FIBER BUNDLE MORPHISMS AS A FRAMEWORK FOR MODELING MANY-TO-MANY MAPS

Elizabeth Coda, Nico Courts, Colby Wight, Loc Truong, WoongJo Choi, Charles Godfrey, Tegan Emerson,[*] Keerti Kappagantula, Henry Kvinge[†]
Pacific Northwest National Lab
{first.last}@pnnl.gov

ABSTRACT

While it is not generally reflected in the 'nice' datasets used for benchmarking machine learning algorithms, the real-world is full of processes that would be best described as many-to-many. That is, a single input can potentially yield many different outputs (whether due to noise, imperfect measurement, or intrinsic stochasticity in the process) and many different inputs can yield the same output (that is, the map is not injective). For example, imagine a sentiment analysis task where, due to linguistic ambiguity, a single statement can have a range of different sentiment interpretations while at the same time many distinct statements can represent the same sentiment. When modeling such a multivalued function $f : X \to Y$, it is frequently useful to be able to model the distribution on $f(x)$ for specific input x as well as the distribution on fiber $f^{-1}(y)$ for specific output y. Such an analysis helps the user (i) better understand the variance intrinsic to the process they are studying and (ii) understand the range of specific input x that can be used to achieve output y. Following existing work which used a fiber bundle framework to better model many-to-one processes, we describe how morphisms of fiber bundles provide a template for building models which naturally capture the structure of many-to-many processes.

1 INTRODUCTION

Variation is ubiquitous in the real-world. This is especially true for physical processes where a single input to the system can produce many different possible outputs. For example, in nearly all scientific fields it is common for an experiment to yield (slightly) different results each time it is repeated, even when all controllable parameters are held fixed. On the other hand, variation in systems can also flow in the other direction. It is common to have many inputs to the system that yield a single output. For example, there are infinitely many images of cats that all map to the label 'cat'. From the perspective of machine learning (ML) we can think of processes that possess both of these properties as being many-to-many. Each input to the system may yield many different outputs and many different inputs can also yield a single output. A non-injective multivalued function underlies such processes.

In this paper we develop a deep learning architecture capable of modeling the distributions found on both images and fibers of a many-to-many map. More precisely, let $f : X \to Y$ be a many-to-many map, so that $f(x)$ and $f^{-1}(y)$ are both sets for any $x \in X$ and $y \in Y$. A probability distribution on X induces probability distributions on both $f(x)$ and $f^{-1}(y)$. Our goal is to build a model that simultaneously learns all of these distributions over the course of training. Such information may not be necessary for simple inference tasks, but modeling the distributions on both images and fibers gives information that can be critical for decision making. For example, understanding the distribution $f^{-1}(y)$ can help a user identify all possible inputs that can be applied to achieve the

[*]Dr. Emerson holds joint appointments in the Department of Mathematics at Colorado State University and the Department of Mathematical Sciences at the University of Texas, El Paso.

[†]Dr. Kvinge holds a joint appointment in the Department of Mathematics at the University of Washington.

specific desired output y. On the other hand, being able to understand $f(x)$ for varying x can point to those x that may yield less consistent values in Y.

The present work was inspired by a problem in advanced manufacturing where one wants to understand all the different manufacturing processing parameter settings X that will lead to a single desirable material property $y \in Y$. If f represents the manufacturing process this amounts to understanding $f^{-1}(y)$. At the same time, certain processing parameters in X can yield less inconsistent results (more variation in $f(x)$), which is often useful information when each processing run is time-, labor-, and resource-intensive.

To achieve both of these objectives within a single model architecture we build off the work of Courts & Kvinge (2022) which proposed a fiber bundle framework to model many-to-one maps. We find that while fiber bundles offer a useful template for many-to-one maps, bundle morphisms better capture the structure required for an architecture meant to capture many-to-many maps. In the model that we propose, which we call a *Bundle Morphism Network (BMNet)*, we offload the variability intrinsic to f by lifting it to a new 'bundle morphism' $F : E_X \to E_Y$. We incorporate local trivializations into F so that it is easy to move from X to E_X and Y to E_Y. Finally, the variation that emerges when we consider either $f(x)$ or $f^{-1}(y)$ is encoded in simple distributions (e.g., 1-dimensional Gaussians) on the fibers of E_X and E_Y respectively. The local trivializations, which we build into our model using invertible neural networks, are trained to transform these distributions to the distribution on $f(x)$ and $f^{-1}(y)$.

We test BMNet on several synthetic datasets that represent many-to-many maps with a fiber bundle like flavor. We benchmark BMNet against other many-to-many models as well as their relatives such as conditional GANs (cGANs) (Mirza & Osindero, 2014) and conditional normalizing flow (cNF) models (Winkler et al., 2019). We show that BMNet generally outperforms these models on most of our synthetic datasets suggesting that the next step is to benchmark BMNet against a wider variety of more complicated real-world datasets.

2 Fiber Bundles and Bundle Morphisms

Early in the history of geometry it was recognized that one way to analyze a space is to decompose it into simpler constituent parts. The most familiar of such decompositions is the product, wherein space X is identified as the Cartesian product of two simpler spaces Y and Z, $X \cong Y \times Z$. The torus T is a simple example of a space that is (topologically) the product of two spheres, $T \cong S^1 \times S^1$. Some spaces, however, may appear to be a product space locally and yet fail to be a product space globally due to a 'twist' in the product structure. One of the simplest examples of this is the Möbius band. The notion of a fiber bundle, first introduced in Seifert (1933) and Whitney (1935), is meant to capture the essence of this phenomenon. We include a review of the definition of a fiber bundle in Section A.8 of the Appendix.

As with many structures within topology and geometry, it is useful to identify the maps between fiber bundles that preserve their structure. Let $\mathbf{E_1} = (E_1, B_1, Z_1, \pi_1)$ and $\mathbf{E_2} = (E_2, B_2, Z_2, \pi_2)$ be two fiber bundles. A *bundle morphism* F between total spaces $\mathbf{E_1}$ and $\mathbf{E_2}$ is a continuous map $F : E_1 \to E_2$ and continuous map on base spaces $f : B_1 \to B_2$ such the following diagram commutes.

$$\begin{tikzcd} E_1 \arrow[r, "F"] \arrow[d, "\pi_1"'] & E_2 \arrow[d, "\pi_2"] \\ B_1 \arrow[r, "f"] & B_2 \end{tikzcd}$$

Figure 1: Bundle morphism diagram.

While it is not intended to follow this definition "on the nose", the concept of a bundle morphism informs the structure of the Bundle Morphism Network introduced in Section 3.

3 Bundle Morphism Network

Building off of BundleNet, *Bundle Morphism Network (BMNet)* is a neural network $\Phi : X \times Z_1 \times \mathcal{R}_X \times \mathcal{R}_Y \to Y \times Z_2$ where X is the input space for the task, Y is the output space, Z_1 and Z_2 are called the fibers, and $\mathcal{R}_X = \{r_1^X, \ldots, r_{n_1}^X\}$ and $\mathcal{R}_Y = \{r_1^Y, \ldots, r_{n_2}^Y\}$ are a collection of conditioning vectors representing neighborhoods of X and Y respectively. For fixed $r_i^X \in \mathcal{R}_X, r_j^Y \in \mathcal{R}_Y$

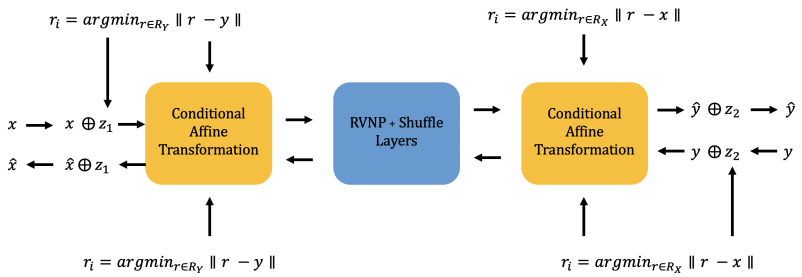

Figure 2: A diagram outlining the architecture of Bundle Morphism Network.

we write $\Phi_{i,j} := \Phi(-, -, r_i^X, r_j^Y)$. $\Phi_{i,j}$ is invertible (we use the `FrEIA` package (Ardizzone et al., 2021) to build Φ in our experiments). For clarity we refer to the process of running the model in the direction $X \times Z_1 \to Y \times Z_2$ as the *forward direction* and running the model in the direction $Y \times Z_2 \to X \times Z_1$ as the *reverse direction*. Z_1 and Z_2 are assumed to follow user chosen distributions $\mathcal{D}_1$ and $\mathcal{D}_2$. In the experiments in this paper we choose to use the uniform distribution on the circle when sampling from Z_1 and Z_2 (justification for this was given in (Courts & Kvinge, 2022)).

Elements of $\mathcal{R}_X$ and $\mathcal{R}_Y$ consist of representatives of neighborhoods in the support of the training data input distribution (in X) and the training data output distribution (in Y) respectively. Given a training set $D \subset X \times Y$, we cluster both the input from X and the labels from Y and let $\mathcal{R}_X$ and $\mathcal{R}_Y$ be the cluster centroids. Described in the forward direction, the model consists of a conditional affine transformation conditioned on elements of $\mathcal{R}_Y$, several RVNP blocks and coordinate permutation layers (Dinh et al., 2016), and an additional conditional affine transformation conditioned on elements of $\mathcal{R}_X$. These invertible conditional layers correspond to the local trivializations for each "bundle".

During training, a batch consists of training examples $\{(x_i, y_i)\}$ such that all x in the batch belong to the same cluster with center $r_i^X \in R_X$ and all y in the batch belong to the same cluster with center $r_j^Y \in R_Y$. For each x, we sample a z_1 from distribution $\mathcal{D}_1$ on Z_1. To run the model in the forward direction, we condition on r_j^Y and r_i^X and obtain $(\hat{y}, \hat{z}_2) = \Phi(x, z_1, r_i^X, r_j^Y)$. To run the model in the reverse direction, we sample z_2 and obtain $(\hat{x}, \hat{z}_1) = \Phi^{-1}(y, z_2, r_i^X, r_i^Y)$. Once we have a collection of such $\hat{y}$, which we denote by $\hat{S}$, and y, which we denote by S, the loss in the forward direction is

$$\mathcal{L}_{\text{forward}}(\hat{S}, S) = \frac{1}{|\hat{S}|} \sum_{\hat{s} \in \hat{S}} \min_{s \in S} ||\hat{s} - s||^2 + \frac{1}{|S|} \sum_{s \in S} \min_{\hat{s} \in \hat{S}} ||\hat{s} - s||^2.$$

$\mathcal{L}_{\text{reverse}}$ is defined analogously. This is a symmetric version of the mean-squared minimum distance, used to ensure that the learned distribution covers the entirety of the training distribution. Notice that this is different from the loss function in Courts & Kvinge (2022). We found that applying an analogous version of the loss found in that paper was unstable when applied to our problem. The detailed training algorithm is outlined in Algorithm 1.

At inference time, we will only have access to one of x or y. If we wish to run the model in the forward direction and generate the distribution of $f(x)$, we need a conditioning vector $r_j^Y \in \mathcal{R}_Y$, which we cannot assume we have since we do not have a y value paired with x. To get around this problem, we find the k nearest neighbors of x from the training set and randomly sample one (x', y'). We use the corresponding y' from (x', y') to determine r_j^Y. The detailed inference algorithm is outlined in Algorithm 2. Running inference in the reverse direction is analogous.

4 Experiments

We evaluate on three synthetic datasets. The first dataset, *Torus-to-circle I*, is illustrated in Figure 4 and consists of a map from the torus to a circle. In the forward direction, (Figure 4, center), each

Algorithm 1 Training Procedure

Input: Training dataset $D = \{(x_i, y_i)\}$, prior distributions $\mathcal{D}_1$ and $\mathcal{D}_2$
Output: Trained model Φ with cluster centers $\mathcal{R}_X$ and $\mathcal{R}_Y$ and updated priors $\mathcal{D}_1$ and $\mathcal{D}_2$

Cluster the training data (in X) to obtain n_1 cluster centers $\mathcal{R}_X$
Cluster the training data (in Y) to obtain n_2 cluster centers $\mathcal{R}_Y$
for each epoch **do**
  for $r_i^X \in \mathcal{R}_X$ and $r_j^Y \in \mathcal{R}_Y$ **do**
    Construct $D_{ij} = \{(x, y) | r_i^X = \arg\min_{r \in \mathcal{R}_X} |r - x|, r_j^Y = \arg\min_{r \in \mathcal{R}_Y} |r - y|\}$
    $X = \text{proj}_x(D_{ij})$ (set projection to 1st coordinate)
    $Y = \text{proj}_y(D_{ij})$ (set projection to 2nd coordinate)
    Sample $Z_1 \sim \mathcal{D}_1^i$ and $Z_2 \sim \mathcal{D}_2^j$
    $(\hat{Y}, \hat{Z}_2) = \Phi(X, Z_1, r_i^X, r_j^Y)$
    $(\hat{X}, \hat{Z}_1) = \Phi^{-1}(Y, Z_2, r_i^X, r_j^Y)$
    $\mathcal{L} = \mathcal{L}_{forward}(\hat{Y}, Y) + \mathcal{L}_{reverse}(\hat{X}, X)$
    Update the weights of Φ
    Update the parameters of the prior distributions $\mathcal{D}_1^i$ and $\mathcal{D}_2^j$
  end for
end for

Algorithm 2 Inference Procedure: Forward Direction

Input: Input x, training dataset $D = \{(x_i, y_i)\}$, trained model Φ, with neighborhood centers $\mathcal{R}_X$ and $\mathcal{R}_Y$ and priors $\mathcal{D}_1$ and $\mathcal{D}_2$
Output: A set S of n samples from the distribution $f(x)$

$S = \{\}$
$r_i^X = \arg\min_{r \in \mathcal{R}_X} |r - x|$
Construct a dataset $D_x = \{(x_i, y_i) | (x_i, y_i) \in D \text{ and } x_i \in \text{Neigh}(x)\}$
while $|S| < n$ **do**
  Sample $y' \sim \text{proj}_y(D_x)$
  $r_j^Y = \arg\min_{r \in \mathcal{R}_Y} |r - y'|$
  Sample $z \sim \mathcal{D}_1^i$
  $(\hat{y}, \hat{z}_2) = \Phi(x, z, r^X, r^Y)$
  $S = S \cup \{\hat{y}\}$
end while

slice of the torus maps to a random point sampled uniformly from an interval of the circle centered on the projection of that slice. The second dataset, *Torus-to-circle II* (Figure 5), similarly maps the torus to the circle. In the forward direction a slice of the torus with axis of revolution angle θ is projected to one of two points on the circle with angles $\theta/2$ or $\theta/2 + \pi$, with equal probability. The third dataset, *Möbius-to-circle*, is very similar to *Torus-to-circle I*, with the torus replaced by a Möbius band. More details on each dataset can be found in Section A.4 of the Appendix.

We compare BMNet to two other many-to-many models, Augmented CycleGAN (AugCGAN) (Almahairi et al.) and Latent Normalizing Flows for Many-to-Many Cross-Domain Mappings (LNFMM) (Mahajan et al., 2020). Additionally, we include a pair of cGAN models (Mirza & Osindero, 2014) and a pair of cNF models (Winkler et al., 2019). Note that in order to adapt cGANs and cNFs for many-to-many problems, we train one cGAN and cNF for the forward direction and one for the backward direction. We follow the evaluation procedure proposed in Courts & Kvinge (2022). In the forward direction, to evaluate the global distribution, we generate 5000 points and compare to 5000 points from the true distribution. To evaluate local distributions, we sample 15 points U from X and for each $x \in U$, use the model to generate 200 points in Y and compare this to the true distribution defined by the dataset. Evaluation of the backward direction is analogous. To compare true and reconstructed distributions we use a range of different metrics, including the 1-Wasserstein metric, mean squared minimum distance (MSMD), maximum mean discrepancy (MMD), and KL divergence. To obtain confidence intervals, we train each model 5 times and report the mean with 95% confidence intervals.

4.1 RESULTS

As can be seen in Table 1, at the global level our model is the best performing model in terms of the Wasserstein-1 distance on most of the datasets. While AugCGAN performs well at the global level, as seen in Table 2, this many-to-many model does not learn the local distributions. The cNF model also generally performs well, particularly at the local level, though requires training a separate model in each direction and does not learn the discrete nature of the Torus-to-circle II dataset in the forward direction. This is depicted in Appendix A.6.

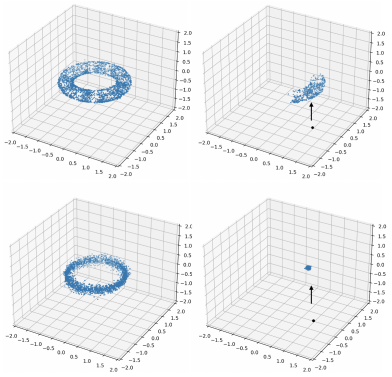

Figure 3: Global and local distributions generated by our model (top row) and an AugCGAN (bottom row) for Torus-to-circle I in the reverse direction.

To give a flavor of the difference in reconstructions for models explicitly designed to model many-to-many processes, the reader should consult Figure 3 which depicts the global and local distributions of this dataset reconstructed by BMNet (top row) and an AugCGAN (bottom row). Locally, while the AugCGAN struggles to learn the full range of possible inputs to achieve a given y, our model is able to recreate a reasonable approximation of cylindrical sections of the torus. In general, we see this pattern repeated across datasets in both the forward and backward direction. This suggests to us that the structure of BMNet makes it more capable of handling the problem set forth in this short paper. Full results for each dataset, all models, all metrics, and further visualizations are available in SectionsA.6 and A.7 of the Appendix.

Table 1: The Wasserstein-1 ($\times 10^{-2}$) global metric on all datasets and models.

	BMNet	AugCGAN	LNFMM	cGAN	cNF
Torus-to-circle I: Fwd	1.91 ± 0.34	9.80 ± 0.88	15.01 ± 5.55	4.38 ± 1.61	$\mathbf{1.28 \pm 0.16}$
Torus-to-circle I: Rev	$\mathbf{4.38 \pm 0.57}$	19.13 ± 3.51	25.95 ± 3.16	28.96 ± 3.72	7.10 ± 0.58
Torus-to-circle II: Fwd	$\mathbf{0.38 \pm 0.10}$	0.45 ± 0.05	23.36 ± 5.94	6.29 ± 0.84	27.54 ± 5.49
Torus-to-circle II: Rev	$\mathbf{3.51 \pm 0.17}$	9.48 ± 1.0	20.54 ± 0.35	48.84 ± 12.01	8.13 ± 0.70
Möbius-to-circle: Fwd	1.85 ± 0.39	$\mathbf{0.68 \pm 0.11}$	6.50 ± 2.62	8.49 ± 1.54	2.58 ± 0.59
Möbius-to-circle: Rev	$\mathbf{2.33 \pm 0.19}$	2.37 ± 0.16	23.89 ± 11.95	103.7 ± 4.1	3.36 ± 0.05

Table 2: The Wasserstein-1 ($\times 10^{-2}$) local metric on all datasets and models.

	BMNet	AugCGAN	LNFMM	cGAN	cNF
Torus-to-circle I: Fwd	9.30 ± 1.26	119.9 ± 10.4	20.31 ± 2.22	20.33 ± 2.01	$\mathbf{5.68 \pm 0.72}$
Torus-to-circle I: Rev	$\mathbf{12.01 \pm 0.95}$	123.6 ± 9.9	46.23 ± 0.71	37.48 ± 1.80	12.43 ± 0.28
Torus-to-circle II: Fwd	16.05 ± 4.38	127.1 ± 2.8	31.71 ± 2.29	$\mathbf{9.37 \pm 1.01}$	32.86 ± 2.26
Torus-to-circle II: Rev	$\mathbf{4.49 \pm 0.54}$	116.1 ± 13.7	23.58 ± 0.15	56.39 ± 5.73	10.36 ± 0.43
Möbius-to-circle: Fwd	9.48 ± 0.98	87.3 ± 13.41	16.67 ± 2.08	42.63 ± 3.71	$\mathbf{6.08 \pm 0.93}$
Möbius-to-circle: Rev	8.77 ± 0.79	94.32 ± 14.2	42.78 ± 2.03	117.17 ± 1.84	$\mathbf{8.60 \pm 0.38}$

5 CONCLUSION

Many-to-many processes are common in nature. While there are a range of deep learning-based frameworks that can be used to solve simple tasks related to these processes, network architectures for more comprehensive modeling have until now remained limited. Guided by the concept of a bundle morphism, in this paper we introduce the first model architecture explicitly designed to capture the more nuanced aspects of many-to-many processes, providing the capability to model not only the distribution as a whole, but also the distributions of both the image of points and their fibers.

ACKNOWLEDGMENTS

This research was supported by the Mathematics for Artificial Reasoning in Science (MARS) initiative via the Laboratory Directed Research and Development (LDRD) investments at Pacific Northwest National Laboratory (PNNL). PNNL is a multi-program national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE-AC05-76RL0-1830.

REFERENCES

Amjad Almahairi, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data.

Lynton Ardizzone, Jakob Kruse, Peter Steinbach, Till Bungert, Peter Sorrenson, Alexandre René, and Johnson Yue. Framework for easily invertible architectures, 2021. URL https://github.com/VLL-HD/FrEIA.

Nico Courts and Henry Kvinge. Bundle networks: Fiber bundles, local trivializations, and a generative approach to exploring many-to-one maps. 2022.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *CoRR*, abs/1605.08803, 2016. URL http://arxiv.org/abs/1605.08803.

Tingran Gao, Jacek Brodzki, and Sayan Mukherjee. The geometry of synchronization problems and learning group actions. *Discrete & Computational Geometry*, 65(1):150–211, 2021.

Felix Knöppel and Ulrich Pinkall. Complex line bundles over simplicial complexes and their applications. In *Advances in Discrete Differential Geometry*, pp. 197–239. Springer, Berlin, Heidelberg, 2016.

Shweta Mahajan, Iryna Gurevych, and Stefan Roth. Latent normalizing flows for many-to-many cross-domain mappings. 2020.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Luis Scoccola and Jose A Perea. Approximate and discrete euclidean vector bundles. *arXiv preprint arXiv:2104.07563*, 2021.

H. Seifert. Topologie Dreidimensionaler Gefaserter Räume. *Acta Mathematica*, 60(none):147 – 238, 1933. doi: 10.1007/BF02398271. URL https://doi.org/10.1007/BF02398271.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28: 3483–3491, 2015.

Hassler Whitney. Sphere-spaces. *Proceedings of the National Academy of Sciences*, 21(7):464–468, 1935. ISSN 0027-8424. doi: 10.1073/pnas.21.7.464. URL https://www.pnas.org/content/21/7/464.

Christina Winkler, Daniel E. Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

Ke Ye and Lek-Heng Lim. Cohomology of cryo-electron microscopy. *SIAM Journal on Applied Algebra and Geometry*, 1(1):507–535, 2017.

Sharon Zhou, Eric Zelikman, Fred Lu, Andrew Y Ng, Gunnar Carlsson, and Stefano Ermon. Evaluating the disentanglement of deep generative models through manifold topology. *arXiv preprint arXiv:2006.03680*, 2020.

A APPENDIX

A.1 RELATED WORK

As discussed above, this work builds on Courts & Kvinge (2022) where a fiber bundle-inspired deep learning framework was used to address the problem of modeling the fibers in many-to-one machine learning tasks. Our work differs from that one in that (i) the problem (modeling both fibers and images in many-to-many maps) is distinct from the problem that is considered in Courts & Kvinge (2022), (ii) the inspiration for BMNet's architecture is the concept of a bundle morphism, while the inspiration for BundleNet was an individual fiber bundle, and (iii) BMNet's training routine and architecture are distinct from BundleNet.

Beyond BundleNet, fiber bundles (and more specifically vector bundles) have recently begun to see more use as a tool in data science. Scoccola & Perea (2021) for example, seek to develop the theory required to use vector bundles in a rigorous way in data science. Recent applications of vector bundles include: cryo-electron microscopy (Ye & Lim, 2017; Gao et al., 2021) and computer graphics (Knöppel & Pinkall, 2016). The present work differs from those listed above in that we only use the concept of a bundle morphism to guide the problem framework and model architecture for learning many-to-many maps.

More broadly, deep learning-based generative models have recently made great strides forward, both in terms of the complexity of distributions that can be modeled and the fidelity of reconstructions. While the majority of these approaches seek to model a single distribution, there have also been a significant number of efforts to develop conditional generative approaches which are capable of modeling multiple distributions conditioned on additional input. These include conditional variational autoencoders (Sohn et al., 2015) and conditional generative adversarial networks (Mirza & Osindero, 2014). One of the results of this paper was to show that like the many-to-one fiber modeling problem introduced in (Courts & Kvinge, 2022), the many-to-many fiber and image modeling problem cannot be satisfactorily addressed with existing architectures. Recent topological approaches to evaluating generative model performance includes (Zhou et al., 2020).

A.2 LIMITATIONS

Because it fit well within the training routine we wanted to use, we chose to use an architecture that is invertible after conditioning (as in Courts & Kvinge (2022)). Of course, in practice bundle morphisms need not be invertible (though the local trivializations that are built into our model are) and it is likely that this constraint will hurt performance on certain types of datasets. In the future it would be worth revisiting the training procedure to understand how to include non-invertible components into the architecture.

Beyond this, the major limitation of this work is that the experiments that we describe are limited to synthetic datasets. We felt that this was a reasonable choice based on space limitations and the novelty of the problem, but future work should identify and test against real-world many-to-many datasets. One challenge related to the former is that most existing benchmark ML datasets are chosen to be 'nice' and thus there is a bias against one-to-many type phenomena.

A.3 THE ANALOGY BETWEEN A BUNDLE MORPHISM NETWORK AND A TRUE BUNDLE MORPHISM

We caution that while BMNet is inspired by the idea of a bundle morphism, we did not strictly constrain the model to this definition. In this section we briefly lay out the analogy between components of the model architecture and a bundle morphism. To fix notation assume that $\mathbf{E_1} = (E_1, B_1, Z_1, \pi_1)$ and $\mathbf{E_2} = (E_2, B_2, Z_2, \pi_2)$ are two fiber bundles where E_i is the total space, B_i is the base space, Z_i is the fiber, and π_i is the projection operator for $i \in \{1, 2\}$. Also let $F : E_1 \to E_2$ be a bundle morphism that descends to a continuous map $f : B_1 \to B_2$ making the required diagram commute. On the other hand, let $\Phi : X \times Z_1 \times \mathcal{R}_X \times \mathcal{R}_Y \to Y \times Z_2$ be an instance of BMNet.

- B_1 and B_2 correspond to X and Y respectively.

- Z_1 and Z_2 in the fiber bundle set-up correspond to Z_1 and Z_2 in the description of BMNet. Note the distributions associated with Z_1 and Z_2 in BMNet are critical to the model, whereas probability distributions on the fibers of fiber bundles are not commonly considered in topology or geometry. This may be an interesting area for further investigation.
- The representatives $\mathcal{R}_X = \{r_1^X, \ldots, r_{n_1}^X\}$ and $\mathcal{R}_Y = \{r_1^Y, \ldots, r_{n_2}^Y\}$ correspond to the local trivialization neighborhoods in B_1 and B_2.
- $\Phi_{i,j}$ corresponds to $\phi_j \circ F \circ \phi_i^{-1}$, where $\phi_i : \pi_1^{-1}(U_i) \to U_i \times F_1$ and $\phi_j : \pi_2^{-1}(U_j) \to U_j \times F_2$ are local trivializations between neighborhoods $\pi_1^{-1}(U_i) \subseteq E_1$ and $\pi_2^{-1}(U_j) \subseteq E_2$ respectively.

A.4 Datasets

Torus-to-circle I: Here we describe the synthetic datasets used in more detail. For the Torus-to-circle I dataset, given radius r and radius R, the torus representation is:

$$T = \{((R + r\ \cos\ \theta)\ \cos\ \phi, (R + r\ \cos\ \theta)\ \sin\ \phi, r\ \sin\ \theta) : 0 \leq \theta, \phi \leq 2\pi\}$$

We sample from T with $R = 1$ and $r = 0.25$ by randomly sampling θ and ϕ. We pair this with a point from the circle by randomly sampling a perturbation α to add to ϕ. :

$$C = \{(\cos(\phi + \alpha), \sin(\phi + \alpha), : 0 \leq \phi \leq 2\pi, \frac{-\pi}{4} \leq \alpha \leq \frac{\pi}{4}\}$$

This dataset is visualized in Figure 4.

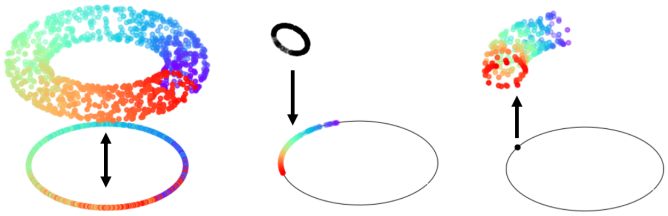

Figure 4: The *Torus-to-circle I* many-to-many dataset (left). Each torus slice maps to a uniformly sampled random point on an interval of the circle (center).

Torus-to-circle II: Generation of a point from the Torus-to-circle II dataset begins by sampling a point from T as above and then pairing that with one of two points from C, either $(\cos(\theta/2), \sin(\theta/2))$ or $(\cos(\theta/2 + \pi), \sin(\theta/2 + \pi))$ with equal probability. This dataset is depicted in Figure 5.

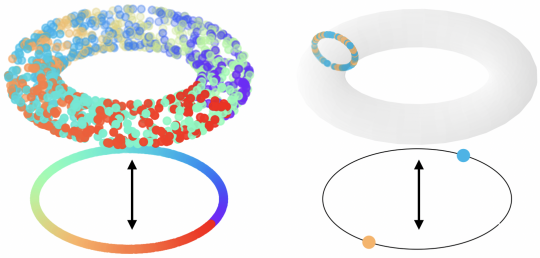

Figure 5: The *Torus-to-circle II* dataset. A point at angle θ of the torus maps to one of two points on the circle, $\theta/2$ and $\theta/2 + \pi$ (right) with equal probability.

Möbius-to-circle: The Möbius-to-circle dataset is given by:

$$M = \{(R\ \cos\theta - s\ \cos\frac{\theta}{2}\cos\theta, R\ \sin\theta - s\ \cos\frac{\theta}{2}\sin\theta, s\ \sin\frac{\theta}{2}) : 0 \leq \theta \leq 2\pi, -r \leq s \leq r\}$$

We sample from M with $R = 1$ and $r = 0.25$ by randomly sampling θ and s. We pair this with a point from the circle as in Torus-to-Circle I.

A.5 Training Details

Each model was trained on 5,000 training examples on a single GPU for 2,000 epochs. An initial learning rate of 10^{-4} was used, and was reduced by a factor of 10 at 1000 and 1500 epochs.

Experimentally, we have found that the most effective prior to impose on Z_1 and Z_2 was a uniform circular prior. During training, we update the parameters of the prior distributions based on the learned z_1 and z_2. Additionally, we enforce a regularization term on z_1 and z_2 for model stability during training.

A.6 Visualizing reconstructions

In this section we provide visualizations of the reconstructions of BMNet and the other models in Figures 6 through 9.

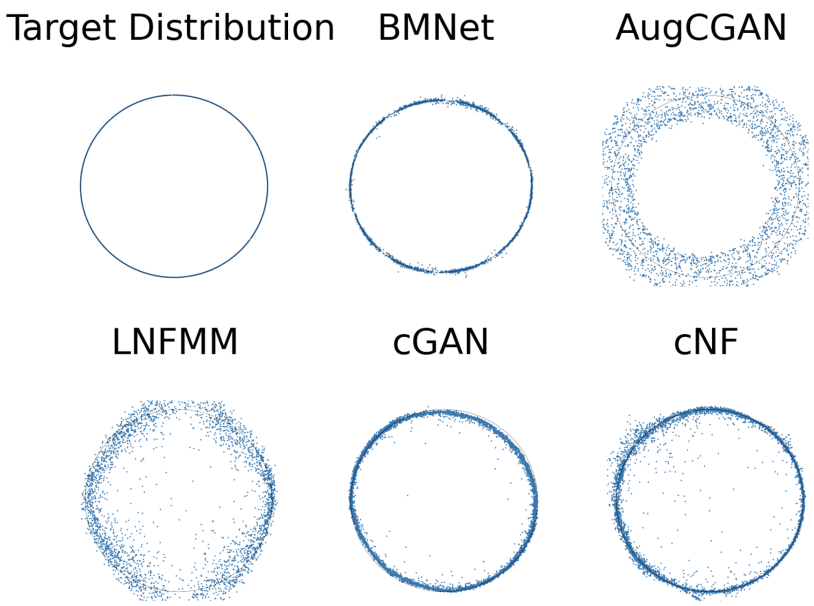

Figure 6: Global distributions generated by the different models in the forward direction on Torus-to-circle I.

A.7 Full Results

We provide full results with all metrics for both models on all three datasets in Tables 3 to 14.

Table 3: All metrics Torus-to-circle I: forward direction (global)

	BMNet	AugCGAN	LNFMM	cGAN	cNF
MSMD ($\times 10^{-4}$)	$\mathbf{3.08 \pm 2.56}$	189.8 ± 24.9	661.2 ± 428.6	44.80 ± 20.97	17.24 ± 7.06
MMD ($\times 10^{-3}$)	0.54 ± 0.29	0.82 ± 0.13	6.27 ± 3.0	2.35 ± 1.22	$\mathbf{0.49 \pm 0.13}$
KL-fwd	$\mathbf{0.93 \pm 0.10}$	4.42 ± 0.16	4.21 ± 0.36	3.93 ± 0.49	1.27 ± 0.07
KL-bwd	$\mathbf{3.52 \pm 0.21}$	6.97 ± 0.12	6.79 ± 0.44	6.71 ± 0.54	3.82 ± 0.10
$\mathcal{W}_1$ ($\times 10^{-2}$)	1.91 ± 0.34	9.80 ± 0.88	15.01 ± 5.55	4.38 ± 1.61	$\mathbf{1.28 \pm 0.16}$
$\mathcal{W}_2$ ($\times 10^{-2}$)	$\mathbf{0.11 \pm 0.06}$	0.97 ± 0.13	2.62 ± 1.42	0.56 ± 0.27	0.12 ± 0.09

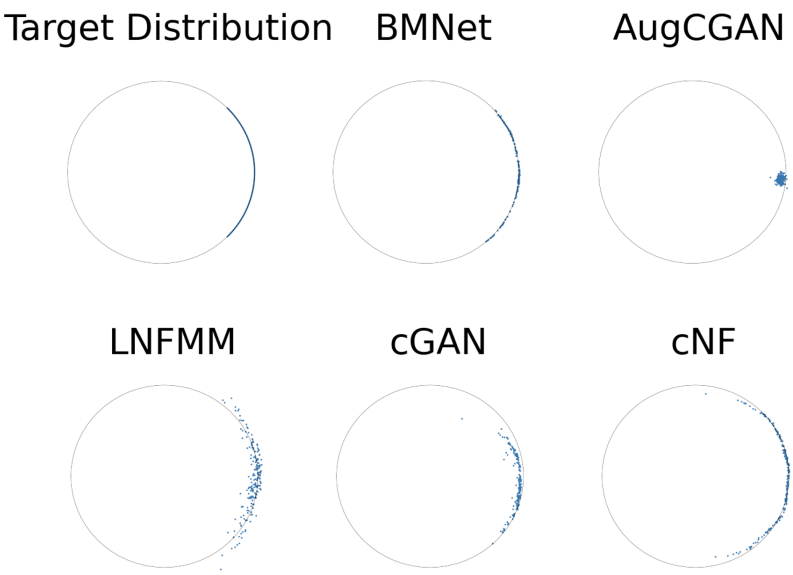

Figure 7: Local distributions generated by the different models in the forward direction on Torus-to-circle I.

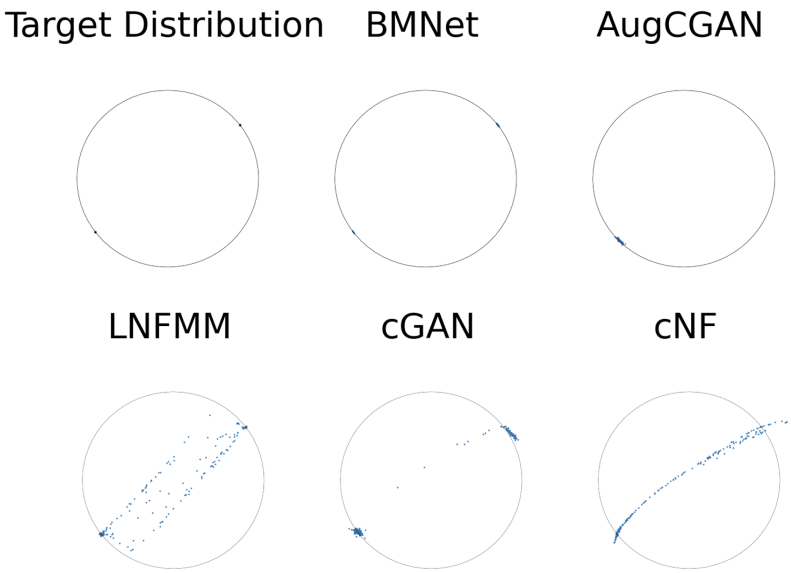

Figure 8: Local distributions generated by the different models in the forward direction on Torus-to-circle II.

Table 4: All metrics Torus-to-circle I: forward direction (local)

	BMNet	AugCGAN	LNFMM	cGAN	cNF
MSMD ($\times 10^{-4}$)	$\mathbf{13.98 \pm 4.09}$	7127 ± 1230	400.1 ± 142.1	49.27 ± 13.87	75.82 ± 32.74
MMD ($\times 10^{-3}$)	14.48 ± 3.31	612.4 ± 57.6	37.33 ± 5.50	64.29 ± 8.69	$\mathbf{11.09 \pm 2.45}$
KL-fwd	1.57 ± 0.29	9.87 ± 0.46	3.66 ± 0.37	3.50 ± 0.33	$\mathbf{0.71 \pm 0.10}$
KL-bwd	3.64 ± 0.42	12.05 ± 0.32	5.90 ± 0.30	5.92 ± 0.35	$\mathbf{2.38 \pm 0.16}$
$\mathcal{W}_1$ ($\times 10^{-2}$)	9.30 ± 1.26	119.9 ± 10.4	20.31 ± 2.22	20.33 ± 2.01	$\mathbf{5.68 \pm 0.72}$
$\mathcal{W}_2$ ($\times 10^{-2}$)	1.37 ± 0.32	90.27 ± 11.1	4.78 ± 0.95	4.33 ± 0.62	$\mathbf{0.94 \pm 0.94}$

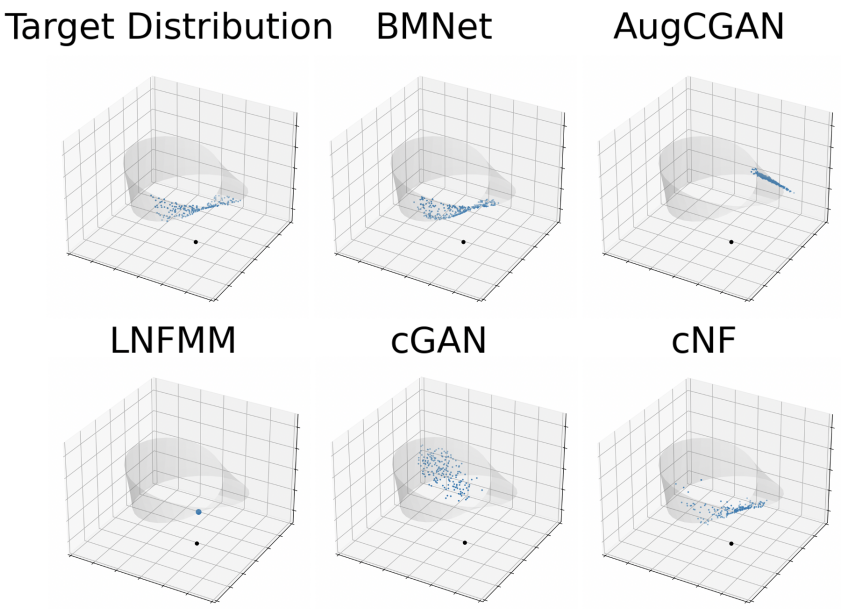

Figure 9: Local distributions generated by the different models in the reverse direction on Möbius-to-circle.

Table 5: All metrics Torus-to-circle I: reverse direction (global)

	BMNet	AugCGAN	LNFMM	cGAN	cNF
MSMD ($\times 10^{-4}$)	$\mathbf{7.39 \pm 0.58}$	272.4 ± 186.2	193.2 ± 24.9	257.4 ± 88.5	87.86 ± 15.23
MMD ($\times 10^{-3}$)	1.59 ± 0.78	8.93 ± 4.91	16.51 ± 9.64	29.46 ± 10.50	$\mathbf{0.38 \pm 0.13}$
KL-fwd	$\mathbf{0.18 \pm 0.07}$	4.41 ± 0.72	13.85 ± 3.16	6.28 ± 0.88	1.04 ± 0.19
KL-bwd	$\mathbf{0.64 \pm 0.16}$	5.71 ± 0.67	7.65 ± 0.12	6.62 ± 0.55	2.05 ± 0.13
$\mathcal{W}_1$ ($\times 10^{-2}$)	$\mathbf{4.38 \pm 0.57}$	19.13 ± 3.51	25.95 ± 3.16	28.96 ± 3.72	7.10 ± 0.58
$\mathcal{W}_2$ ($\times 10^{-2}$)	$\mathbf{0.32 \pm 0.09}$	2.46 ± 1.01	4.67 ± 1.33	6.78 ± 1.73	12.43 ± 0.28

Table 6: All metrics Torus-to-circle I: reverse direction (local)

	BMNet	AugCGAN	LNFMM	cGAN	cNF
MSMD ($\times 10^{-4}$)	$\mathbf{46.28 \pm 2.60}$	6422 ± 1005	243.1 ± 28.2	258.0 ± 44.4	153.0 ± 9.4
MMD ($\times 10^{-3}$)	13.00 ± 2.70	554.6 ± 55.9	127.5 ± 9.0	113.6 ± 12.2	$\mathbf{4.67 \pm 0.65}$
KL-fwd	0.90 ± 0.21	10.82 ± 0.71	26.4 ± 1.59	5.51 ± 0.29	$\mathbf{0.46 \pm 0.04}$
KL-bwd	$\mathbf{0.89 \pm 0.20}$	9.13 ± 0.41	6.72 ± 0.04	4.74 ± 0.14	1.40 ± 0.05
$\mathcal{W}_1$ ($\times 10^{-2}$)	$\mathbf{12.01 \pm 0.95}$	123.6 ± 9.9	46.23 ± 0.71	37.48 ± 1.80	12.43 ± 0.28
$\mathcal{W}_2$ ($\times 10^{-2}$)	1.40 ± 0.22	95.18 ± 10.78	13.64 ± 0.38	9.72 ± 0.90	$\mathbf{1.06 \pm 0.06}$

Table 7: All metrics Torus-to-circle II: forward direction (global)

	BMNet	AugCGAN	LNFMM	cGAN	cNF
MSMD ($\times 10^{-4}$)	$\mathbf{0.09 \pm 0.06}$	0.94 ± 0.13	1397 ± 377	251.0 ± 27.7	1571 ± 522
MMD ($\times 10^{-3}$)	$\mathbf{0.17 \pm 0.07}$	0.26 ± 0.08	26.02 ± 8.57	1.83 ± 0.41	33.90 ± 11.76
KL-fwd	$\mathbf{0.32 \pm 0.16}$	1.45 ± 0.17	4.39 ± 0.7	3.53 ± 0.43	5.04 ± 0.53
KL-bwd	$\mathbf{2.27 \pm 0.38}$	3.9 ± 0.14	6.12 ± 0.87	6.11 ± 0.53	7.01 ± 0.35
$\mathcal{W}_1$ ($\times 10^{-2}$)	$\mathbf{0.38 \pm 0.10}$	0.45 ± 0.05	23.36 ± 5.94	6.29 ± 0.84	27.54 ± 5.49
$\mathcal{W}_2$ ($\times 10^{-2}$)	$\mathbf{0.04 \pm 0.01}$	0.06 ± 0.02	6.28 ± 1.28	1.39 ± 0.14	6.08 ± 1.47

Table 8: All metrics Torus-to-circle II: forward direction (local)

	BMNet	AugCGAN	LNFMM	cGAN	cNF
MSMD ($\times 10^{-4}$)	$\mathbf{3.70 \pm 0.91}$	8401 ± 1411	1831 ± 139	328.2 ± 72.6	1942 ± 195
MMD ($\times 10^{-3}$)	38.49 ± 11.42	567.8 ± 38.8	60.31 ± 7.57	$\mathbf{9.37 \pm 2.70}$	65.62 ± 9.80
KL-fwd	5.53 ± 0.18	11.97 ± 0.47	$\mathbf{5.00 \pm 0.11}$	5.28 ± 0.14	6.26 ± 0.16
KL-bwd	NA	NA	NA	NA	NA
$\mathcal{W}_1$ ($\times 10^{-2}$)	16.05 ± 4.38	127.09 ± 2.76	31.71 ± 2.29	$\mathbf{9.37 \pm 1.01}$	32.86 ± 2.26
$\mathcal{W}_2$ ($\times 10^{-2}$)	12.75 ± 3.97	98.67 ± 0.19	12.62 ± 1.41	$\mathbf{2.18 \pm 0.38}$	11.25 ± 1.02

Table 9: All metrics Torus-to-circle II: reverse direction (global)

	BMNet	AugCGAN	LNFMM	cGAN	cNF
MSMD ($\times 10^{-4}$)	$\mathbf{6.21 \pm 0.26}$	160.2 ± 40.5	419.4 ± 26.9	1154 ± 756	113.6 ± 19.6
MMD ($\times 10^{-3}$)	0.98 ± 0.39	0.91 ± 0.25	3.04 ± 0.13	49.84 ± 17.35	$\mathbf{0.45 \pm 0.05}$
KL-fwd	$\mathbf{0.12 \pm 0.05}$	1.53 ± 0.15	16.6 ± 0.32	7.16 ± 0.98	1.55 ± 0.22
KL-bwd	$\mathbf{0.28 \pm 0.08}$	2.49 ± 0.09	7.51 ± 0.07	7.04 ± 0.66	2.40 ± 0.17
$\mathcal{W}_1$ ($\times 10^{-2}$)	$\mathbf{3.51 \pm 0.17}$	9.48 ± 1.0	20.54 ± 0.35	48.84 ± 12.01	8.13 ± 0.70
$\mathcal{W}_2$ ($\times 10^{-2}$)	0.21 ± 0.04	0.24 ± 0.0	2.61 ± 0.11	19.89 ± 7.42	$\mathbf{0.08 \pm 0.28}$

Table 10: All metrics Torus-to-circle II: reverse direction (local)

	BMNet	AugCGAN	LNFMM	cGAN	cNF
MSMD ($\times 10^{-4}$)	$\mathbf{35.43 \pm 17.46}$	1457 ± 2375	403.7 ± 26.6	2671 ± 511	192.6 ± 18.3
MMD ($\times 10^{-3}$)	6.70 ± 1.30	568.0 ± 59.2	21.68 ± 1.85	329.6 ± 49.8	$\mathbf{3.94 \pm 0.61}$
KL-fwd	$\mathbf{2.52 \pm 0.24}$	9.83 ± 0.57	33.94 ± 0.95	8.10 ± 0.35	2.66 ± 0.08
KL-bwd	$\mathbf{5.64 \pm 0.20}$	17.36 ± 0.65	14.17 ± 0.08	14.84 ± 0.40	8.23 ± 0.13
$\mathcal{W}_1$ ($\times 10^{-2}$)	$\mathbf{4.49 \pm 0.54}$	116.1 ± 13.7	23.58 ± 0.15	56.39 ± 5.73	10.36 ± 0.43
$\mathcal{W}_2$ ($\times 10^{-2}$)	$\mathbf{0.39 \pm 0.08}$	91.66 ± 15.36	3.37 ± 0.06	22.25 ± 3.75	0.97 ± 0.09

Table 11: All metrics Möbius-to-circle: forward direction (global)

	BMNet	AugCGAN	LNFMM	cGAN	cNF
MSMD ($\times 10^{-4}$)	2.93 ± 1.82	$\mathbf{2.78 \pm 0.36}$	160.3 ± 97.2	136.8 ± 30.4	49.13 ± 23.95
MMD ($\times 10^{-3}$)	0.37 ± 0.16	$\mathbf{0.34 \pm 0.23}$	2.07 ± 1.13	4.55 ± 1.65	1.42 ± 0.74
KL-fwd	$\mathbf{1.29 \pm 0.51}$	1.65 ± 0.11	3.75 ± 0.17	4.56 ± 0.42	$\mathbf{1.29 \pm 0.03}$
KL-bwd	3.83 ± 0.57	4.20 ± 0.09	6.35 ± 0.21	7.58 ± 0.62	$\mathbf{3.78 \pm 0.08}$
$\mathcal{W}_1$ ($\times 10^{-2}$)	1.85 ± 0.39	$\mathbf{0.68 \pm 0.11}$	6.50 ± 2.62	8.49 ± 1.54	2.58 ± 0.59
$\mathcal{W}_2$ ($\times 10^{-2}$)	0.14 ± 0.03	$\mathbf{0.10 \pm 0.06}$	0.69 ± 0.46	1.308 ± 0.372	0.14 ± 0.14

Table 12: All metrics Möbius-to-circle: forward direction (local)

	BMNet	AugCGAN	LNFMM	cGAN	cNF
MSMD ($\times 10^{-4}$)	$\mathbf{5.68 \pm 1.70}$	6074 ± 2153	204.2 ± 64.8	203.7 ± 60.6	56.49 ± 22.10
MMD ($\times 10^{-3}$)	14.69 ± 3.23	394.3 ± 67.8	35.71 ± 5.55	184.1 ± 24.4	$\mathbf{13.37 \pm 3.50}$
KL-fwd	1.83 ± 0.24	6.95 ± 1.02	3.98 ± 0.31	4.58 ± 0.42	$\mathbf{0.66 \pm 0.14}$
KL-bwd	4.22 ± 0.31	10.64 ± 0.44	6.21 ± 0.39	8.25 ± 0.35	$\mathbf{2.30 \pm 0.21}$
$\mathcal{W}_1$ ($\times 10^{-2}$)	9.48 ± 0.98	87.3 ± 13.41	16.67 ± 2.08	42.63 ± 3.71	$\mathbf{6.08 \pm 0.93}$
$\mathcal{W}_2$ ($\times 10^{-2}$)	1.26 ± 0.27	62.3 ± 14.5	3.29 ± 0.65	15.06 ± 2.26	$\mathbf{1.07 \pm 0.28}$

A.8 Review: Fiber Bundles

A *fiber bundle* is a 4-tuple (E, B, Z, π) of topological spaces and a (surjective) continuous map $\pi : E \to B$. E is assumed to be covered by a collection of open sets $\{U_i\}_{i \in \mathcal{I}}$ and each U_i has a

Table 13: All metrics Möbius-to-circle: reverse direction (global)

	BMNet	AugCGAN	LNFMM	cGAN	cNF
MSMD ($\times 10^{-4}$)	$\mathbf{4.39 \pm 1.61}$	13.39 ± 2.12	562.6 ± 543.5	6581 ± 561	39.31 ± 1.91
MMD ($\times 10^{-3}$)	$\mathbf{0.57 \pm 0.11}$	0.36 ± 0.13	19.64 ± 16.99	126.3 ± 8.5	0.88 ± 0.23
KL-fwd	$\mathbf{0.09 \pm 0.05}$	0.36 ± 0.03	14.09 ± 1.47	12.95 ± 0.81	0.28 ± 0.06
KL-bwd	$\mathbf{0.94 \pm 0.29}$	1.29 ± 0.05	7.87 ± 0.96	11.06 ± 0.23	1.04 ± 0.11
$\mathcal{W}_1$ ($\times 10^{-2}$)	$\mathbf{2.33 \pm 0.19}$	2.37 ± 0.16	23.89 ± 11.95	103.7 ± 4.1	3.36 ± 0.05
$\mathcal{W}_2$ ($\times 10^{-2}$)	0.15 ± 0.03	$\mathbf{0.01 \pm 0.01}$	0.23 ± 1.16	61.24 ± 5.11	$0.13 \pm .10$

Table 14: All metrics Möbius-to-circle: reverse direction (local)

	BMNet	AugCGAN	LNFMM	cGAN	cNF
MSMD ($\times 10^{-4}$)	$\mathbf{18.95 \pm 1.32}$	7230 ± 2280	91.96 ± 44.44	5652 ± 2396	92.58 ± 14.33
MMD ($\times 10^{-3}$)	12.81 ± 2.76	386.5 ± 64.2	126.7 ± 18.6	503.1 ± 24.1	$\mathbf{8.12 \pm 1.01}$
KL-fwd	0.71 ± 0.11	8.02 ± 1.15	26.44 ± 0.53	12.08 ± 1.06	$\mathbf{0.35 \pm 0.06}$
KL-bwd	0.98 ± 0.14	8.97 ± 0.59	7.97 ± 0.13	8.67 ± 0.24	$\mathbf{0.87 \pm 0.08}$
$\mathcal{W}_1$ ($\times 10^{-2}$)	8.77 ± 0.79	94.32 ± 14.2	42.78 ± 2.03	117.17 ± 1.84	$\mathbf{8.60 \pm 0.38}$
$\mathcal{W}_2$ ($\times 10^{-2}$)	1.10 ± 0.24	70.97 ± 16.03	12.98 ± 1.34	74.42 ± 2.67	$\mathbf{0.79 \pm 0.09}$

homeomorphism $\varphi_i : \pi^{-1}(U_i) \to U_i \times Z$, called a *local trivialization*, associated with it. Informally, φ_i ensures that each neighborhood $\pi^{-1}(U_i)$ looks like a direct product of U_i and Z. Finally, each φ_i should respect the projection π so that the following diagram commutes.

$$\begin{tikzcd} \pi^{-1}(U_i) \arrow[r, "\varphi_i"] \arrow[d, "\pi"'] & U_i \times Z \arrow[dl, "\text{proj}"] \\ U_i & \end{tikzcd} \tag{1}$$

E is known as the *total space*, B is known as the *base space*, and Z is known as the *fiber*. Note that the product space $X \cong Y \times Z$ is trivially identified as a fiber bundle where X is the total space E, Y (respectively Z) is the base space B, Z (resp. Y) is the fiber, π is the usual projection map from $Y \times Z$ to Y (resp. Z), $\{U_i\}_{i \in \mathcal{I}}$ consists of a single neighborhood which is all of Y so that $\pi^{-1}(Y) = Y \times Z$, and φ_i is just the identity map. The reader can check that diagram equation 1 easily commutes in this case.