DNAGAST: Generative Adversarial Set Transformers for High-throughput Sequencing

David W. Ludwig II
Department of Computer Science
Middle Tennessee State University
Murfreesboro, TN. USA
dwl2x@mtmail.mtsu.edu

Joshua L. Phillips
Department of Computer Science
Middle Tennessee State University
Murfreesboro, TN. USA
joshua.phillips@mtsu.edu

Abstract

High-throughput sequencing (HTS) is a modern DNA sequencing technology used to rapidly read thousands of genomic fragments from microorganisms given a sample. The large amount of data produced by this process makes deep learning, whose performance often scales with dataset size, a suitable fit for processing HTS samples. While deep learning models have utilized sets of DNA sequences to make informed predictions, to our knowledge, there are no models in the current literature capable of generating synthetic HTS samples, a tool which could enable experimenters to predict HTS samples given some environmental parameters. Furthermore, the unordered nature of HTS samples poses a challenge to nearly all deep learning architectures because they have an inherent dependence on input order. To address this gap in the literature, we introduce DNA Generative Adversarial Set Transformer (DNAGAST), the first model capable of generating synthetic HTS samples. We qualitatively and quantitatively demonstrate DNAGAST's ability to produce realistic synthetic samples and explore various methods to mitigate mode-collapse. Additionally, we propose novel quantitative diversity metrics to measure the effects of mode-collapse for unstructured set-based data.

Introduction

Exploring and understanding functional roles in microbial communities is one of the most important research aspects of microbiology and bioinformatics. Using modern high-throughput sequencing (HTS) technologies, an experimenter can profile and analyze a microbiome from a sample by sequencing genomic material contained within it in the form of thousands of short DNA sequences. This allows the experimenter to identify what microorganisms are present and determine key factors that drive microbial communities. Currently, in order to determine what a microbiome would look like in a given scenario, one would have to actually perform the experiment. However, generative modeling methods using deep learning could be used to produce synthetic HTS data highly resembling what would actually be sequenced in the experiment, potentially expediting the research process.

Generative adversarial networks (GANs), one of the most powerful generative model architectures, have been used to generate synthetic DNA sequences independently with desired properties (Gupta and Zou [2019]). In a sample context, the ability to generate synthetic HTS samples with desired properties could be used to inform experimenters the important microbial interactions required to achieve such properties. Unlike DNA sequences that are structured by nature (i.e. the order of the base pairs matters), HTS samples are unstructured in that they are comprised of a set of DNA sequences with no inherent order. This poses a significant challenge to most deep learning methods as nearly all architectures have an inherent dependence on the input order. The usage of generative models such as GANs on unstructured/unordered data in the current literature is scarce and focuses primarily on 2D and 3D point cloud generation (Stelzner et al. [2020], Li et al. [2018]).

In this work, we present DNA Generative Adversarial Set Transformer (DNAGAST): the first generative model capable of generating synthetic HTS samples. We demonstrate that DNAGAST is capable of producing synthetic HTS samples that resemble their real counterparts. We also show that mode-collapse mitigation methodologies and techniques can be incorporated to improve sample diversity. Lastly, we present custom performance measures to quantitatively measure mode-collapse for unstructured set-based data.

Background

Generative Adversarial Networks

GANs by Goodfellow et al. [2014] are generative deep learning models that synthesize unique data that nevertheless resembles the source training data distribution. A typical GAN architecture consists of two models: the generator and the discriminator. The goal of the generator is to learn some distribution, p_g, over the given data, $\boldsymbol{x}$, allowing it to map a latent-space vector $p_{\boldsymbol{z}}(\boldsymbol{z})$ to the ambient-space to produce samples resembling the training data. The discriminator then predicts the probability that the given distribution comes from the data distribution, p_{data}, or the generated distribution, p_g, via another neural network, $D(\boldsymbol{x})$. The generator (G) and discriminator (D) are trained together in a minimax fashion, where the discriminator aims to maximize its performance in correct classification, while the genera-

tor tries to minimize its loss to fool the discriminator. This yields the adversarial loss function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}))] \tag{1}$$

The primary purpose of GANs and other generative models in the existing literature is image generation and generating structured data. They have been shown to be successful at generating DNA sequences with desired properties (Gupta and Zou [2019], Hazra et al. [2022]). However, the application of GANs for unstructured data is seemingly limited to low-dimensional geometric point clouds (Stelzner et al. [2020], Li et al. [2018]). This makes it unclear in the literature how one would go about generating other domains of unstructured data such as HTS.

Generative Adversarial Set Transformer

Generative Adversarial Set Transformer (GAST) by Stelzner et al. [2020] is a GAN architecture designed for unstructured data generation. It is built using components from the Set Transformer (ST) framework by Lee et al. [2019] which take advantage of the permutation-equivariant processing properties of the original transformer architecture by Vaswani et al. [2017]. While GAST's application was limited to 2D polygon and point cloud generation, the use of transformers is highly appealing as they are not only the state-of-the-art architecture across nearly every domain, but attention scores within the multi-head attention mechanism can be mined for explainable predictions (Dosovitskiy et al. [2021]).

Set Transformer Components ST defines the *multi-head attention block* (MAB) in terms of the standard transformer architecture without position encodings:

$$\begin{aligned} \text{MAB}(X,Y) &= \text{LN}(H + \text{FFN}(H)), \\ \text{where } H &= \text{LN}(X + \text{MHA}(X,Y,Y)) \end{aligned} \tag{2}$$

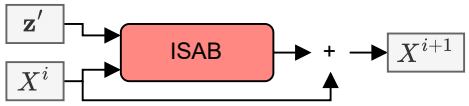
where $X, Y \in \mathbb{R}^{n\times d}$ are each a set of d-dimensional vectors, MHA is multi-head scaled dot-product attention as defined by Vaswani et al. [2017], FFN is any row-wise feed-forward network, and LN is layer normalization as proposed by Ba et al. [2016]. In order to process larger set sizes with linear time complexity, ST proposes the *induced set attention block* (ISAB) which utilizes a set of m inducing points, $I \in \mathbb{R}^{m\times d}$, where m is a hyperparameter specified by the experimenter. These inducing points can either be learned or predicted. The ISAB is defined as:

$$\begin{aligned} \text{ISAB}_m(X) &= \text{MAB}(X,H) \in \mathbb{R}^{n\times d} \\ \text{where } H &= \text{MAB}(I,X) \in \mathbb{R}^{m\times d} \end{aligned} \tag{3}$$

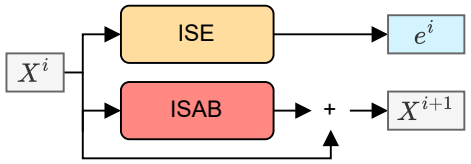
Finally, to expressively pool a set, GAST provides a modification of ST's *pooling by multi-head attention* component as shown below:

$$\text{ISE}_m(X) = \sum_{i=1}^{m} \text{MAB}(I_i, X) \in \mathbb{R}^{m\times d} \tag{4}$$

where $I \in \mathbb{R}^{m\times d}$ is a set of m inducing points.



(a) The generator Set Transformer block



(b) The discriminator Set Transformer block

Figure 1: The Set Transformer blocks for GAST.

Mitigating Mode Collapse

While GANs are capable of producing realistic results, they are notoriously difficult to train (Salimans et al. [2016],Srivastava et al. [2017],Arjovsky et al. [2017]. The most common issue to arise from training GANs is mode collapse where the model learns only a portion of the true data distribution, limiting the diversity of generated data. Though the solution to the issue of mode-collapse is an ongoing research problem, there are many techniques in the literature that can assist in its prevention. In this work, we focus on two prominent remedies, neither of which has to our knowledge yet been examined in the context of unstructured GAN models.

WGAN-GP Wasserstein GAN by Arjovsky et al. [2017] combined with gradient penalty (WGAN-GP) by Gulrajani et al. [2017] has been shown to significantly reduce mode-collapse and improve generated sample quality by replacing the discriminator with a critic network with a modified training objective to maximize confusion. The WGAN-GP objective is shown in the equation below.

$$\begin{aligned} \min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[D(G(\boldsymbol{z}))] \\ - \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[D(\boldsymbol{x})] \\ + \lambda \mathbb{E}_{\hat{\boldsymbol{x}}\sim p_{\hat{\boldsymbol{x}}}}[(\|\nabla_x D(\hat{\boldsymbol{x}})\|_2 - 1)^2] \end{aligned} \tag{5}$$

where λ is a fixed hyperparameter and $p_{\hat{\boldsymbol{x}}}$ is a uniformly-sampled set of interpolated data points acquired by linearly-interpolating uniformly between pairs of real and generated sample points.

VEEGAN VEEGAN by Srivastava et al. [2017] is a variant of the GAN architecture designed specifically to mitigate mode-collapse. This architecture introduces a 'reconstructor' component that aims to invert the generator, mapping data from the ambient space back to latent-space representations, forming an autoencoder with the generator and reconstructor. The loss function for this autoencoder is defined as:

$$\mathcal{L}_{recon} = \frac{1}{N} \sum_{i=1}^{N} d(z_i, z_i') \tag{6}$$

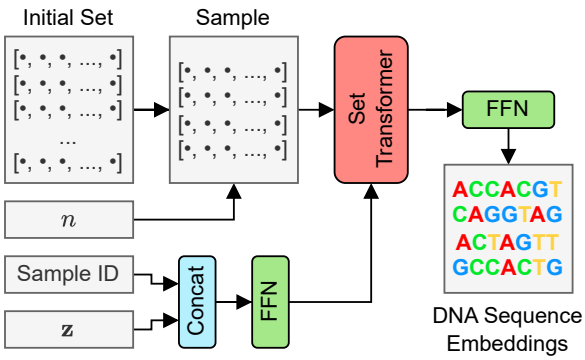


Figure 2: The generator architecture for DNAGAST.

where $\boldsymbol{z} \sim p_{\boldsymbol{z}}$, $\boldsymbol{z}'$ is the reconstructed version of $\boldsymbol{z}$ given its ambient-space mapping, N is the number of components in z/z', and d is any loss function such as mean-squared error. The discriminator is also modified to accept both the latent-space and ambient-space data representations (real and generated) as input. By enforcing the reconstructor's output distribution to be Gaussian with respect to both the real and generated data, the log-likelihood of the reconstructor's output can be computed and added to the generator's loss function. By doing so, the generator is better encouraged to generate diverse data across the whole range of the true data distribution.

Methods

High-throughput Sequencing Data

For this work, we utilize 210 raw high-throughput sequencing runs from soil samples obtained as part of a grassland restoration study by Barber et al. [2023]. Each run is comprised of sequences of 150 bp in length. This study took place at Nachusa Grasslands in northern Illinois as part of an ecosystem restoration project. Soil samples were collected from 13 restored prairies, 2 remnant prairies, and two agricultural fields rotated between corn and soy. Restored prairies were former agricultural fields planted with diverse seed mixes between 1987 and 2013, and remnants were prairies that were never converted to rowcrop agriculture. See Barber et al. [2023] for more details on sites and sampling.

DNA Sequence-level Embeddings

In order to embed individual DNA sequences, we employ a modified version of DNABERT by Ji et al. [2021], the current state-of-the-art model for DNA sequence embeddings as shown by Wang et al. [2023]. We simplify the DNABERT architecture by retaining only the special class and mask tokens as we are only interested in sequence embedding and reconstruction. Next, we modify the transformer blocks to use the pre-LN technique by Xiong et al. [2020] for more stable training. Lastly, the original DNABERT architecture employs absolute-position encodings, whereas we utilize relative position encodings as described by Shaw et al. [2018]. We hypothesize that using relative-position

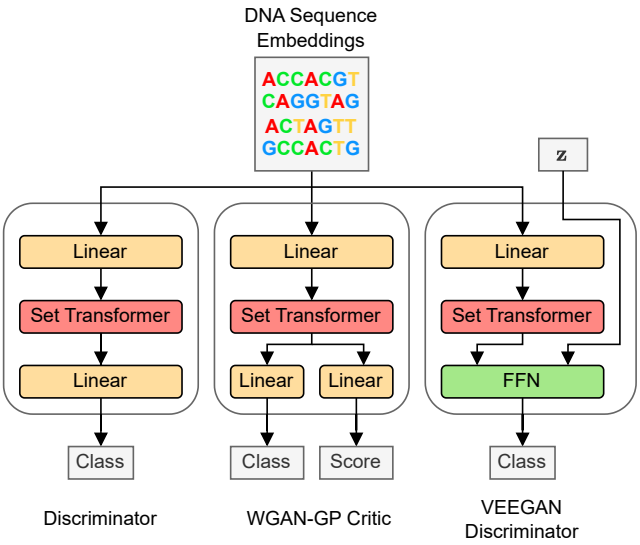


Figure 3: The primary discriminator architectures for DNAGAST.

encodings over absolute-position encodings result in higher quality sequence embeddings as the distance between nucleotides is more meaningful than their global positions in a given sequence. We pre-trained DNABERT on our high-throughput sequencing dataset following the same procedure outlined in the original DNABERT manuscript using 3-mer tokens and 8D embeddings. Each sequence is augmented by randomly trimming either end so that the final sequence length is 150 bp, and unknown/ambiguous bases are uniformly assigned a random concrete nucleotide base.

DNAGAST

We now introduce DNAGAST: the first generative model capable of synthesizing HTS samples. The goal of the DNAGAST generator is to produce subsamples of n sequences that effectively capture the taxonomy distribution of corresponding whole HTS samples from the training data. As mentioned previously, GAST and other GAN architectures that work with unstructured data are limited to low-dimensional point clouds. We exploit this fact by instead only requiring our generator to produce latent-space DNA sequence embeddings which is effectively a high-dimensional point cloud. These sequence embeddings can then be decoded later using any DNA sequence decoder model.

When designing this model, we utilize both the Set Transformer framework and the GAST framework as they have laid out most of the initial groundwork. As per the name, our models are highly-inspired by and build on top of the GAST framework; however, there are several modifications to the architectures and training regimes that we employ. First, as we are interested in generating specific samples, we modify the model to create an AC-GAN-like architecture and training regime. Next, we continue to employ the pre-LN technique in each of the transformer blocks follow-

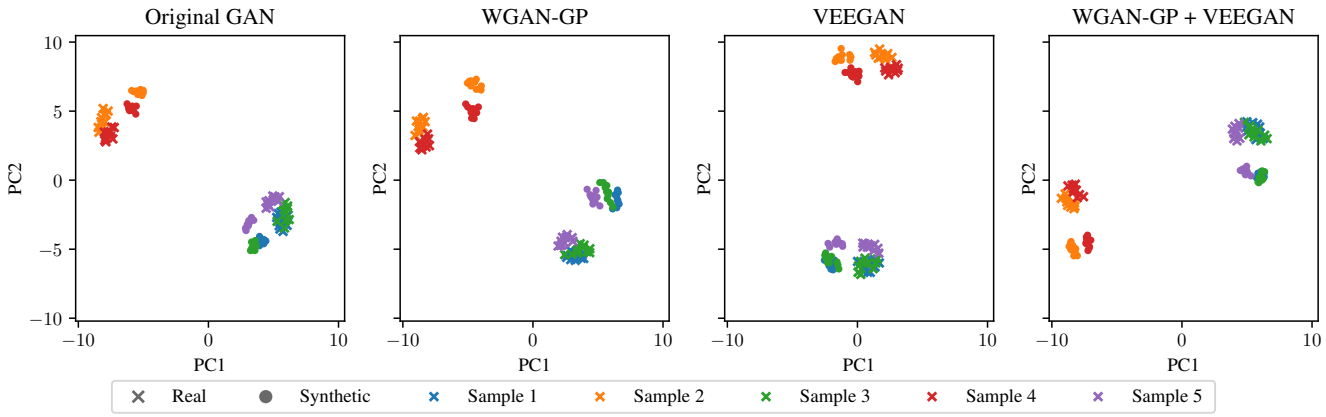


Figure 4: 2D metric-MDS projections of the real and synthetic sub-samples using Chamfer distances for each DNAGAST variant. Each point represents a subsample of 1,000 sequences from a corresponding sample. The Xs represent subsamples from actual data, and the circles represent synthetic subsamples from our models. The distance between two points indicates their similarity where points close together are more similar than points farther apart.

ing Xiong et al. [2020] for more stable training. We remove the spectral-normalization (SN) methods by Miyato et al. [2018] that the GAST framework implements as we found SN to negatively impact the quality of the generated samples. Lastly, we implement novel GAST architectures based on WGAN-GP and VEEGAN, as well as the combination of both, in order to combat mode collapse.

The Generator The generator architecture as portrayed in Figure 2. is very similar to the generator in the original GAST framework. To generate a set, it first produces a random initial set in the same manner as the original GAST generator, and selects n elements uniformly at random without replacement. This random set is then passed through a series of residually-connected ISABs where the set is conditioned on inducing points predicted by some feature vector. This feature vector is predicted using the provided latent-space vector. In order to implement conditional generation to produce specific sample types, our generator accepts a 'Sample ID' as additional input. These sample IDs can be provided in the form of one-hot encodings; however, we instead associate a learned embedding with each sample. The sample ID embedding and latent-space vectors are then concatenated together and passed through a feed-forward network to predict the feature vector. The output of the ISAB stack is then fed through a final feed-forward network where it is projected down to the sequence-level dimensionality, yielding the synthetic set of DNA sequence embeddings.

We use 64-dimensional vectors for both the sample ID and latent-space vectors. The initial set is randomly sampled in a high dimensional latent-space of 256 dimensions and fed through a block of 4 residually-connected ISABs. The ISABs also use 256-dimensional projections of the set elements when performing multi-head attention with 4 attention heads and 48 predicted inducing points. Each feed-forward network consists of small, two layer neural networks, and the ReLU activation function is employed for their inner layers. As mentioned in the background, unlike the original GAST architecture, we do not include the SN layers as we found them to negatively impact our models' performances.

The Discriminator Like the generator, our discriminator model for DNAGAST is also similar to the original GAST discriminator, continuing to employ the modifications as described previously for the generator. The standard discriminator is shown in Figure 3. It accepts a set of DNA sequence embeddings corresponding to a subsample as input and passes it through GAST's pooling blocks which produce a concatenated list of encoding vectors. These vectors are linearly-projected down to produce the probability distribution of the provided sample's label. The output layer consists of a single value corresponding to each sample label, along with an additional 'fake' class label. The discriminator then attempts to classify each sample as one of the real sample labels provided, otherwise it is simply considered fake.

We reuse many of the same hyperparameters from the generator for the discriminator. The set pooling blocks are comprised of 3 ISEs/ISABs, each using 256-dimensional projections, 4 attention heads and 48 learned inducing points. The ReLU activation function is also employed for intermediate layers. We tested SN solely on the discriminator during development as proposed by Miyato et al. [2018], however, we still observed inferior performance.

Mode-collapse Mitigating Methods During the development of these models, mode-collapse was a frequent issue that we encountered, making it difficult for the generator to produce diverse subsamples. In our efforts to resolve these issues, we investigated and combined multiple techniques that have been shown to prevent it. Specifically, we implemented both WGAN-GP and VEEGAN variations of DNAGAST, as well as the combination of the two methods which, to our knowledge, has not been done in the current literature.

We first look at the implementation of WGAN-GP, as it is rather simple to incorporate into our current DNAGAST

Model	Arithmetic	Geometric	Harmonic
DNAGAST	**2.964**	**2.961**	**2.959**
DNAGAST (WGAN-GP)	4.836	4.832	4.828
DNAGAST (VEEGAN)	3.040	3.037	3.035
DNAGAST (WGAN-GP + VEEGAN)	3.494	3.487	3.481

Table 1: The median of the computed mean real-to-fake chamfer distances across 10 independent evaluations.

model. As mentioned previously, WGAN-GP has been demonstrated to be generally superior to typical GAN architectures (Arjovsky et al. [2017], Gulrajani et al. [2017]). However, due to the nature of the critic's output, it is not immediately obvious how one can incorporate it with an AC-GAN like architecture where the discriminator produces a probability distribution of classes. In our model as shown in Figure 3, we give the critic two outputs: the critic score, and the predicted class probability distribution. The loss of this distribution can then be computed either by targeting the correct, real label, or target a 50/50 split between the real label and the fake label. While the 50/50 split is more inline with the WGAN-GP idea, we did not find any noticeable difference between the two methods in early testing, As a result, we simply compute the categorical cross-entropy of the correct label.

Alongside WGAN-GP, we also chose to implement a VEEGAN-based variant of the DNAGAST model. As described earlier, this necessitates a modification to the discriminator to include an additional input to represent the latent-space vector representation of the provided sample. This modified architecture is shown in Figure 3. We incorporate this into the discriminator's architecture by concatenating it with the already-concatenated list of encoding vectors before feeding it through the final feed-forward network. To create the reconstructor network, we use a nearly identical architecture to that of our original discriminator, only replacing the final linear projection with a feed-forward network to predict the latent-space vector of the provided sample. During training, we compute the loss of the reconstructor in much of the same manner as a typical autoencoder by computing the log-likelihood of the resulting latent-space vector as described by the original implementation from Srivastava et al. [2017]. The reconstructor is then trained in an autoencoder fashion with the generator, and the reconstructor's loss is added to the generator's loss.

Lastly, we design and implement a fourth architecture combining DNAGAST, WGAN-GP, and VEEGAN into a single model. Since both WGAN-GP and VEEGAN have been demonstrated to significantly reduce mode-collapse, we use this model to determine whether or not they are compatible with each other and capable of preventing mode-collapse even further.

Training

We train each of the GANs with slightly-modified versions of their corresponding defined training regimes. As we are using conditional GANs, we replace (include in the WGAN-based critics) the original discriminator's output with a categorical cross-entropy loss of the predicted label probability distribution. For each model, all components employ the Adam optimizer with a static learning rate of 0.0001. We then train each model for roughly 50,000 steps each.

In terms of the provided/generated data, a batch size of 16 subsamples is used, where each subsample consists of 1,000 sequences. Subsamples from the real data distribution are generated randomly and embedded using DNABERT on-the-fly during training. As two of our samples were too small, we generate subsamples uniformly from 208 different samples. A subsample is constructed by uniformly selecting 1,000 sequences at random without replacement from a particular sample. The individual sequences may be slightly longer than our desired 150-nucleotide length. When this occurs, the sequences are augmented by randomly truncating one or both sides to trim them down to 150 nucleotides.

Evaluation

Employing traditional sample comparison metrics would require reconstructing the DNA sequences from the embeddings and performing taxonomic analysis methods. In this work, we instead opt to make comparisons using the latent-space DNA sequence embeddings themselves. In order to make quantitative comparisons between samples then, we must utilize permutation-invariant metrics. For this work, we use Chamfer distance as shown in the formula below:

$$\mathrm{d_{CD}}(A, B) = \sum_{a \in A} \min_{b \in B} \|a - b\| + \sum_{b \in B} \min_{a \in A} \|b - a\| \quad (7)$$

where A and B are distinct sets (i.e. sets of DNA sequence embeddings). It is worth noting that we use L_1-norm to compute the distances between two vectors as evidence from Aggarwal et al. [2001] suggests that Manhattan distance can be more meaningful for high-dimensional data.

In order to provide a quantitative measure of mode collapse, we propose custom performance measures: intra-cluster diversity ratio and inter-cluster diversity ratio. Intra-cluster diversity is measured by computing the mean distance between all subsamples drawn from a particular sample which pertains to the intra-mode variance. Inter-cluster diversity is measured by computing the mean distance between all subsamples regardless of the originating sample, pertaining to the the inter-mode (co)variance. By computing these metrics for the real data and synthetic data independently, the real/synthetic ratio can be computed to obtain a quantitative diversity measure.

The source code for this work is publicly available on Github (`https://github.com/DLii-Research/dnagast`).

Model	Intra-cluster Diversity			Inter- cluster Diversity		
	Arithmetic	Geometric	Harmonic	Arithmetic	Geometric	Harmonic
DNAGAST	1.865	1.909	1.963	1.084	1.158	1.318
DNAGAST (WGAN-GP)	**1.263**	**1.284**	**1.306**	1.085	**1.028**	**1.013**
DNAGAST (VEEGAN)	1.516	1.535	1.551	1.040	1.097	1.198
DNAGAST (WGAN-GP + VEEGAN)	2.285	2.314	2.340	**1.028**	1.223	1.560

Table 2: The medians of the real/fake ratios for intra-cluster and inter-cluster diversities computing using arithmetic, geometric, and harmonic means across 10 independent evaluations.

Results

We first examine our generated subsamples by making qualitative comparisons using 2D MDS projections of the subsample Chamfer distances in Figure 4, where similarity is proportional to distance. Each point in the MDS plots represents a single subsample cluster of 1,000 sequences drawn from its corresponding sample. We find that all four models are able to learn the different modes of the data and produce synthetic samples resembling the real counterparts.

In order to obtain a quantitative measure on the sample quality, we compute the chamfer distance between real and synthetic subsamples for each of the 5 samples. For each sample, we obtain 10 real subsamples and 10 synthetic subsamples, compare all possible real-synthetic pairs, and then compute the arithmetic, geometric, and harmonic means of the resulting distances. Repeating this process across 10 independent evaluations, the median for each mean is obtained and shown in Table 1. We find that the standard DNAGAST implementation achieves the best result for this metric, with the VEEGAN variant following closely. The samples produced by the WGAN-GP variants were of significantly worse quality compared to the non-WGAN-GP variants.

We also evaluate subsample cluster diversity via intra-cluster and inter-cluster comparisons. Using these measurements, we can determine if any of our models are suffering from effects of mode-collapse. Starting with intra-cluster comparisons (i.e. subsamples compared only to subsamples that correspond to the same whole sample), we compute the intra-cluster diversity ratio across 10 independent evaluations and plot the median using arithmetic, geometric, and harmonic means in Table 2. We find that DNAGAST (WGAN-GP) produces significantly more diverse subsamples than the other architectures. We find that both the WGAN-GP and VEEGAN variants generally produce significantly more diverse subsamples than the original DNAGAST or the combined WGAN-GP + VEEGAN models as their ratios are closer to 1.0. While the WGAN-GP and VEEGAN variants are superior to DNAGAST in terms of sample diversity, it is surprising to find that the combination of the WGAN-GP and VEEGAN results in a significantly worse model. While further investigation is required to determine the exact cause for this unexpected behavior, this could be due to some incompatability between the WGAN-GP and VEEGAN architectures. Due to the chaotic nature of GANs, it could also be be that the combined WGAN-GP + VEEGAN variant requires significantly more training time.

Lastly, we analyze the inter-cluster diversity by comparing the ratio of real-to-real and fake-to-fake subsample distances across samples. We again perform 10 different comparisons using the arithmetic, geometric, and harmonic means of the chamfer distances for each model and list the values in Table 2. As with the intra-cluster diversity results, we find that the WGAN-GP and VEEGAN variants are both superior in terms of inter-cluster diversity compared to the standard DNAGAST model, and, while still successful, the combined model unexpectedly suffers the most.

Discussion

In this paper, we presented the first generative adversarial networks capable of synthesizing subsamples of unstructured HTS samples. Building off of the Set Transformer and Generative Adversarial Set Transformer frameworks, we develop four different models by incorporating modern GAN architectures and training techniques. While all four models learned to generate reasonably similar subsamples as evaluated using Chamfer distances, we found that the standard DNAGAST and VEEGAN models performed similarly in producing the highest quality subsamples. However, the intra/inter-cluster subsample diversity of the standard DNAGAST variant was significantly worse. It was found that the WGAN-GP variant of DNAGAST produced the most diverse subsamples, closely followed by the VEEGAN variant. Considering the trade-off of high subsample quality and subsample diversity, we believe that the VEEGAN variant of DNAGAST is the superior version.

While the integration of WGAN-GP/VEEGAN-like architectures into the DNAGAST models successfully improved the subsample diversity as expected, the combination of the methods resulted in a significantly worse performing model compared to the other three. As we are unaware of any incompatibility between the architectures, this issue could simply be an outlier. It is possible that by retraining with different initial weights, we may obtain an improved model. We plan to investigate this more in future work.

Though we believe this research lays the groundwork for developing powerful HTS generation models, there are still many questions that we wish to answer. First, GANs can map different characteristics in different regions of the latent-space. By creating general latent-space transformations, it may be possible to predict the ecological impacts of adding/removing certain microorganisms. These predictions can also be experimentally verified and allow us to better understand the driving factors of microbial communities.

References

Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the Surprising Behavior of Distance Metrics in High Dimensional Space. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, Jan Van den Bussche, and Victor Vianu, editors, *Database Theory — ICDT 2001*, volume 1973, pages 420–434. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. ISBN 978-3-540-41456-8 978-3-540-44503-6. Series Title: Lecture Notes in Computer Science.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN, December 2017. arXiv:1701.07875 [cs, stat].

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. *arXiv:1607.06450 [cs, stat]*, July 2016. arXiv: 1607.06450.

Nicholas A Barber, Desirae M Klimek, Jennifer K Bell, and Wesley D Swingley. Restoration age and reintroduced bison may shape soil bacterial communities in restored tallgrass prairies. *FEMS Microbiology Ecology*, 99(3): fiad007, February 2023. ISSN 1574-6941.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv:2010.11929 [cs]*, June 2021. arXiv: 2010.11929.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *arXiv:1406.2661 [cs, stat]*, June 2014. arXiv: 1406.2661.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved Training of Wasserstein GANs. *arXiv:1704.00028 [cs, stat]*, December 2017. arXiv: 1704.00028.

Anvita Gupta and James Zou. Feedback GAN for DNA optimizes protein functions. *Nature Machine Intelligence*, 1 (2):105–111, February 2019. ISSN 2522-5839. Number: 2 Publisher: Nature Publishing Group.

Debapriya Hazra, Mi-Ryung Kim, and Yung-Cheol Byun. Generative Adversarial Networks for Creating Synthetic Nucleic Acid Sequences of Cat Genome. *International Journal of Molecular Sciences*, 23(7):3701, March 2022. ISSN 1422-0067.

Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, August 2021. ISSN 1367-4803, 1460-2059.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. *arXiv:1810.00825 [cs, stat]*, May 2019. arXiv: 1810.00825.

Chun-Liang Li, Manzil Zaheer, Yang Zhang, Barnabas Poczos, and Ruslan Salakhutdinov. Point Cloud GAN. *arXiv:1810.05795 [cs, stat]*, October 2018. arXiv: 1810.05795.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral Normalization for Generative Adversarial Networks, February 2018. arXiv:1802.05957 [cs, stat].

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved Techniques for Training GANs, June 2016. arXiv:1606.03498 [cs].

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-Attention with Relative Position Representations. *arXiv:1803.02155 [cs]*, April 2018. arXiv: 1803.02155.

Akash Srivastava, Lazar Valkov, Chris Russell, Michael U. Gutmann, and Charles Sutton. VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning, November 2017. arXiv:1705.07761 [stat].

Karl Stelzner, Kristian Kersting, and Adam R Kosiorek. Generative Adversarial Set Transformers. page 6, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv:1706.03762 [cs]*, December 2017. arXiv: 1706.03762.

Ruheng Wang, Yi Jiang, Junru Jin, Chenglin Yin, Haoqing Yu, Fengsheng Wang, Jiuxin Feng, Ran Su, Kenta Nakai, Quan Zou, and Leyi Wei. DeepBIO: an automated and interpretable deep-learning platform for high-throughput biological sequence prediction, functional annotation and visualization analysis. *Nucleic Acids Research*, 51(7): 3017–3029, April 2023. ISSN 0305-1048.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On Layer Normalization in the Transformer Architecture, June 2020. arXiv:2002.04745 [cs, stat].