state-of-the-art for key disease prediction tasks including colorectal cancer, type 2 diabetes, and Crohn's disease. In addition, our data augmentations enable us to define a novel contrastive learning model, which improves on previous representation learning approaches for microbiome compositional data.

1 Introduction

Data augmentation, i.e., generating synthetic training examples to be used in model fitting, is a core component of modern deep learning pipelines [51, 15]. In computer vision, augmentations such as image resizing and shifting have been used since as early as LeNet-5 [34]. These and many other augmentations have become essential to highly successful state-of-the-art architectures, ranging from AlexNet [33] and ResNet [29] to recent contrastive learning models such as SimCLR [7] and MoCo [28]. As such, a growing body of literature has emerged to develop and characterize data augmentation techniques, particularly in computer vision [43, 61, 11, 8, 9, 60], as well as natural language processing [50, 10, 15]. However, defining useful data augmentations is highly domain-dependent, and fewer works have studied augmentations for more general data modalities, such as tabular data [61, 59]. The goal of our work is to extend the success of data augmentation to a previously unexplored data modality; namely, compositional data.

Compositional data (CoDa) are those that represent the parts of a whole, and therefore carry only *relative* information. Equivalently, we can think of CoDa as a set of datapoints living in the simplex:

$$\mathcal{D} = \left\{\boldsymbol{x}_i \in \Delta^{p-1}\right\}_{i=1}^{n}, \quad \text{where} \quad \Delta^{p-1} = \left\{\boldsymbol{x} \in \mathbb{R}_+^p : \sum_{j=1}^{p} x_j = 1\right\}. \tag{1}$$

Examples of CoDa arise throughout the sciences, most notably in microbiology [21, 20, 47, 2], geochemistry [4, 3], ecology [13], materials science [39], economics [17], and machine learning [22, 23]. As a result, starting with the seminal work of Aitchison [1], numerous bespoke techniques have been developed for the statistical analysis of CoDa [42, 41, 16].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

A rapidly growing area of application for CoDa is the human microbiome, which comprises the populations of microorganisms that reside inside and on the human body [55]. Microbiome data arise from an inexhaustive sampling procedure as a result of high-throughput sequencing [21, 48]. In particular, each feature typically represents the *relative abundance* of some species of microorganism; as such, each observation can be normalized to the simplex prior to downstream analyses [21, 20, 47, 36, 40].

The microorganisms that constitute the microbiome are known to impact human physiology, both in health and in disease [19, 37, 18]. Thus, a central problem in CoDa is to learn the association $\mathbf{x}_i \mapsto y_i$, where $\mathbf{x}_i \in \Delta^{p-1}$ denotes the microbial composition, and y_i denotes the disease status of the ith subject in a clinical study. For example, the composition of the gut microbiome, as recorded from a stool sample, can be predictive of colorectal cancer, which is the third most prevalent form of cancer [14, 32]. This observation offers the potential for a noninvasive alternative to traditional colonoscopy procedures used for early detection of colorectal cancer [53]. In turn, accurate predictive models for microbiome CoDa are a key stepping stone towards achieving this potential [38, 32]. Note that cancer is just one application; the demand for improved predictive models holds more broadly across medicine and biological science [32, 19]. For example, the microbiome has been linked to type 2 diabetes, Crohn's disease, obesity, and others [27, 57, 44, 18].

Classical techniques including logistic regression, support vector machines, and random forests have been widely used as predictive models for microbiome data [56, 5, 35]. More recently, specialized deep learning architectures such as DeepCoDa [46], MetaNN [36], and DeepMicro [40] have been developed. However, the capacity of these deep networks and other expressive models has been limited by the low sample size and high dimensionality of typical microbiome studies. These characteristics have also spurred the use of strong regularization through early stopping, weight decay and dropout, among others [54, 46, 36, 40]. However, no previous works have explored the use of data augmentation for CoDa,[1] which, as we shall demonstrate, provides a cheap and simple technique for boosting the performance of predictive models for this data modality.

Careful consideration of the sample space will motivate our novel data augmentation strategies for CoDa. Our work draws on foundational principles from the field of CoDa, such as the *Aitchison geometry* of the simplex [1], which we combine with popular techniques from the literature on data augmentation, such as Mixup [61] and CutMix [60]. This combination will lead us to define custom data augmentation strategies for CoDa, such as *Aitchison Mixup* and *Compositional CutMix*. In turn, incorporating these novel augmentations into existing supervised learning pipelines will result in consistent performance gains across a wide range of microbiome datasets, including a dozen standard benchmarks from the Microbiome Learning Repo [56]. These performance gains are particularly large for some deep models, for example, DeepCoDa gains more than 10% in test AUC for discriminating colorectal cancer from healthy controls, and over 20% for type 2 diabetes. The gains are also significant across other expressive model families, including random forests and gradient boosting machines. Overall, we set a new state-of-the-art on 8 out of 12 benchmark learning tasks, including clinically relevant disease prediction tasks. Of the remaining 4 tasks, 2 were easily separable, with 100% test accuracy irrespective of whether data augmentation was used. Importantly, our augmentations rarely hurt model performance, and in the few instances that this was the case, the drops were typically of only 1% in test AUC.

In addition to supervised learning, our novel augmentations will allow us to define *contrastive representation learning for CoDa*, which to the best of our knowledge, represents the first contrastive learning model for compositional data. Our novel data augmentations are at the core of this approach; the contrastive loss uses randomly augmented training examples to define a self-supervised optimization objective whose labels are generated from unlabelled data [26, 12, 7, 28]. In particular, our contrastive model is trained to discriminate between compositions that were generated as random augmentations of the same training example, and those that were generated as random augmentations of different training examples. Our implementation is adapted from SimCLR [7], but using a smaller network architecture together with our novel augmentations. Unlike SimCLR, we use our individual augmentation strategies in isolation, and find them sufficient to surpass previous representation learning approaches for microbiome CoDa [2, 40]. Altogether, we expect our data augmentations

[1] With the possible exception of dropout, which can be viewed as a form of data augmentation. Note that applications of dropout to CoDa, such as [36], use standard implementations that do not exploit the special structure of CoDa.

will enable significant future progress, possibly in combination with novel architectures, in both supervised and representation learning pipelines for microbiome CoDa.

2 Related Work

Data augmentation: numerous data augmentation strategies have been proposed in the context of image data [51] and text data [15]. LeNet-5 [34] used random shifts and resizing, and AlexNet [33] used reflections and color distortions, which have become widely adopted in computer vision [29]. More general purpose data augmentations include Mixup, which generates synthetic samples by taking convex combinations of training examples, and has been applied to both image and tabular data [61]. Random masking, whereby random input features are hidden during training, has also seen a diverse range of applications, including natural language processing (e.g., BERT [10]), computer vision (e.g., Cutout [11]), and tabular data (e.g., VIME [59]). In the context of CoDa, masking is related to *subcompositions*, which we will use to define an analogous augmentation on the simplex. CutMix [60] is akin to both masking and Mixup, in that random patches from different images are pasted together to generate new synthetic images. We note that a subsequent line of work has developed techniques to automatically select optimal data augmentations for a given dataset [8, 9], somewhat akin to hyperparameter optimization. While the data augmentation strategies that we present in the next Section could also be combined and finetuned in similar ways, such experimentation falls outside the scope of our work and is left to future research.

Microbiome models: random forests remain a strong baseline across many microbiome studies, due to their expressivity and robustness [54]. Gradient boosting machines such as XGBoost provide similarly strong performance and have also enjoyed significant adoption [54]. AutoML [30] has shown potential for microbiome data, with a recent approach called mAML achieving state-of-the-art results on several benchmarks [58]. mAML uses cross-validation to automatically select the best model class and hyperparameters for each learning task. In addition, several specialized deep learning architectures have been proposed for microbiome CoDa. DeepCoDa [46] introduced the log-bottleneck layer, which ensures the hidden units remain scale-invariant, a key desiderata in CoDa models. This architecture obtained strong predictive performance using weight decay regularization. MetaNN [36] employs a multilayer architecture regularized using dropout [52]. The authors of MetaNN also considered generating synthetic data from a negative binomial distribution, but found no performance improvements beyond the use of dropout. DeepMicro [40] uses a deep autoencoder architecture to learn low-dimensional representations of the microbial composition. In turn, these representations are fed to a downstream classifier trained with a supervised objective.

Contrastive learning: the goal of contrastive learning is to learn low-dimensional representations by optimizing some *pretext task*, where the objective function is similar to those used for supervised learning, but using labels that are derived from unlabelled data only [26]. Commonly, the pretext task is to discriminate augmented instances of the same training example from augmented instances of different training examples [12, 7, 28]. Recent contrastive learning architectures have enjoyed tremendous success, setting the state-of-the-art across a range of computer vision benchmarks [7, 28, 31, 6]. We highlight SimCLR [7] for its relative simplicity and strong empirical performance. This method uses a nonlinear projection head between the representations and the contrastive loss computed with normalized embeddings. Our own experiments on contrastive learning borrow these elements from SimCLR, but use a smaller network architecture together with our specialized data augmentations for CoDa.

3 Methods

In the following subsections, we introduce 3 novel data augmentation techniques for CoDa. Note that many variations of our augmentation schemes could be considered; the versions we present here are intended to be as concise as possible. Our goal is not to design augmentations that are empirically or theoretically optimal, but rather to demonstrate the effectiveness of our novel methodology by establishing simple and performant baselines. For instance, whenever a random mixing parameter is required, we use a $U(0, 1)$ distribution, even though other choices can likely result in increased performance. For clarity, we will focus on the classification setting; generalizing to regression problems is straightforward. We will use the notation $\mathbf{v}, \mathbf{x} \in \Delta^{p-1}$ for simplex-valued vectors, $\lambda \in \mathbb{R}$ a scalar, and $\mathcal{D} = \{\mathbf{x}_i, y_i\}$ our training data.

3.1 Aitchison Mixup

Aitchison [1] defined a Hilbert space structure on the simplex, known as the *Aitchison geometry*, with the following vector addition, scalar multiplication, and inner product:

$$\mathbf{v} \oplus \mathbf{x} = \frac{1}{\sum_{j=1}^{p} v_j x_j}[v_1 x_1, \ldots, v_p x_p], \tag{2}$$

$$\lambda \odot \mathbf{x} = \frac{1}{\sum_{j=1}^{p} x_j^{\lambda}}[x_1^{\lambda}, \ldots, x_p^{\lambda}], \tag{3}$$

$$\langle \mathbf{v}, \mathbf{x} \rangle = \frac{1}{2p} \sum_{j=1}^{p} \sum_{k=1}^{p} \log\left(\frac{v_j}{v_k}\right) \log\left(\frac{x_j}{x_k}\right). \tag{4}$$

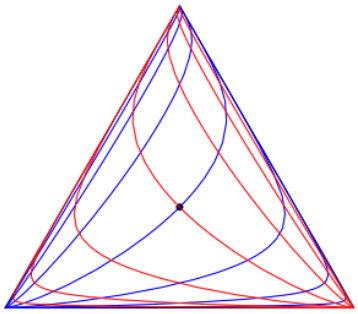

Figure 1: Orthogonal grid on Δ^2, in the Aitchison sense [42]. The centroid of the simplex corresponds to the additive identity. The red lines are parallel and equally spaced by 1 unit in Aitchison distance, as are the blue lines. The red lines are also orthogonal to the blue lines.

This geometry provides a principled definition of linear combinations and a distance metric on the simplex; parallel and orthogonal lines are illustrated in Figure 1. The Aitchison geometry is closely related to the *centered-log-ratio* transformation [1], which is an inverse of the softmax and defines an isometry between Δ^{p-1} and Euclidean space [42]. Taken together, these properties form the basis of much of CoDa methodology [1, 42, 41, 16].

Our first data augmentation strategy, which we call *Aitchison Mixup*, is based on taking linear combinations of the training points, in the Aitchison sense. While general linear combinations may be used, for simplicity we focus on intra-class pairwise convex combinations. Namely, each new datapoint is sampled as follows:

1. Draw a class c from the class prior and draw $\lambda \sim U(0, 1)$.
2. Draw two training points i_1, i_2 such that $y_{i_1} = y_{i_2} = c$, uniformly at random.
3. Set $\mathbf{x}^{\text{aug}} = (\lambda \odot \mathbf{x}_{i_1}) \oplus ((1 - \lambda) \odot \mathbf{x}_{i_2})$ and $y^{\text{aug}} = c$.

Put more succinctly, for each class we generate convex combinations of the points in that class, in the Aitchison sense. Note this augmentation strategy is a CoDa analogue of Mixup, which generates Euclidean convex combinations of images and tabular data [61]. Much like Mixup is capable of boosting predictive accuracy by regularizing models towards a linear decision boundary in the regions between training examples, Aitchison Mixup aims to enforce the equivalent regularization in the Aitchison geometry of the simplex.

Note that, once we restrict to convex combinations (as opposed to general linear combinations), our augmented data will remain inside the simplex regardless of whether we operate in the Euclidean or the Aitchison geometry. However, we found the latter to be more empirically effective, perhaps unsurprisingly given the associated vector space structure of the simplex. Note also that this augmentation strategy may be generalized straightforwardly by taking non-convex combinations, or by including multiple training points from different classes; the study of such approaches is left to future work.

3.2 Random Subcompositions

In CoDa, a subcomposition of $\mathbf{x} \in \Delta^{p-1}$ refers to a lower-dimensional composition, $\mathbf{x}^{\text{sub}} \in \Delta^{k-1}$ with $k < p$, formed by taking a subset of the elements of $\mathbf{x}$ and renormalizing to a unit total. Namely, if $j_1, \ldots, j_k$ denotes a subset of the indices $\{1, \ldots, p\}$, the corresponding subcomposition is defined as:

$$\mathbf{x}^{\text{sub}} = \frac{1}{x_{j_1} + \cdots + x_{j_k}}[x_{j_1}, \ldots, x_{j_k}]. \tag{5}$$

We can generate augmented data by taking *Random Subcompositions* of the training points; such a strategy is analogous to masking in language data or cropping in image data. Since our predictive models require inputs of fixed dimension p, rather than discarding elements of $\mathbf{x}$ we simply zero them out. Thus, each new datapoint is generated as follows:

1. Draw $\lambda \sim U(0,1)$. Draw a training point i uniformly at random and set $\tilde{\mathbf{x}} = \mathbf{x}_i$.
2. For each $j \in \{1, \ldots, p\}$, draw $I_j \overset{iid}{\sim} \text{Bernoulli}(\lambda)$, and set $\tilde{x}_j = 0$ if $I_j = 0$.
3. Set $\mathbf{x}^{\text{aug}} = \tilde{\mathbf{x}}/(\sum_{j=1}^{p} \tilde{x}_j)$ and $y^{\text{aug}} = y_i$.

In short, we zero out random entries of the training points and renormalize. Intuitively, this strategy encourages our predictive models to become robust to partially observed inputs. Note that, distinctively from random masking, our augmentation includes an additional renormalization to a unit total. Importantly, this renormalization ensures that the augmented samples remain in the support of the training data, i.e., the simplex.

Note that many CoDa models, including mAML and DeepCoDa, apply log transformations to their inputs, and therefore require that these be non-zero. For this reason, we implement a zero-replacement step where we add a small positive quantity to all the parts of each composition and renormalize. This small quantity is set to $1/L_i$, where L_i corresponds to the library size from the high-throughput sequencing procedure. L_i can simply be thought of as a large number such that $1/L_i$ is smaller than the non-zero components of $\mathbf{x}_i$. Note that this transformation is a standard preprocessing step in CoDa [47].

3.3 Compositional CutMix

Our third augmentation scheme, which we call *Compositional CutMix*, combines elements of our previous 2 augmentations. Like in Mixup, we generate new datapoints by combining pairs of training points from the same class. However, instead of combining these training points linearly (in the Aitchison sense), we take complementary subcompositions and renormalize. Namely, we generate each new datapoint as follows:

1. Draw a class c from the class prior and draw $\lambda \sim U(0,1)$.
2. Draw two training points i_1, i_2 such that $y_{i_1} = y_{i_2} = c$, uniformly at random.
3. For each $j \in \{1, \ldots, p\}$, draw $I_j \overset{iid}{\sim} \text{Bernoulli}(\lambda)$, and set $\tilde{x}_j = x_{i_1 j}$ if $I_j = 0$ or $\tilde{x}_j = x_{i_2 j}$ if $I_j = 1$.
4. Set $\mathbf{x}^{\text{aug}} = \tilde{\mathbf{x}}/(\sum_{j=1}^{p} \tilde{x}_j)$ and $y^{\text{aug}} = c$.

Note this strategy can be thought of as a CoDa analogue of *CutMix*, whereby new images are formed by pasting together patches from different training images [60]. Note also that in computer vision, CutMix and related methods take *local* image patches as opposed to randomly sampled pixels. Likewise, one could sample microbial subcompositions according to biologically relevant groupings, for example using a phylogenetic tree [47]. Such a strategy would require incorporating additional domain knowledge and is left to future work, but we expect it would further increase the quality of our data augmentations. This remark applies both to Random Subcompositions and Compositional CutMix.

4 Experiments

4.1 Supervised learning

We evaluate our augmentation strategies on 12 standard binary classification tasks taken from the Microbiome Learning Repo [56]. These datasets comprise various disease and phenotype prediction tasks, including colorectal cancer [32], type 2 diabetes [44], Crohn's disease [18], and cirrhosis [45], as well as multiple body sites including the gut, skin, oral cavity, airways, and vagina [37, 49]. As such, this benchmark provides a comprehensive evaluation for predictive models of the human microbiome [58, 46, 24]. More details on these 12 learning tasks can be found in Table 1; note this benchmark is constructed from the original Microbiome Learning Repo by filtering datasets that contain a minimum sample size of 100, with at least 50 in either class.

For each learning task, we take 20 independent 80/20 train/test splits and we fit Random Forest, XGBoost, mAML [58], DeepCoDa [46], and MetaNN [36], first to the original training data, then on 3 augmented training sets obtained using our 3 augmentation strategies. Thus, we train a total of

Table 1: Evaluation benchmark consisting of 12 binary classification tasks taken from the Microbiome Learning Repo [56], after filtering to datasets containing at least 100 samples with at least 50 in each class. For each task we show the number of samples (n), the number of features (p), a description of the two classes and the number of samples in each, together with a reference to the original studies that each dataset was obtained from.

Task	n	p	Class 1 / Class 2	# in 1	# in 2	Reference
1	140	992	Crohn's disease / Without (ileum)	78	62	[18]
2	160	992	Crohn's disease / Without (rectum)	68	92	[18]
3	2070	3090	Gastrointestinal tract / Oral	227	1843	[37]
4	180	3090	Female / Male	82	98	[37]
5	404	3090	Stool / Tongue (dorsum)	204	200	[37]
6	408	3090	Subgingival / Supragingival plaque	203	205	[37]
7	172	980	Healthy / Colorectal cancer	86	86	[32]
8	124	2526	Diabetes / Without	65	59	[44]
9	130	2579	Cirrhosis / Without	68	62	[45]
10	199	660	Black / Hispanic	104	95	[49]
11	342	660	Nugent score high / Low	97	245	[49]
12	200	660	Black / White	104	96	[49]

Table 2: Data augmentation performance for Aitchison Mixup. We show the test AUC, averaged over 20 train/test bootstraps, for each learning task and predictive model, trained with and without data augmentation. Bold numbers indicate whether the version with or without augmentation performed best. Underlined numbers indicate the overall best model for that task. Models trained with Aitchison Mixup consistently outperformed those without (with the possible exception of MetaNN, which performed worst across the board). Error bars are given in Appendix A.

Task	RF	Aug	XGB	Aug	mAML	Aug	DeepCoDa	Aug	MetaNN	Aug
1	0.72	**0.79**	0.76	**0.79**	0.72	**0.74**	0.73	**0.79**	**0.74**	**0.74**
2	0.78	**0.82**	**0.81**	0.80	**0.80**	**0.80**	0.78	**0.83**	**0.74**	**0.74**
3	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
4	0.60	**0.64**	**0.57**	**0.57**	0.56	**0.58**	**0.58**	**0.58**	0.50	**0.51**
5	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
6	0.81	**0.83**	0.82	**0.83**	**0.84**	0.83	0.78	**0.82**	0.75	**0.76**
7	**0.68**	0.67	0.67	**0.69**	0.73	**0.74**	0.63	**0.73**	**0.59**	0.54
8	0.62	**0.65**	0.66	**0.68**	0.64	**0.65**	0.45	**0.70**	**0.64**	**0.64**
9	**0.93**	**0.93**	0.94	**0.95**	0.92	**0.93**	0.84	**0.90**	0.76	**0.82**
10	0.53	**0.60**	0.57	**0.61**	0.61	**0.62**	0.62	**0.63**	**0.63**	0.61
11	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.96**	0.95
12	0.55	**0.61**	0.58	**0.65**	**0.61**	**0.61**	**0.66**	0.65	0.58	**0.60**
Mean	0.77	**0.79**	0.78	**0.80**	0.78	**0.79**	0.75	**0.80**	**0.74**	**0.74**

$12 \times 20 \times 5 \times 4 = 4\,800$ models.[2] We evaluate test performance using ROC AUC, and we note that our datasets do not present severe class imbalance (with the exception of tasks 3 and 11, both of which were well separated by all our classifiers and therefore had no effect on the overall comparison).

For the augmented training sets, we generated 10 times as many synthetic samples as there were original training examples. The factor of 10 was chosen so as to obtain a relatively large augmented sample, in order to reduce the sampling variance from our random augmentations. In turn, we compensate for the fact that our augmented data is then much more numerous than our original training data, by downweighting the synthetic samples by a factor of 10; the total weight of the original and synthetic data is then equal to $1/2$ each.

Our results are shown in Tables 2 (Aitchison Mixup), 3 (Random Subcompositions), and 4 (Compositional CutMix):

[2]We train these models in parallel on a CPU cluster.

Table 3: Data augmentation performance for Random Subcompositions, similar to Table 2. Training sets augmented with Random Subcompositions consistently performed better than those without.

Task	RF	Aug	XGB	Aug	mAML	Aug	DeepCoDa	Aug	MetaNN	Aug
1	0.72	**0.78**	0.76	**0.77**	**0.72**	**0.72**	0.73	**0.78**	0.74	**0.76**
2	0.78	**0.81**	**0.81**	**0.81**	**0.80**	**0.80**	0.78	**0.85**	0.74	**0.76**
3	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
4	0.60	**0.62**	**0.57**	**0.57**	**0.56**	**0.56**	**0.58**	0.52	**0.50**	**0.50**
5	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
6	0.81	**0.82**	0.82	**0.83**	**0.84**	0.83	0.78	**0.82**	0.75	**0.78**
7	**0.68**	**0.68**	**0.67**	**0.67**	0.73	**0.74**	0.63	**0.74**	**0.59**	0.53
8	0.62	**0.64**	0.66	**0.69**	**0.64**	0.63	0.45	**0.61**	0.64	**0.65**
9	**0.93**	0.92	**0.94**	**0.94**	**0.92**	**0.92**	0.84	**0.90**	0.76	**0.82**
10	0.53	**0.58**	0.57	**0.61**	**0.61**	**0.61**	**0.62**	0.55	0.62	**0.63**
11	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	0.98	**0.99**	**0.96**	**0.96**
12	0.55	**0.60**	0.58	**0.63**	**0.61**	**0.61**	**0.66**	0.65	0.58	**0.59**
Mean	0.77	**0.79**	0.78	**0.79**	**0.78**	**0.78**	0.75	**0.78**	0.74	**0.75**

Table 4: Data augmentation performance for Compositional CutMix, similar to Table 2. Training sets augmented with Compositional CutMix consistently enjoyed better test performance than those without, for all of our predictive models. Note that models trained with this data augmentation set a new state-of-the-art on 8 out of 12 tasks, including disease prediction for colorectal cancer, type 2 diabetes, and Crohn's disease.

Task	RF	Aug	XGB	Aug	mAML	Aug	DeepCoDa	Aug	MetaNN	Aug
1	0.72	**0.78**	0.76	**0.77**	0.72	**0.74**	0.73	**0.79**	**0.74**	**0.74**
2	0.78	**0.81**	0.81	**0.82**	0.80	**0.81**	0.78	**0.83**	0.74	**0.77**
3	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
4	0.60	**0.65**	0.57	**0.59**	0.56	**0.59**	**0.58**	0.57	**0.50**	**0.50**
5	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
6	0.81	**0.82**	0.82	**0.83**	**0.84**	**0.84**	0.78	**0.82**	0.75	**0.78**
7	0.68	**0.69**	0.67	**0.68**	**0.73**	0.72	0.63	**0.76**	**0.59**	0.55
8	0.62	**0.66**	0.66	**0.72**	**0.64**	**0.64**	0.45	**0.69**	0.64	**0.65**
9	**0.93**	**0.93**	0.94	**0.95**	0.92	**0.93**	0.84	**0.91**	0.76	**0.81**
10	0.53	**0.57**	0.57	**0.59**	**0.61**	**0.61**	**0.62**	0.61	0.62	**0.65**
11	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	0.98	**0.99**	**0.96**	**0.96**
12	0.55	**0.57**	0.58	**0.63**	**0.61**	**0.61**	**0.66**	0.65	0.58	**0.61**
Mean	0.77	**0.79**	0.78	**0.80**	0.78	**0.79**	0.75	**0.80**	0.74	**0.75**

- Table 2 shows that Aitchison Mixup improved the test performance of existing models across a large majority of learning tasks. On average, we obtained a 5% gain in test AUC for DeepCoDa, 2% for random forests and XGBoost, and 1% for mAML. MetaNN remained flat, but this was also the worst performing model overall. Importantly, in the few instances where Aitchison Mixup hurt the performance of a model, the loss was typically of just 1%.

- Table 3 shows that Random Subcompositions also result in consistent performance improvements across most models and tasks, though to a slightly lesser degree than Aitchison Mixup. On average, DeepCoDa enjoyed a 3% gain in test AUC, random forests 2%, and XGBoost and MetaNN both 1% (mAML remained flat).

- Table 4 shows that Compositional CutMix obtained the strongest classification performance out of our 3 augmentation strategies. On average, DeepCoDa saw a 5% gain in test AUC, with 2% for random forests and XGBoost, and 1% for mAML and MetaNN. Moreover, models trained with Compositional CutMix set a new state-of-the-art on 8 out of 12 tasks, including colorectal cancer, type 2 diabetes, and Crohn's disease. Of the remaining 4 tasks, 3 were tied and only on 1 was the best model one that did not use Compositional CutMix.

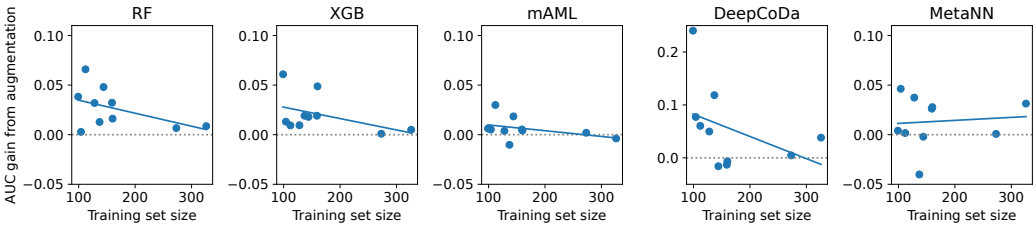

Figure 2: Difference in test AUC between models trained with Compositional CutMix, and those without, shown as a function of training set size. Each point represents one of our 12 benchmark learning tasks (note that tasks 3 and 5 are removed from the plot, since they already enjoyed 100% classification accuracy prior to applying any data augmentation). The linear trendlines show a somewhat greater outperformance on the smaller datasets relative to the larger datasets.

Figure 2 shows the effect of data augmentation on test performance as a function of the number of samples in the training set, for Compositional CutMix. We note that datasets with smaller sample sizes tended to benefit more from data augmentation. This suggests that our methodology may prove beneficial to many microbiome research projects, where datasets with low-hundreds of samples are commonplace. Figure 3 in Appendix A shows a similar scatterplot, as a function of the dataset dimensionality rather than sample size; the gains from augmentation appear consistent across lower- and higher-dimensional microbiome CoDa.

In addition, we evaluated the impact of our data augmentations on the expected calibration error (ECE) [25] of our models. Previous works have noted that increasingly flexible predictive models such as deep neural networks, while enabling greater predictive accuracy, tend to become overconfident in their predictions, degrading uncertainty quantification [25]. In Appendix A, we show the ECE obtained by our models, and we verify that our data augmentations do not hurt model calibration overall (in fact, some modest improvements are obtained).

4.2 Contrastive representation learning

We have shown that our novel data augmentation strategies provide consistent performance gains across supervised training pipelines. Next, we take our data augmentations one step further to provide the first application of contrastive representation learning to microbiome CoDa.

As a point of comparison, we use DeepMicro [40], a deep representation learning architecture tuned specifically for microbiome data. DeepMicro trains a deep autoencoder to learn a low-dimensional representation of the microbial composition. This learned representation is fed to a classification head downstream. As per the author's implementation, the encoder has 2 hidden layers with 256 and 128 units respectively, and ReLU nonlinearities, and a 64-dimensional output. The decoder architecture is a mirror image of the encoder. The decoder output is of the same dimension as the encoder input, and the model is trained to minimize the mean squared error between the two, using the Adam optimizer with default parameters for 2,000 epochs.

To ensure a fair comparison, our contrastive model uses the same encoder architecture from DeepMicro. However, we can discard the decoder layers; instead, the 64-dimensional latent representation is passed to a multilayer projection head that is trained jointly with the encoder using a contrastive loss, as is done in SimCLR [7]. Our projection head contains one 32-dimensional hidden layer with ReLU activations, and a 16-dimensional projection output, which is normalized to 1 unit in L2 norm and passed to a temperature-scaled cross-entropy loss. Intuitively, this loss function is designed to draw the representations of positive pairs of examples close together, and push those of negative pairs far apart. Positive pairs refer to two synthetic samples generated by random data augmentations of the same training example; negative pairs refer to synthetic samples generated from different training examples. In our implementation, at each training step we sample two Random Subcompositions from each training example, and compute the contrastive loss over all such pairs. In particular, our batch size corresponds to the entire training set, which is reasonable given the small sample sizes in our data. We again used the Adam optimizer with default parameters for 2,000 epochs.

Table 5: Representation learning performance for DeepMicro and contrastive learning (ours). For each task, we show the test AUC, averaged over 20 train/test bootstraps, under the linear evaluation protocol and finetuning. The representations obtained via contrastive learning consistently achieve higher AUC than those learned by DeepMicro. In addition, a randomly initialized encoder network of the same architecture is shown for comparison (No pretrain). Error bars are given in Appendix A.

	Linear Evaluation			Finetuning		
Task	No pretrain	DeepMicro	Contrastive	No pretrain	DeepMicro	Contrastive
1	0.59	0.68	**0.76**	0.72	0.75	**0.77**
2	0.67	0.76	**0.80**	0.76	0.76	**0.79**
3	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
4	0.47	0.53	**0.59**	0.50	0.53	**0.55**
5	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
6	0.75	0.76	**0.77**	**0.77**	0.76	**0.77**
7	0.59	**0.68**	0.63	**0.59**	**0.59**	0.57
8	0.65	**0.66**	**0.66**	0.66	**0.68**	**0.68**
9	0.72	0.71	**0.86**	0.76	0.77	**0.82**
10	0.53	0.58	**0.62**	0.60	0.61	**0.63**
11	0.96	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**
12	0.66	**0.68**	0.66	0.62	**0.64**	**0.64**
Mean	0.72	0.75	**0.78**	0.74	0.76	**0.77**

We evaluate the quality of our learned representations under two standard protocols. First, linear evaluation, where the encoder weights are frozen and a linear classification head is trained using the supervised cross-entropy loss. Second, finetuning, where the linear classification head is trained jointly with the encoder network. These evaluation protocols are applied for both DeepMicro and our contrastive model, in addition to a "no pretraining" control. This control is a randomly initialized encoder network and linear head with the same architecture, trained only on the supervised objective.

The results are shown in Figure 5; the representations learned by our contrastive model consistently outperform those learned by DeepMicro, both in linear evaluation and finetuning. Importantly, note that the comparison to DeepMicro is conservative, in the sense that we replicated the architecture and simply changed the pretraining objective from reconstruction error to contrastive loss; finetuning the encoder architecture itself under a contrastive objective would likely lead to further performance improvements. We conclude that Random Subcompositions provide a valuable data augmentation for contrastive learning on the simplex.

Note that our other 2 augmentation strategies, Aitchison Mixup and Compositional CutMix, may also be used for contrastive learning. However, these augmentations require paired training examples to generate each synthetic sample, and the implementation is therefore slightly more involved. In particular, we generate each pair of positive samples by taking one pair of training examples, and drawing two random combinations (in the Aitchison Mixup or the Compositional CutMix sense) of that one pair. The training examples are randomly partitioned into pairs at each epoch, and negative samples correspond to those that originate from disjoint pairs. Note that we no longer require that the pairs are drawn from the same class for the contrastive objective. The results are provided in Appendix A and show similarly strong performance to Table 5.

5 Conclusion

By combining ideas from data augmentation with the principles of CoDa, we have defined 3 novel augmentation strategies for CoDa: Aitchison Mixup, Random Subcompositions, and Compositional CutMix. Our augmentations offer cheap performance gains across a wide range of benchmark datasets, advancing the state-of-the-art on standard classification benchmarks for colorectal cancer, type 2 diabetes, and Crohn's disease. Our data augmentations can also be used to define an effective contrastive loss, which improves on existing representation learning frameworks for microbiome data. Data augmentation and contrastive learning have enjoyed tremendous success in other application domains such as computer vision; our novel methodology aims to help spur similar developments in the fields of CoDa and the microbiome.

References

[1] J. Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):139–160, 1982.

[2] M. Avalos, R. Nock, C. S. Ong, J. Rouar, and K. Sun. Representation learning of compositional data. *Advances in Neural Information Processing Systems*, 31, 2018.

[3] A. Buccianti and E. Grunsky. Compositional data analysis in geochemistry: Are we sure to see what really occurs during natural processes?, 2014.

[4] A. Buccianti, G. Mateu-Figueras, and V. Pawlowsky-Glahn. Compositional data analysis in the geosciences: from theory to practice. Geological Society of London, 2006.

[5] G. Cammarota, G. Ianiro, A. Ahern, C. Carbone, A. Temko, M. J. Claesson, A. Gasbarrini, and G. Tortora. Gut microbiome, big data and machine learning to promote precision medicine for cancer. *Nature reviews gastroenterology & hepatology*, 17(10):635–648, 2020.

[6] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[7] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[8] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[9] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[12] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in neural information processing systems*, 27, 2014.

[13] J. C. Douma and J. T. Weedon. Analysing continuous proportions in ecology and evolution: A practical introduction to beta and dirichlet regression. *Methods in Ecology and Evolution*, 10(9):1412–1430, 2019.

[14] P. Favoriti, G. Carbone, M. Greco, F. Pirozzi, R. E. M. Pirozzi, and F. Corcione. Worldwide burden of colorectal cancer: a review. *Updates in surgery*, 68(1):7–11, 2016.

[15] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*, 2021.

[16] P. Filzmoser, K. Hron, and M. Templ. Applied compositional data analysis. *Cham: Springer*, 2018.

[17] J. M. Fry, T. R. Fry, and K. R. McLaren. Compositional data analysis and zeros in micro data. *Applied Economics*, 32(8):953–959, 2000.

[18] D. Gevers, S. Kugathasan, L. A. Denson, Y. Vázquez-Baeza, W. Van Treuren, B. Ren, E. Schwager, D. Knights, S. J. Song, M. Yassour, et al. The treatment-naive microbiome in new-onset crohn's disease. *Cell host & microbe*, 15(3):382–392, 2014.

[19] J. A. Gilbert, M. J. Blaser, J. G. Caporaso, J. K. Jansson, S. V. Lynch, and R. Knight. Current understanding of the human microbiome. *Nature medicine*, 24(4):392–400, 2018.

[20] G. B. Gloor, J. M. Macklaim, V. Pawlowsky-Glahn, and J. J. Egozcue. Microbiome datasets are compositional: and this is not optional. *Frontiers in microbiology*, 8:2224, 2017.

[21] G. B. Gloor and G. Reid. Compositional analysis: a valid approach to analyze microbiome high-throughput sequencing data. *Canadian journal of microbiology*, 62(8):692–703, 2016.

[22] E. Gordon-Rodriguez, G. Loaiza-Ganem, and J. Cunningham. The continuous categorical: a novel simplex-valued exponential family. In *International Conference on Machine Learning*, pages 3637–3647. PMLR, 2020.

[23] E. Gordon-Rodriguez, G. Loaiza-Ganem, G. Pleiss, and J. P. Cunningham. Uses and abuses of the cross-entropy loss: Case studies in modern deep learning. In *Proceedings on "I Can't Believe It's Not Better!" at NeurIPS Workshops*, volume 137 of *Proceedings of Machine Learning Research*, pages 1–10. PMLR, 2020.

[24] E. Gordon-Rodriguez, T. P. Quinn, and J. P. Cunningham. Learning sparse log-ratios for high-throughput sequencing data. *Bioinformatics*, 38(1):157–163, 2022.

[25] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[26] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[27] A. V. Hartstra, K. E. Bouter, F. Bäckhed, and M. Nieuwdorp. Insights into the role of the microbiome in obesity and type 2 diabetes. *Diabetes care*, 38(1):159–165, 2015.

[28] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[29] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[30] X. He, K. Zhao, and X. Chu. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, 2021.

[31] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

[32] A. D. Kostic, D. Gevers, C. S. Pedamallu, M. Michaud, F. Duke, A. M. Earl, A. I. Ojesina, J. Jung, A. J. Bass, J. Tabernero, et al. Genomic analysis identifies association of fusobacterium with colorectal carcinoma. *Genome research*, 22(2):292–298, 2012.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[34] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[35] Y.-X. Liu, Y. Qin, T. Chen, M. Lu, X. Qian, X. Guo, and Y. Bai. A practical guide to amplicon and metagenomic analysis of microbiome data. *Protein & cell*, 12(5):315–330, 2021.

[36] C. Lo and R. Marculescu. Metann: accurate classification of host phenotypes from metagenomic data using neural networks. *Bmc Bioinformatics*, 20(12):1–14, 2019.

[37] B. A. Methé, K. E. Nelson, M. Pop, H. H. Creasy, M. G. Giglio, C. Huttenhower, D. Gevers, J. F. Petrosino, S. Abubucker, J. H. Badger, et al. A framework for human microbiome research. *nature*, 486(7402):215, 2012.

[38] M. Mulenga, S. A. Kareem, A. Q. M. Sabri, M. Seera, S. Govind, C. Samudi, and S. B. Mohamad. Feature extension of gut microbiome data for deep neural network-based colorectal cancer classification. *IEEE Access*, 9:23565–23578, 2021.

[39] J. H. Na, M. D. Demetriou, M. Floyd, A. Hoff, G. R. Garrett, and W. L. Johnson. Compositional landscape for glass formation in metal alloys. *Proceedings of the National Academy of Sciences*, 111(25):9031–9036, 2014.

[40] M. Oh and L. Zhang. Deepmicro: deep representation learning for disease prediction based on microbiome data. *Scientific reports*, 10(1):1–9, 2020.

[41] V. Pawlowsky-Glahn and A. Buccianti. *Compositional data analysis: Theory and applications*. John Wiley & Sons, 2011.

[42] V. Pawlowsky-Glahn, J. J. Egozcue, and R. Tolosana Delgado. Lecture notes on compositional data analysis. 2007.

[43] L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[44] J. Qin, Y. Li, Z. Cai, S. Li, J. Zhu, F. Zhang, S. Liang, W. Zhang, Y. Guan, D. Shen, et al. A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature*, 490(7418):55–60, 2012.

[45] N. Qin, F. Yang, A. Li, E. Prifti, Y. Chen, L. Shao, J. Guo, E. Le Chatelier, J. Yao, L. Wu, et al. Alterations of the human gut microbiome in liver cirrhosis. *Nature*, 513(7516):59–64, 2014.

[46] T. Quinn, D. Nguyen, S. Rana, S. Gupta, and S. Venkatesh. Deepcoda: personalized interpretability for compositional health data. In *International Conference on Machine Learning*, pages 7877–7886. PMLR, 2020.

[47] T. P. Quinn, I. Erb, M. F. Richardson, and T. M. Crowley. Understanding sequencing data as compositions: an outlook and review. *Bioinformatics*, 34(16):2870–2878, 2018.

[48] T. P. Quinn, E. Gordon-Rodriguez, and I. Erb. A critique of differential abundance analysis, and advocacy for an alternative. *arXiv preprint arXiv:2104.07266*, 2021.

[49] J. Ravel, P. Gajer, Z. Abdo, G. M. Schneider, S. S. Koenig, S. L. McCulle, S. Karlebach, R. Gorle, J. Russell, C. O. Tacket, et al. Vaginal microbiome of reproductive-age women. *Proceedings of the National Academy of Sciences*, 108(Supplement 1):4680–4687, 2011.

[50] R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*, 2015.

[51] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[52] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[53] S. Tarallo, G. Ferrero, G. Gallo, A. Francavilla, G. Clerico, A. Realis Luc, P. Manghi, A. M. Thomas, P. Vineis, N. Segata, et al. Altered fecal small rna profiles in colorectal cancer reflect gut microbiome composition in stool samples. *Msystems*, 4(5):e00289–19, 2019.

[54] B. D. Topçuoğlu, N. A. Lesniak, M. T. Ruffin IV, J. Wiens, and P. D. Schloss. A framework for effective application of machine learning to microbiome-based classification problems. *MBio*, 11(3):e00434–20, 2020.

[55] P. J. Turnbaugh, R. E. Ley, M. Hamady, C. M. Fraser-Liggett, R. Knight, and J. I. Gordon. The human microbiome project. *Nature*, 449(7164):804–810, 2007.

[56] P. Vangay, B. M. Hillmann, and D. Knights. Microbiome learning repo (ml repo): A public repository of microbiome regression and classification tasks. *Gigascience*, 8(5):giz042, 2019.

[57] E. K. Wright, M. A. Kamm, S. M. Teo, M. Inouye, J. Wagner, and C. D. Kirkwood. Recent advances in characterizing the gastrointestinal microbiome in crohn's disease: a systematic review. *Inflammatory bowel diseases*, 21(6):1219–1228, 2015.

[58] F. Yang and Q. Zou. maml: an automated machine learning pipeline with a microbiome repository for human disease classification. *Database*, 2020, 2020.

[59] J. Yoon, Y. Zhang, J. Jordon, and M. van der Schaar. Vime: Extending the success of self- and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33:11033–11043, 2020.

[60] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[61] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Compare Abstract/Intro to Methods/Experiments
   (b) Did you describe the limitations of your work? [Yes] See Section 3.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Not the focus of our paper, but briefly discussed model calibration in Section 4.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] No theoretical results.
   (b) Did you include complete proofs of all theoretical results? [N/A] No theoretical results.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 or code in supplemental material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See appendix A.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.
   (b) Did you mention the license of the assets? [No] Assets are open-source.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code provided in supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] Datasets used are in the public domain.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] No identifiable or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] Not used.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Not used.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Not used.

A Additional experimental results

In Figure 3, we show the gain in test AUC obtained from data augmentation (Compositional CutMix), plotted as a function of the dimensionality of the dataset. Note that the performance gains from augmentation are broadly consistent across higher- and lower-dimensional datasets. Tables 6, 7, and 8 show the effect on expected calibration error (ECE) of adding data augmentation to our various models. Notice that overall calibration error stays the same or improves slightly after incorporating data augmentation. Tables 11, 12, and 13, show error bars for Tables 2, 3, and 4, respectively.

In Tables 9 and 10, we show the performance of our contrastive models defined with Aitchison Mixup and Compositional CutMix, respectively. The latter performs equally strongly to Random Subcompositions, as shown in Section 4.2, the former performs noticeably worse, however still no worse than DeepMicro, thus demonstrating the robustness of our contrastive framework to multiple augmentation strategies. Further gains may be obtained by combining augmentations in a contrastive loss, a direction which is left to future work. Table 5 shows error bars for Table 14, respectively.

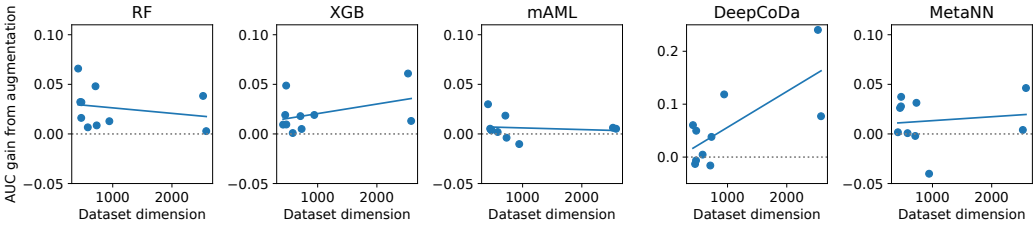

Figure 3: Similar to Figure 2. Difference in test AUC between models trained with Compositional CutMix, and those without, shown as a function of dataset dimension.

Table 6: Augmentation performance for Aitchison Mixup. We show the test ECE, averaged over 20 train/test bootstraps, for each learning task and predictive model, trained with and without data augmentation.

Task	RF	Aug	XGB	Aug	mAML	Aug	DeepCoDa	Aug	MetaNN	Aug
1	0.13	0.13	0.20	0.22	0.21	0.18	0.15	0.26	0.30	0.31
2	0.14	0.14	0.19	0.19	0.15	0.18	0.18	0.21	0.31	0.31
3	0.03	0.02	0.00	0.00	0.02	0.02	0.01	0.00	0.00	0.00
4	0.10	0.08	0.27	0.29	0.23	0.19	0.15	0.27	0.46	0.47
5	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.11	0.11	0.13	0.15	0.10	0.11	0.14	0.12	0.26	0.26
7	0.13	0.13	0.23	0.24	0.16	0.18	0.20	0.22	0.40	0.45
8	0.08	0.12	0.24	0.25	0.20	0.19	0.43	0.23	0.38	0.38
9	0.15	0.13	0.11	0.11	0.13	0.13	0.20	0.16	0.28	0.23
10	0.16	0.13	0.31	0.32	0.22	0.22	0.13	0.12	0.25	0.28
11	0.07	0.08	0.04	0.04	0.07	0.08	0.07	0.06	0.05	0.06
12	0.18	0.14	0.31	0.28	0.15	0.22	0.09	0.13	0.24	0.27
Mean	0.11	0.10	0.17	0.17	0.14	0.14	0.15	0.15	0.24	0.24

B Other augmentation strategies

We propose an additional augmentation strategy, *Multinomial Resampling*, which can be thought of as a CoDa analogue of image blurring. The goal of this augmentation is to inject noise across the coordinates of $\mathbf{x}$ in a principled manner. Additive Gaussian noise is clearly ill-suited for CoDa, as it is not constrained to the simplex. Instead, inspired by the recording mechanism for microbiome CoDa, we propose generating new datapoints from a multinomial distribution. Note that high-throughput sequencing technologies record the microbial composition present in a specimen by subsampling the larger population. The total number of reads in this subsample, known as the sequencing depth, is an artifact of the measurement process. Assuming the subsample is small relative to the population,

Table 7: Augmentation performance for Random Subcompositions. We show the test ECE, averaged over 20 train/test bootstraps, for each learning task and predictive model, trained with and without data augmentation.

Task	RF	Aug	XGB	Aug	mAML	Aug	DeepCoDa	Aug	MetaNN	Aug
1	0.13	0.13	0.20	0.22	0.21	0.21	0.15	0.24	0.30	0.26
2	0.14	0.13	0.19	0.16	0.15	0.15	0.18	0.13	0.31	0.27
3	0.03	0.01	0.00	0.00	0.02	0.02	0.01	0.00	0.00	0.00
4	0.10	0.10	0.27	0.27	0.23	0.21	0.15	0.12	0.46	0.46
5	0.02	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00
6	0.11	0.10	0.13	0.13	0.10	0.11	0.14	0.13	0.26	0.24
7	0.13	0.11	0.23	0.24	0.16	0.17	0.20	0.12	0.40	0.44
8	0.08	0.09	0.24	0.26	0.20	0.19	0.43	0.23	0.38	0.37
9	0.15	0.11	0.11	0.12	0.13	0.13	0.20	0.18	0.28	0.24
10	0.16	0.16	0.31	0.29	0.22	0.22	0.13	0.06	0.25	0.24
11	0.07	0.08	0.04	0.04	0.07	0.10	0.07	0.05	0.05	0.05
12	0.18	0.17	0.31	0.27	0.15	0.17	0.09	0.10	0.24	0.25
Mean	0.11	0.10	0.17	0.17	0.14	0.14	0.15	0.11	0.24	0.24

Table 8: Augmentation performance for Compositional CutMix. We show the test ECE, averaged over 20 train/test bootstraps, for each learning task and predictive model, trained with and without data augmentation.

Task	RF	Aug	XGB	Aug	mAML	Aug	DeepCoDa	Aug	MetaNN	Aug
1	0.13	0.13	0.20	0.22	0.21	0.19	0.15	0.24	0.30	0.29
2	0.14	0.13	0.19	0.19	0.15	0.17	0.18	0.17	0.31	0.29
3	0.03	0.01	0.00	0.00	0.02	0.02	0.01	0.00	0.00	0.00
4	0.10	0.10	0.27	0.27	0.23	0.15	0.15	0.25	0.46	0.48
5	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.11	0.09	0.13	0.16	0.10	0.12	0.14	0.12	0.26	0.24
7	0.13	0.14	0.23	0.24	0.16	0.16	0.20	0.18	0.40	0.44
8	0.08	0.11	0.24	0.23	0.20	0.22	0.43	0.27	0.38	0.38
9	0.15	0.11	0.11	0.10	0.13	0.12	0.20	0.15	0.28	0.25
10	0.16	0.16	0.31	0.31	0.22	0.19	0.13	0.13	0.25	0.26
11	0.07	0.06	0.04	0.04	0.07	0.10	0.07	0.05	0.05	0.06
12	0.18	0.18	0.31	0.29	0.15	0.19	0.09	0.12	0.24	0.25
Mean	0.11	0.10	0.17	0.17	0.14	0.14	0.15	0.14	0.24	0.24

the multinomial distribution provides a crude approximation for this generative process, with each trial being drawn according to the true proportions in the underlying population. In this way, we can sample new datapoints from a multinomial distribution where the number of trials corresponds to the sequencing depth. Namely, each new datapoint is generated as follows:

1. Draw a training point i uniformly at random.
2. Draw $\tilde{\mathbf{x}} \sim \text{Multinomial}(L_i, \mathbf{x}_i)$, where L_i is the sequencing depth (a.k.a. library size).
3. Set $\mathbf{x}^{\text{aug}} = \tilde{\mathbf{x}}/(\sum_{j=1}^{p} \tilde{x}_j)$ and $y^{\text{aug}} = y_i$.

Notice that the number of trials L_i controls the noise level; as $L_i \to \infty$, $\mathbf{x}^{\text{aug}} \to \mathbf{x}_i$. When applied to CoDa that does not arise from high-throughput sequencing, L_i can be specified arbitrarily, or treated as a hyperparameter. In our microbiome datasets, the sequencing depth is typically on the order of $L \sim 10\,000$.

Table 9: Similar to Table 5, but the contrastive model is now defined using Aitchison Mixup.

	Linear Evaluation			Finetuning		
Task	No pretrain	DeepMicro	Contrastive	No pretrain	DeepMicro	Contrastive
1	0.59	0.68	**0.72**	0.72	0.75	**0.77**
2	0.67	0.76	**0.77**	0.76	0.76	**0.77**
3	**1.00**	**1.00**	0.96	**1.00**	**1.00**	**1.00**
4	0.47	**0.53**	**0.53**	0.50	**0.53**	**0.53**
5	**1.00**	**1.00**	0.99	**1.00**	**1.00**	**1.00**
6	0.75	**0.76**	**0.76**	0.77	0.76	**0.78**
7	0.59	0.68	**0.75**	0.59	0.59	**0.65**
8	0.65	**0.66**	0.60	0.66	**0.68**	0.66
9	0.72	0.71	**0.77**	0.76	0.77	**0.79**
10	0.53	**0.58**	0.54	0.60	**0.61**	**0.61**
11	0.96	**0.98**	0.97	**0.98**	**0.98**	0.97
12	0.66	**0.68**	0.63	0.62	**0.64**	0.62
Mean	0.72	0.75	0.75	0.74	0.76	0.76

Table 10: Similar to Table 5, but the contrastive model is now defined using Compositional CutMix.

	Linear Evaluation			Finetuning		
Task	No pretrain	DeepMicro	Contrastive	No pretrain	DeepMicro	Contrastive
1	0.59	0.68	**0.76**	0.72	0.75	**0.76**
2	0.67	0.76	**0.80**	0.76	0.76	**0.77**
3	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
4	0.47	0.53	**0.57**	0.50	0.53	**0.54**
5	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
6	0.75	0.76	**0.80**	0.77	0.76	**0.78**
7	0.59	0.68	**0.72**	0.59	0.59	**0.61**
8	0.65	**0.66**	**0.66**	0.66	**0.68**	**0.68**
9	0.72	0.71	**0.85**	0.76	0.77	**0.81**
10	0.53	0.58	**0.63**	0.60	0.61	**0.63**
11	0.96	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**
12	0.66	**0.68**	0.64	0.62	**0.64**	0.63
Mean	0.72	0.75	**0.78**	0.74	0.76	**0.77**

Table 11: Error bars for Table 2.

Task	RF	Aug	XGB	Aug	mAML	Aug	DeepCoDa	Aug	MetaNN	Aug
1	0.02	0.01	0.01	0.01	0.02	0.02	0.01	0.01	0.02	0.02
2	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.02
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
7	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
8	0.02	0.03	0.02	0.02	0.03	0.02	0.02	0.02	0.03	0.03
9	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02
10	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01
12	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.02	0.02
Mean	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02

Table 12: Error bars for Table 3.

Task	RF	Aug	XGB	Aug	mAML	Aug	DeepCoDa	Aug	MetaNN	Aug
1	0.02	0.01	0.01	0.01	0.02	0.02	0.01	0.01	0.02	0.02
2	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.02
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
7	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
8	0.02	0.02	0.02	0.03	0.03	0.03	0.02	0.03	0.03	0.03
9	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02
10	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01
12	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Mean	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02

Table 13: Error bars for Table 4.

Task	RF	Aug	XGB	Aug	mAML	Aug	DeepCoDa	Aug	MetaNN	Aug
1	0.02	0.01	0.01	0.01	0.02	0.02	0.01	0.01	0.02	0.01
2	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.01
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
7	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
8	0.02	0.02	0.02	0.03	0.03	0.03	0.02	0.02	0.03	0.02
9	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.02	0.02
10	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01
12	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.02	0.02
Mean	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01

Table 14: Error bars for Table 5.

	Linear Evaluation			Finetuning		
Task	No pretrain	DeepMicro	Contrastive	No pretrain	DeepMicro	Contrastive
1	0.02	0.02	0.01	0.02	0.02	0.01
2	0.03	0.02	0.02	0.02	0.02	0.02
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.02	0.02	0.02	0.01	0.01	0.02
5	0.00	0.00	0.00	0.00	0.00	0.00
6	0.01	0.01	0.01	0.01	0.01	0.01
7	0.02	0.02	0.02	0.02	0.02	0.02
8	0.02	0.02	0.02	0.02	0.02	0.02
9	0.02	0.02	0.02	0.02	0.02	0.02
10	0.01	0.02	0.02	0.02	0.02	0.02
11	0.00	0.00	0.00	0.00	0.00	0.00
12	0.02	0.01	0.01	0.02	0.02	0.02
Mean	0.01	0.01	0.01	0.01	0.01	0.01