PSEUDO-NONLINEAR DATA AUGMENTATION: AN ENERGY MINIMIZATION VIEWPOINT

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a simple yet novel data augmentation method for general data modalities based on energy-based modeling and principles from information geometry. Unlike most existing generative models, which rely on learning latent representations with black-box models, our proposed framework enables constructing a geometrically aware latent space that depends on the structure of the data itself, which further supports efficient and explicit encoding and decoding procedures. We then present and discuss how to design latent spaces that will subsequently control the augmentation with the proposed algorithm. Empirical results demonstrate that our data augmentation method achieves competitive downstream task performance compared to other baselines, while offering fine-grained controllability that is lacking in other baselines.

1 INTRODUCTION

Data augmentation has advanced considerably in recent years, driven largely by the increasing use of generative models (Kingma & Welling, 2014; Chadebec et al., 2022; Antoniou, 2017; Trabucco et al., 2024) to meet the demand for larger and more diverse datasets (Feng et al., 2021; Wong et al., 2016). Beyond traditional domains such as images, these methods have been extended to a wide range of modalities. Despite their promise, however, generative-model-based augmentation faces several fundamental challenges. First, data augmentation is most valuable when training data is scarce, yet in such cases, we typically lack a pre-trained foundational model for the target domain. This creates a paradox: before we can augment the data, we must first train a generative model—reintroducing the very problem of limited data. Second, even when suitable foundational models are available, the computationally intensive nature of deep generative methods poses practical obstacles. Since effective augmentation often requires generating data that is several times larger than the original dataset, the cost of large-scale generation can quickly become prohibitive. Third, augmenting data with generative models raises concerns about their interpretability and controllability (Guidotti et al., 2018). Consequently, even when these models perform well, the lack of understanding of the underlying transformations of the augmented data makes it difficult to control the generated outputs, which poses a significant risk in the case of high-stakes scenarios (Rudin, 2019).

In this work, we propose a new data augmentation framework that addresses the challenges outlined above by providing a **learning-free**, **efficient**, and **controllable** algorithm applicable across diverse data modalities. Our approach builds on the well-established theory of *energy-based models* (Xie et al., 2016), together with recent advances in *log-linear models on partially ordered sets (posets)* (Sugiyama et al., 2016; 2017) and *information geometry* (Amari, 2016; Amari & Nagaoka, 2000; Ay et al., 2017). Conceptually, our framework resembles an autoencoder (Kingma & Welling, 2014). We begin by parametrizing data as discrete probability distributions on a *curved* statistical manifold $\mathcal{S}$. The data is then *encoded* into a chosen "latent space" $\mathcal{B} \subseteq \mathcal{S}$ via *forward projection*. Within this latent space, simple augmentation procedures informed by the encoded data are applied. Finally, the resulting "augmented representation" is *backward projected* to the local data space $\mathcal{D} \subseteq \mathcal{S}$, yielding new augmented data. Because our approach combines the linearity of projection with the nonlinearity induced by the curved structure of $\mathcal{S}$, we describe it as a form of *pseudo-nonlinear* data augmentation.

This design offers three key advantages. First, it is learning-free: the submanifold structure is constructed explicitly, allowing direct control over the properties of the augmented data without the

need to train a generative model. Second, it is computationally efficient: both forward and backward projections can be formulated as convex programs and solved with efficient first-order methods such as gradient descent. Third, it provides controllability: leveraging prior knowledge about relationships among features, one can adjust the choice of $\mathcal{S}$ and the submanifold of projection to tailor the statistical properties of the augmented data. Our contributions are summarized as follows:

- We propose a novel framework for modeling structured data (e.g., tensors) within a statistical manifold using energy-based models. This framework captures the intrinsic geometry of data and enables the design of geometry-aware algorithms.
- We develop the *pseudo-nonlinear* data augmentation algorithm under this framework. The method is **learning-free**, **efficient**, and **controllable**, and it applies broadly across different data modalities.
- We empirically validate the effectiveness of our approach, showing that it achieves competitive or superior performance compared to both generative-model-based baselines (e.g., autoencoders) and classical augmentation methods across multiple datasets and modalities.

2 Related work

2.1 Data augmentation

Data augmentation has proven to be effective in enhancing deep learning training by increasing dataset size, improving model robustness (Rebuffi et al., 2021), and introducing implicit regularization (Hernández-García & König, 2018). These techniques have been applied across various modalities, including text (Shorten et al., 2021; Feng et al., 2021; Li et al., 2022a) and images (Shorten & Khoshgoftaar, 2019; Mumuni & Mumuni, 2022; Wang et al., 2017). Although there are data augmentation methods that do not rely on generative models (Maharana et al., 2022; Zhang et al., 2018), these often depend on the knowledge of the underlying data generation mechanisms, which are typically unknown for uncommon modalities. As a result, much of the recent progress in data augmentation for general modalities has been driven by advancements in generative models, such as autoencoders (Kingma & Welling, 2014; Chadebec et al., 2022), generative adversarial networks (Antoniou, 2017), as well as diffusion models (Trabucco et al., 2024). Despite the progress, to date, there is no fully satisfactory solution for the two challenges (efficient and controllable) mentioned for generative-model-based data augmentation. For example, the design of controllable GANs is still evolving (Li et al., 2022b; She et al., 2021), and efficient flow-based models remain an active area of research (Geng et al., 2025). Moreover, these methods remain largely limited to specific domains, such as images, where classical data augmentation methods already exist.

2.2 Dimension reduction as data augmentation

Classical dimension reduction techniques, such as Principal Component Analysis (PCA) (Wold et al., 1987) and Singular Value Decomposition (SVD) (Stewart, 1993), work by identifying the optimal *linear* subspace that minimizes reconstruction error, typically through the orthogonal projection of data onto this subspace. These methods are straightforward, explicit, and also provide valuable geometric insights. For instance, PCA highlights the principal directions that capture the most variance in the data, uncovering important structural patterns.

However, one of the challenges in applying linear dimension reduction methods to data augmentation is the *inverse* problem, where reconstructing the original data from the space of reduced dimension is highly non-trivial. While some studies have explored indirect approaches to using linear dimension reduction for data augmentation (Abayomi-Alli et al., 2020; Sirakov et al., 2024), they are often application-specific and hard to generalize, limiting their broader applicability.

The *non-linear* generalizations, often called *manifold learning* (Meilă & Zhang, 2024), offer an alternative approach to dimension reduction. Popular methods like t-SNE (Hinton & Roweis, 2002; Van der Maaten & Hinton, 2008), Isomap (Tenenbaum et al., 2000), and UMAP (McInnes et al., 2018) are based on the manifold hypothesis, which suggests that high-dimensional data lie on a lower-dimensional manifold within the ambient space. The goal is to uncover this manifold and develop a smooth embedding that captures the intrinsic low-dimensional structure of the data.

In theory, manifold learning avoids the *inverse* problem by aiming to recover the underlying low-dimensional manifold of the data with near-zero information loss, making it conceptually appealing for data augmentation. However, this is rarely achieved in practice (Han et al., 2022); hence, solving the *inverse* problem is still necessary to generate realistic augmented data. Additionally, classical manifold learning methods are often limited to providing fixed embeddings for training data and cannot perform out-of-sample extensions (Duque et al., 2020), further limiting their ability to augment data. Recent approaches to address this limitation via either a more complicated algorithm or learning-based methods (Coifman & Lafon, 2006; Williams & Seeger, 2000; Vladymyrov & Carreira-Perpinán, 2013; Duque et al., 2020),

3 PRELIMINARY

3.1 DUALLY-FLATNESS IN INFORMATION GEOMETRY

Information geometry studies the structure of *statistical manifolds* $\mathcal{S}$ within the space of probability distributions. In this paper, we are primarily concerned with the space of an exponential family $\{p_\theta(x) \mid \theta \in \mathbb{R}^D\}$, where each p_θ denotes a probability density function parameterized by θ. We focus on the key concept in this field, *dually-flatness*, in this preliminary, while directing readers to Appendix A and Amari (2016) for more comprehensive details.

The starting point is the observation that the *log-partition function* $\psi(\theta)$ (also known as the *cumulant generating function* in statistics and *free energy* in physics) of an exponential family with density p_θ is convex in the *natural parameter* $\theta \in \mathbb{R}^D$. This convexity induces a natural coordinate system, θ, on $\mathcal{S}$, defining both the Riemannian metric $g = \nabla^2\psi(\theta)$ and the Bregman divergence (Bregman, 1967) $D_\psi(p_\theta, p_{\theta'})$. With these structures, the manifold $(\mathcal{S}, g)$ is flat, meaning that any curve $\theta(t) = at + b$ (where $a, b \in \mathbb{R}^D$ are constants) is a geodesic and lies entirely within $\mathcal{S}$. This flatness is known as *e-flatness*, and the geodesics are referred to as *e-geodesics* or *primal-geodesics*.

The dual structure arises from the *Legendre transform* (Legendre, 1787), which generates the dual function $\psi^*(\eta)$, where $\eta \in \mathbb{R}^D$ is the *expectation parameter*. This dual function is also convex, giving rise to the expectation coordinate system η, the dual Riemannian metric g^*, and also the dual Bregman divergence D_{ψ^*} which is the well-known Kullback-Leibler divergence D_{KL} (Eq.(3)). The corresponding flatness is termed *m-flatness*, with *m-geodesics* or *dual-geodesics* as its geodesics.

Remark 3.1. *An e-flat (m-flat) sub-manifold can be defined by forcing linear constraints on the θ coordinates (η coordinates) (Amari, 2016, Chapter 2).*

Dually-flatness emerges from the interplay between these two structures. Specifically, for any point (distribution) p in $\mathcal{S}$, there is a unique point p^* on an e-flat sub-manifold $\mathcal{B} \subseteq \mathcal{S}$ that minimizes the dual Bregman divergence $D_{\psi^*}(p, q) = D_{\mathrm{KL}}(p, q)$ (Amari, 2016, Theorem 1.5). This process, known as the *m-projection*, can be efficiently solved via convex optimization (Appendix A.2). The dual holds when switching e and m. Projection is a central tool in information geometry with profound implications for understanding the geometry of $\mathcal{S}$, which we will use later.

3.2 STATISTICAL MANIFOLD ON POSETS

A set Ω is a *partially ordered set (poset)* if it is equipped with a *partial order* "$\leq$", a relation satisfying the following for all $x, y, z \in \Omega$: 1.) $x \leq x$ (reflexivity); 2.) $x \leq y$ and $y \leq x$ implies $x = y$ (antisymmetry); 3.) $x \leq y$ and $y \leq z$ implies $x \leq z$ (transitivity). We focus on finite posets Ω with a bottom element $\perp$ such that $\perp \leq x$ for all $x \in \Omega$.

Given such a poset Ω, consider a discrete random variable X with finite support Ω with its probability mass function $p\colon \Omega \to \mathbb{R}_{\geq 0}$, $p(x) = \Pr(X = x)$ for $x \in \Omega$. For a discrete probability distribution p over a poset Ω, the *log-linear model on posets* recursively defines $\theta\colon \Omega \to \mathbb{R}$ as $\log p(x) =: \sum_{y \leq x} \theta(y)$ for all $x \in \Omega$ (Sugiyama et al., 2017). Intuitively, one can think of $\theta(x)$ for each $x \in \Omega$ as specifying the energy for x that correctly represents $p(x)$, where the dependence between θ's on different elements depends on the poset structure. This model belongs to the exponential family, with θ corresponding to the natural parameters, except for $\theta(\perp)$ which coincides with the partition function. Thus, all discrete probability distributions over Ω form a $(|\Omega| - 1)$-

dimensional dually-flat statistical manifold $\mathcal{S} \coloneqq \{p\colon \Omega \to \mathbb{R}_{\geq 0} \mid \sum_{x\in\Omega} p(x) = 1\}$, with dual coordinate systems (θ, η) defined by the poset structure.

Prior studies (Sugiyama et al., 2017; 2018; Ghalamkari et al., 2024) have shown that log-linear models on posets provide an effective way to represent data. In particular, by incorporating the underlying geometric relationships among features, these models yield a meaningful curved statistical space that captures the structure of the data more faithfully than flat representations.

4 PSUEDO-NONLINEAR DATA AUGMENTATION

We first present our proposed framework in Section 4.1 and the projection algorithms in Section 4.2, then we combine and apply them to data augmentation in Section 4.3. Finally, we discuss two important features of the proposed method regarding **controllability** and **efficiency** in Section 4.4. Throughout this section, we will use *positive tensors* as our running example (Example 4.1).

4.1 LOG-LINEAR MODEL ON POSETS FRAMEWORK

Given a dataset $\{z_i\}_{i=1}^n$, our proposed framework embeds the data into a statistical manifold $\mathcal{S}$ by leveraging the log-linear model on posets, which provides a geometric structure induced by the energy-based modeling. The process works in three steps: 1.) models each z_i as a *real-valued poset*; 2.) embeds the real-valued poset into the statistical manifold $\mathcal{S}$ by viewing it as a probability distribution; 3.) computes the corresponding two coordinate representations using the log-linear model on posets. See Figure 1 for an illustration. We now explain each step in detail below.

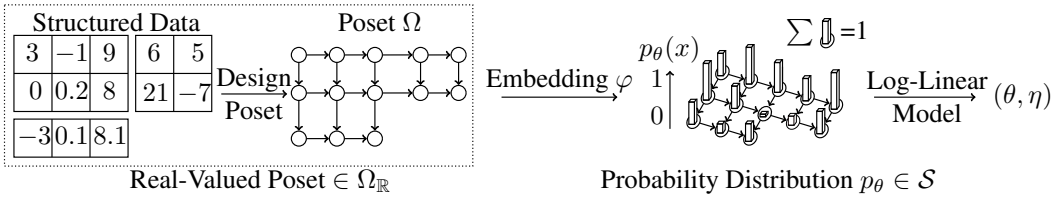


Figure 1: Given structured data, we design a poset Ω that reflects the structure and embed the resulting real-valued poset as a discrete probability distribution $p_\theta(x)$ via a natural embedding φ into the statistical manifold $\mathcal{S}$. Then the log-linear model on posets provides the dually-flat coordinates (θ, η) for p_θ, which can be efficiently calculated (Section 3.2).

Real-Valued Poset. In the usual machine learning pipeline, inputs are often constrained to be vectors or matrices, failing to deal with more complex data. In contrast, posets are flexible enough to capture data with structures, including vectors, matrices, or tensors. For instance, given a D-dimensional vector $z \in \mathbb{R}^D$ (i.e., 1^{st}-order tensor), the underlying data structure obtained by omitting the feature associated with each entry (dimension) can be modeled by the poset $\Omega \coloneqq [D]$ with the partial order being the natural order between positive integers. Similarly, other common data structures, such as matrices or tensors, can be treated in the same way. In general, any data structure that admits a natural partial order can be modeled by a poset.

With the features associated with each entry in the data structure, we can define the *real-valued poset*, which is a mapping from the poset Ω to the set of real numbers $\mathbb{R}$ such that each entry (element) of the data structure (poset) $x \in \Omega$ is associated with a feature in $\mathbb{R}$. We denote the set of real-valued posets as $\Omega_{\mathbb{R}}$. In the D-dimensional vector example, $\Omega = [D]$, each element $x \in \Omega$ corresponds to one of the D dimensions. Associating a real number to each dimension clearly corresponds to an element in $\Omega_{\mathbb{R}}$.

Natural Embedding. To embed the data $\{z_i \in \Omega_{\mathbb{R}}\}_{i=1}^n$, which are now modeled as real-valued posets, to the statistical manifold $\mathcal{S}$ which concerns with discrete probability distributions, we want an embedding $\varphi\colon \Omega_{\mathbb{R}} \to \mathcal{S}$ such that $\sum_{x\in\Omega}(\varphi(z_i))_x = 1$ for all z_i with $\dim(\mathcal{S}) = D - 1$.[1] In

[1]One can also consider the manifold of positive measures of dimension D and avoid the potential scaling issues. For simplicity, we omit this trivial extension in the presentation.

other words, $\varphi(z_i)$ gives a probability mass function of a discrete random variable over the poset, where $(\varphi(z_i))_x$ is the probability of sampling $x \in \Omega$ when sampled from $\varphi(z_i)$. From the perspective of energy-based modeling, φ is oftentimes naturally induced, e.g., for tabular frequency data. Moreover, φ often admits a natural inverse φ^{-1}, or an empirical approximation based on the data.

Dually-Flat Coordinates. From the log-linear model on posets introduced in Section 3.2, for each $z_i' \coloneqq \varphi(z_i) \in \mathcal{S}$, we can associate the dually-flat coordinates $\theta(z_i') \in \mathbb{R}^{D-1}$ and $\eta(z_i') \in \mathbb{R}^{D-1}$. Such coordinate systems are with respect to the underlying poset structure Ω and give a non-trivial, energy-based geometric structure of the dataset.

Example 4.1 (Positive tensor)**.** *A d^{th}-order* tensor $T \in \mathbb{R}^{I_1 \times \cdots \times I_d} =: \mathbb{R}^D$ *is a multidimensional array with real entries for every index vector $v = (i_1, \ldots, i_d) \in [I_1] \times \cdots \times [I_d] =: \Omega$ where for each k, $[I_k] \coloneqq \{1, 2, \ldots, I_k\}$ for a positive integer I_k. Tensors with entries all being positive are called positive tensors, denoted as $P \in \mathbb{R}_{\geq 0}^{I_1 \times \cdots \times I_d}$.*

For tensors, a natural partial order "$\leq$" one can impose on Ω between two index vectors $v = (i_1, \ldots, i_d)$, $w = (j_1, \ldots, j_d)$ is that $v \leq w$ if and only if $i_k \leq j_k$ for all $k = 1, \ldots, d$. Finally, for positive tensors, a simple embedding $\varphi \colon \mathbb{R}_{\geq 0}^{I_1 \times \cdots \times I_d} \to \mathcal{S}$ where $P' \coloneqq \varphi(P) \colon \Omega \to \mathbb{R}_{\geq 0}$ such that $P'_v \coloneqq P_v / \sum_{w \in \Omega} P_w$ for all $v \in \Omega$ can be defined with a natural empirical inverse (see Remark 4.3).

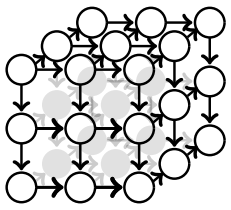


Figure 2: Natural poset structure of 3rd-order tensors in $\mathbb{R}^{3 \times 3 \times 3}$.

Example 4.1 illustrates how the framework applies to common data modalities. For example, a color image can be represented as a 3rd-order tensor, where the first two dimensions correspond to height and width and the third dimension encodes color channels. More generally, the log-linear model on posets can represent a wide range of "structured data," including 2-dimensional matrices (2nd-order tensors), vector-valued data (1st-order tensors), or, more broadly, any data where relationships among features exhibit a structured form. While this flexibility is a strength, it is also important to clarify a structural limitation of the framework to delineate the boundaries of its applicability.

Remark 4.2. *A key limitation of the framework is its reliance on a partial order over the index set. This dependence makes it difficult to model invariances under index permutations, which introduces bias into the model. However, a key advantage is that we can explicitly recognize how this bias arises and, if desired, control it by adjusting the partial order. Moreover, when data inherently have a partial (or total) order, as in directed graphs, images, or time series, this dependence is not a limitation but a natural advantage.*

4.2 Forward and Backward Projection

We now demonstrate how to incorporate projection theory to conduct data augmentation. Our algorithm mimics the architecture of autoencoders, focusing on two of the central building blocks: the *encoder* $\mathsf{Enc}(\cdot)$ and the *decoder* $\mathsf{Dec}(\cdot)$. First, for the encoding step, we formally explain how projection theory can be applied to perform dimension reduction and obtain compact representations within our framework. Next, for the decoding step, we introduce our proposed algorithm, termed *backward projection*, which serves as the *inverse* of dimension reduction. Figure 3 gives an intuitive geometric picture for our proposed algorithms.

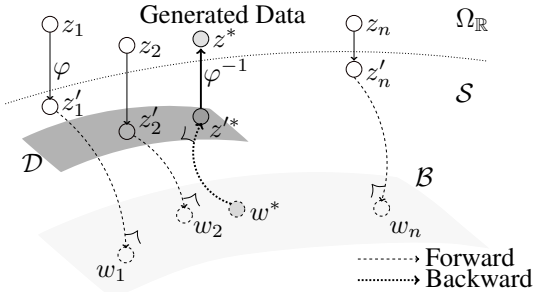


Figure 3: Illustration of forward and backward projection. Here, w_i: latent representation; of the original data z_i, obtained from forward projection to $\mathcal{B}$; w^*: generated latent representation; Here, $w^* \mapsto z'^*$: backward projection to $\mathcal{D}$, obtained from the original data of the nearest neighbor(s) of w^* in the latent space.

Forward Projection. The embedding from $\Omega_{\mathbb{R}}$ to the statistical manifold $\mathcal{S}$ introduced in Section 4.1 maintains the dimensionality. To achieve dimension reduction, we leverage the projection theory: by projecting $z_i' = \varphi(z_i)$ onto a low-dimensional flat sub-manifold called *base sub-manifold* $\mathcal{B} \subseteq \mathcal{S}$ with $\dim(\mathcal{B}) \ll \dim(\mathcal{S})$, we obtain the desired *encoding* $\mathsf{Enc} \coloneqq \mathrm{Proj}_{\mathcal{B}} \circ \varphi \colon \Omega_{\mathbb{R}} \to$

$\mathcal{B}$ that maps the data to a low-dimensional latent representation manifold. Note that the encoding $\mathsf{Enc}(\cdot)$ is smooth and well-defined as the projection is unique when $\mathcal{B}$ is flat and minimizing either the primal or the dual Bregman divergence, depending on either $\mathcal{B}$ is e- or m-flat.

Backward Projection. One of the technical burdens is that the encoding $\mathsf{Enc}(\cdot)$ is not invertible, hence a perfect decoding $\mathsf{Dec}(\cdot)$ is mathematically impossible, even when $\mathsf{Enc}(\cdot)$ only involves a simple linear projection in Euclidean space. Here, we propose a simple, geometrically intuitive, and data-centric solution that aims to find the *inverse* of the projection with theoretical guarantees.

The high-level intuition is simple: we assume that similar data will result in similar projections. Hence, given a point in the low-dimensional latent representation space, we try to "project it back" to approximate the original dataset by exploiting the fact that we have access to the inverse of the dataset's projection, which is the dataset itself. Specifically, we can artificially create a sub-manifold $\mathcal{D}$ around a subset of the dataset that captures the local geometric structure of the dataset around the latent representation, and *backward* project onto it.

Formally, assuming that we have access to the embedded dataset $\{z'_i = \varphi(z_i)\}_{i=1}^n$ and their projected result $\{w_i = \mathrm{Proj}_{\mathcal{B}}(z'_i)\}_{i=1}^n$ for some flat base sub-manifold $\mathcal{B}$. To find the inverse of some given point $w^* \in \mathcal{B}$ assuming it comes from the projection on $\mathcal{B}$, we first find w^*'s k-nearest neighbors among w_i's, obtaining a size k index set $N \subseteq [n]$ with $|N| = k$. Then we create a flat sub-manifold $\mathcal{D}$ called *local data sub-manifold* based on the pre-images z'_i's of these w_i's, and project w^* on $\mathcal{D}$ to obtain the *inverse* $z'^* = \mathrm{Proj}_{\mathcal{B}}^{-1}(w^*) \coloneqq \mathrm{Proj}_{\mathcal{D}}(w^*)$.

The construction of $\mathcal{D}$ can be arbitrary, in particular, one can easily control the degree of freedom of the resulting z'^*: for instance, from Remark 3.1, given the nearest neighbor $z'_{i^\star}$, one can define an e-flat $\mathcal{D} \coloneqq \{\theta \in \mathbb{R}^{\dim(\mathcal{S})} \mid (\theta)_x = \big(\theta(z'_{i^\star})\big)_x \text{ for some } x \in \Omega\}$ by fixing some indexes of θ to be the corresponding θ-coordinate values of z'_{i^*}.

Algorithm 1 in Appendix B summarizes this procedure, which we termed *backward projection*. With access to $\mathrm{Proj}_{\mathcal{B}}^{-1}(\cdot)$, decoding is simply $\mathsf{Dec} \coloneqq \varphi^{-1} \circ \mathrm{Proj}_{\mathcal{B}}^{-1} \colon \mathcal{B} \to \Omega_{\mathbb{R}}$. Algorithm 1 is a geometrically intuitive, data-centric algorithm with desirable theoretical guarantees such as divergence minimizing when projecting on the constructed local data sub-manifold $\mathcal{D}$.

4.3 Psuedo-Nonlinear data augmentation

With all the building blocks in place, we can now formally describe the proposed data augmentation algorithm, which consists of: 1.) encoding, 2.) augmenting, and 3.) decoding.

Encoding. As described in Section 4.2, the encoding $\mathsf{Enc} \coloneqq \mathrm{Proj}_{\mathcal{B}} \circ \varphi$ is simply a combination of the natural embedding followed by a projection. Notation-wise, we write $w_i \coloneqq \mathsf{Enc}(z_i)$.

Augmenting. To generate an augmented data z^*, we first generate a new representation w^* in the latent space, which in our case, is a pre-specified flat base sub-manifold $\mathcal{B}$. This w^* can be generated in various ways, such as controlled perturbations of the original representations or a linear mixture of two arbitrary original representations.

Decoding. As described in Section 4.2, the decoding $\mathsf{Dec} \coloneqq \varphi^{-1} \circ \mathrm{Proj}_{\mathcal{B}}^{-1}$ is simply a combination of backward projection (Algorithm 1) with the inverse of the natural embedding. Notation-wise, we write $z^* \coloneqq \mathsf{Dec}(w^*) = \varphi^{-1}(z'^*)$ where $z'^* \coloneqq \mathrm{Proj}_{\mathcal{B}}^{-1}(w^*) \coloneqq \mathrm{Proj}_{\mathcal{D}}(w^*)$.

The proposed method integrates the *nonlinear* forward and backward projections as encoding and decoding, which we summarize the above in Algorithm 2 in Appendix B with an illustration given by Figure 3.

Remark 4.3 (Positive tensor)**.** *With Algorithm 2, the empirical inverse φ^{-1} for the positive tensors in Example 4.1 can be naturally defined as the* inverse of the average of original scaling among the nearest neighbors*.*

4.4 Sub-manifolds for positive tensors

It is evidence that the proposed method is **learning-free**. In this section, we describe how to construct flat sub-manifolds (for $\mathcal{B}$ and $\mathcal{D}$) to **control** the augmentation process, and how these submanifolds naturally admit **efficient projection algorithms**. For clarity and concreteness, we focus our

discussion on the case of positive tensors, while noting that the general principles and arguments extend to broader settings.

Designing Sub-Manifolds. We start by discussing an intrinsic trade-off of choosing the dimension of $\mathcal{B} \subseteq \mathcal{S}$ we aim to forward project on. It is clear that more information of the data is preserved after forward projection onto $\mathcal{B}$ as $\dim(\mathcal{B})$ increases. Hence, the quality of the backward projection $\mathrm{Proj}_{\mathcal{B}}^{-1}(\cdot)$ (Algorithm 1) increases along with $\dim(\mathcal{B})$. However, in the extreme case when $\dim(\mathcal{B}) \approx \dim(\mathcal{S})$, Algorithm 2 becomes less effective as the augmenting step now suffers from the curse of dimensionality. Previous studies on such a trade-off of choosing $\dim(\mathcal{B})$ (Sugiyama et al., 2018; Ghalamkari et al., 2024) reveal how one should construct flat base sub-manifolds. In particular, the *many-body tensor approximation* (Ghalamkari et al., 2024; Derun & Sugiyama, 2025) aims to capture a *hierarchy* of mode interactions with different $\dim(\mathcal{B})$ for positive tensors within the log-linear model on posets. Specifically, the ℓ-body approximation considers projection on the following sub-manifold:

$$\mathcal{M}_\ell := \{\theta \in \mathbb{R}^{\dim(\mathcal{S})} \mid \theta_x = 0 \text{ for all } \textbf{non } \ell\text{-body parameters } x \in \Omega\}, \tag{1}$$

where the ℓ-body parameter corresponds to ℓ non-one indices, acting as a generalization of one-body and two-body parameters (Ghalamkari et al., 2024). Intuitively speaking, an ℓ-body parameter captures the interaction among ℓ different modes. Hence, when $\mathcal{B} = \mathcal{M}_\ell$, all interactions between modes of orders higher than ℓ are neglected. This approach provides a principled way of designing the latent space with a clear understanding of what each dimension signifies.

On the other hand, a dual-like trade-off exists for the local data sub-manifold $\mathcal{D}$. Recall that the goal of backward projection is to project back to the "local data" space $\mathcal{D}$ given by a set N of k nearest neighbors of a generated latent representation. When $\dim(\mathcal{D})$ increases, backward projecting onto $\mathcal{D}$ has a higher degree of freedom, which is desirable for data augmentation. However, in the extreme case when $\dim(\mathcal{D}) \approx \dim(\mathcal{S})$, the backward projection becomes unconstrained and potentially generates gibberish results. Hence, a natural construction of $\mathcal{D}$ is to consider the "dual" notion of $\mathcal{M}_\ell$, where we now allow all **non** ℓ-body parameters to vary while fixing every ℓ-body parameter to be the average of the θ values among N:

$$\mathcal{M}_\ell^*(N) := \left\{ \theta \in \mathbb{R}^{\dim(\mathcal{S})} \mid \theta_x = \frac{1}{k} \sum_{i^* \in N} \big(\theta(z'_{i^*})\big)_x \text{for all } \ell\text{-body parameters } x \in \Omega \right\}. \tag{2}$$

These two constructions offer a practical design choice for Algorithm 2 while providing the desired properties. For instance, By choosing an appropriate ℓ, both $\mathcal{M}_\ell$ and $\mathcal{M}_\ell^*$ can capture specific information with desired degree of freedom.

Efficient Projection. For sub-manifolds $\mathcal{M}_\ell$ designed for many-body approximation with B many non-fixed indexes (i.e., ℓ-body parameters), the projection can be efficiently computed via formulating the projection as a convex program that can be solved via gradient descent in polynomial time. Note that the gradient of the convex program has a closed-form, which makes the optimization extremely efficient Appendix A.2. Finally, we note that the projection admits other desirable theoretical guarantees (e.g., minimizing the KL-divergence and ensuring uniqueness) and efficient algorithmic implementation. We refer the reader to Appendix A.2 for a detailed discussion.

5 Experiments

We conduct a series of experiments to demonstrate the efficacy of our proposed method.

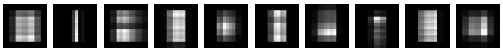
5.1 Setup

We consider classification task on the image (MNIST (LeCun, 1998) and CIFAR-10 (Krizhevsky & Hinton, 2009)), speech (Speech Commands (Warden, 2018)), and tabular data (Connectionist Bench (Sejnowski & Gorman, 1988), Taiwanese Bankruptcy (Journal, 2020), and Wine Quality (Cortez et al., 2009)). For Algorithm 2, we first apply the log-linear model on posets for positive tensors (Example 4.1) by normalizing features to be positive and reshaping the features as a tensor of suitable dimensions. More details and explanations can be found in Appendix C.1.

5.2 CONTROLLABILITY WITH CHOICES OF SUB-MANIFOLDS

As discussed in Section 4.4, constructing the sub-manifold carefully provides the essential **controllability**. We demonstrate this with MNIST and CIFAR due to their simplicity. 1.) For MNIST, we let $\mathcal{B} = \mathcal{M}_1$ and $\mathcal{D} = \mathcal{M}_1^*$, which correspond to preserving *shape* information. 2.) For CIFAR, on the other hand, we first carefully *reshape* the colored images to higher-order tensors, and let $\mathcal{B} = \mathcal{M}_5$ and $\mathcal{D} = \mathcal{M}_4^*$, which correspond to preserving *fine-grained collective shape and color* information. Figures 4(a) and 4(d) show the results of the forward projection, while Figures 4(b) and 4(e) show the results of the backward projection of the latent representations sampled from the kernel density model M fitted on the results (θ-coordinates) of Figures 4(a) and 4(d), respectively.

For MNIST, we see that the augmentation results (Figure 4(b)) with backward projection successfully reconstruct the digit structures, indicating that the essential *shape* information is indeed preserved and separated in the latent space $\mathcal{B}$ to provide non-trivial neighbor information for constructing a sufficiently good $\mathcal{D}$ for backward projection. Note that the local data sub-manifold $\mathcal{D}$ has a dimension of 767, indicating a high degree of freedom for backward projection.

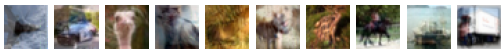


(a) Forward projection on $\mathcal{B}$ with $\dim(\mathcal{B}) = 17$.

(b) Backward projection on $\mathcal{D}$ with $\dim(\mathcal{D}) = 767$.

(c) The closest training data of Figure 4(b).

(d) Forward projection on $\mathcal{B}$ with $\dim(\mathcal{B}) = 1410$.



(e) Backward projection on $\mathcal{D}$ with $\dim(\mathcal{D}) = 2334$.

(f) The closest training data of Figure 4(e).

Figure 4: Results of MNIST (Left) and CIFAR-10 (Right) for Algorithm 2.[2]

More interesting results for CIFAR-10 are shown in Figures 4(d) to 4(f). By our proposed projection-based augmentation method, the *fine-grained shape and color* information is preserved. For instance, the third image, ostrich, successfully preserves the fine-grained shape and color relationship (e.g., colors for eyes and beak, and small pink flowers in the background), while the crude shape-to-color information is lost (e.g., colors for the background without shape details shift noticeably). The same trend can be observed consistently, validating the proposed method's efficacy.

In practice, by carefully reshaping the data into higher-order tensors such that some modes of the tensors contain the essential relationship between features that one wishes to control, with many-body approximation, it is possible to construct suitable sub-manifolds that preserve the chosen information, providing a controllable augmentation of the original data via simple projection operations.

5.3 CLASSIFICATION PERFORMANCE

We evaluate our method on downstream classification tasks. For each dataset, we train a classifier on both the original training set and an augmented training set, where the augmented portion corresponds to 20% of the original training size and is generated from the training data. The classifiers used are ResNet-18 (He et al., 2016) for CIFAR-10, M5 (Dai et al., 2017) for SpeechCommands, and a simple MLP for the remaining datasets. Details of the training setup are provided in Appendix C.1.

We compare our approach against two baselines: a generative-model-based method and classical learning-free methods. Specifically, the compared methods are: 1.) pseudo-nonlinear (**PNL**, ours), 2.) autoencoder-based augmentation (**AE**),[3] and 3.) standard augmentation (**STD**). For images, **STD** includes standard techniques such as random cropping, flipping, rotations, and affine transforma-

[2]We remark that the forward projection (first row) is different from the augmentation result (second row): the backward projection is from a *sampled* latent representation, while the first row shows some representative latent representations from the *dataset*.

[3]More sophisticated generative-model-based methods (e.g., diffusion-based augmentation (Trabucco et al., 2024)) are generally computationally infeasible at our scale, where we target around 10,000 augmented samples per experiment. For reference, diffusion-based approaches are typically applied in a "few-shot" setting with

tions. For speech, **STD** combines random volume scaling, time stretching, MelSpectrogram conversion, frequency masking, and time masking (Park et al., 2019). For other data types, **STD** is implemented as Gaussian noise perturbation.

We denote the original dataset as **OG**, and the augmented version using a method **AG** as $\mathbf{OG}^{\mathbf{AG}}$. Results are summarized in Table 1, where each model is trained on its respective training set and evaluated on 20 randomly bootstrapped test subsets, each containing 50% of the original test data.

Table 1: Test accuracy of classifiers trained on different datasets.

Training Set	**Dataset**					
	MNIST	CIFAR-10	Speech Commands	Connectionist Bench	Taiwanese Bankruptcy	Wine Quality
OG	$98.29 \pm 0.16\%$	$89.08 \pm 0.32\%$	$84.56 \pm 0.62\%$	$78.81 \pm 11.22\%$	$96.59 \pm 0.65\%$	$56.90 \pm 1.55\%$
$\mathbf{OG}^{\mathbf{STD}}$	$97.90 \pm 0.21\%$	$90.31 \pm 0.36\%$	$84.26 \pm 0.44\%$	$75.00 \pm 9.97\%$	$96.36 \pm 0.65\%$	$57.07 \pm 2.10\%$
$\mathbf{OG}^{\mathbf{PNL}}$	$98.03 \pm 0.17\%$	$84.57 \pm 0.70\%$	$86.32 \pm 0.41\%$	$80.00 \pm 6.49\%$	$96.61 \pm 0.67\%$	$58.12 \pm 2.05\%$
$\mathbf{OG}^{\mathbf{AE}}$	$98.09 \pm 0.18\%$	$85.43 \pm 0.50\%$	$86.03 \pm 0.42\%$	$79.29 \pm 9.44\%$	$95.85 \pm 0.62\%$	$56.61 \pm 1.84\%$

We see that in most cases, the classifier trained on the augmented data indeed achieves a better prediction accuracy compared to the one trained only on **OG**. One exception is CIFAR-10, where both $\mathbf{OG}^{\mathbf{PNL}}$ and $\mathbf{OG}^{\mathbf{AE}}$ perform worse than **OG** and $\mathbf{OG}^{\mathbf{STD}}$. We suspect that this is because of the visual complexity of CIFAR-10, and the augmented datasets $\mathbf{OG}^{\mathbf{PNL}}$ and $\mathbf{OG}^{\mathbf{AE}}$ act more like regularizers, while $\mathbf{OG}^{\mathbf{STD}}$ explicitly forces the classifier to learn the correct visual representation. In all cases, **PNL** consistently outperforms **AE**, demonstrating a competitive performance.

Importantly, we also emphasize stability, a crucial but often overlooked goal in data augmentation. This is best illustrated by the Connectionist Bench dataset, which contains only 208 data points and 60 features, posing a significant challenge for consistent generalization. From Table 1, the standard deviations in accuracy on this dataset typically hover around 10%, regardless of the model. In contrast, our method achieves a substantially lower standard deviation of 6.49%, indicating improved consistency across runs.

5.4 Additional Experiments

We conduct a series of additional ablation studies and experiments in Appendices C.2 to C.5. Specifically, Appendices C.2 and C.3 assess the robustness of our proposed method and the impact of augmentation on downstream task performance. In contrast, Appendix C.4 provides a justification for the necessity of forward projection. Finally, Appendix C.5 explores the effect of different latent space design choices on augmentation outcomes, offering insights into how these design decisions can be leveraged to better control the augmentation process.

6 Conclusion

In this paper, we introduced the *pseudo-nonlinear data augmentation* framework, which leverages information geometry and energy-based models to provide a **learning-free**, **efficient**, and **controllable** augmentation method. Our approach, grounded in the log-linear model on posets, endows data with a rich information-geometric structure that facilitates both geometric reasoning and principled algorithm design. A key component is the backward projection algorithm, which reverses dimension reduction in a geometrically intuitive and data-centric way.

Through extensive experiments, we demonstrated the effectiveness of our method across diverse modalities and datasets. In particular, it enables scalable augmentation for general data types while offering controllability via the design of 1.) the base submanifold $\mathcal{B}$, 2.) the local data submanifold $\mathcal{D}$, and 3.) the poset structure Ω. Empirically, our framework outperforms both generative-model-based augmentation baselines and classical standard methods, especially on common modalities such as images and speech.

only around 15 augmented images per experiment. Thus, we select autoencoder-based augmentation as a practical generative baseline.

Ethics Statement

We have carefully reviewed and adhered to the ICLR Code of Ethics. This work does not involve human subjects, personally identifiable information, or sensitive data. All datasets used in our experiments are publicly available and commonly used in the community. We have followed standard practices for dataset preprocessing and model evaluation to avoid introducing unfair bias. To the best of our knowledge, our methodology and findings do not pose potential harm, nor do they raise concerns regarding privacy, security, discrimination, or misuse.

Reproducibility Statement

We are committed to ensuring the reproducibility of our results. The main paper provides a detailed description of the proposed method and the experimental setup. All hyperparameters, training procedures, and evaluation metrics are documented in Appendix C.1. To further facilitate reproducibility, we provide the source code and instructions for reproducing all experiments.

References

Olusola Oluwakemi Abayomi-Alli, Robertas Damaševičius, Michał Wieczorek, and Marcin Woźniak. Data augmentation using principal component resampling for image recognition by deep learning. In *Artificial Intelligence and Soft Computing: 19th International Conference, ICAISC 2020, Zakopane, Poland, October 12-14, 2020, Proceedings, Part II 19*, pp. 39–48. Springer, 2020.

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Shun-ichi Amari. *Information Geometry and Its Applications*, volume 194 of *Applied Mathematical Sciences*. Springer Japan, 2016. ISBN 978-4-431-55977-1 978-4-431-55978-8. doi: 10.1007/978-4-431-55978-8.

Shun-ichi Amari and Hiroshi Nagaoka. *Methods of information geometry*, volume 191. American Mathematical Soc., 2000.

A Antoniou. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

Nihat Ay, Jürgen Jost, Hông Vân Lê, and Lorenz Schwachhöfer. *Information Geometry*, volume 64 of *Ergebnisse Der Mathematik Und Ihrer Grenzgebiete 34*. Springer International Publishing, 2017. ISBN 978-3-319-56477-7 978-3-319-56478-4. doi: 10.1007/978-3-319-56478-4.

Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967.

Clément Chadebec, Elina Thibeau-Sutre, Ninon Burgos, and Stéphanie Allassonnière. Data augmentation in high dimensional low sample size setting using a geometry-based variational autoencoder. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):2879–2896, 2022.

Ronald R Coifman and Stéphane Lafon. Geometric harmonics: a novel tool for multiscale out-of-sample extension of empirical functions. *Applied and Computational Harmonic Analysis*, 21(1): 31–52, 2006.

Paulo Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Wine Quality. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C56S3T.

Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das. Very deep convolutional neural networks for raw waveforms. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 421–425. IEEE, 2017.

Zhou Derun and Mahito Sugiyama. Optimal submanifold structure in log-linear models. In *The 41st Conference on Uncertainty in Artificial Intelligence*, 2025. URL https://openreview.net/forum?id=kVC4QV6Q4b.

Andrés F Duque, Sacha Morin, Guy Wolf, and Kevin Moon. Extendable and invertible manifold learning with geometry regularized autoencoders. In *2020 IEEE International Conference on Big Data (Big Data)*, pp. 5027–5036. IEEE, 2020.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for NLP. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 968–988, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.84.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

Kazu Ghalamkari, Mahito Sugiyama, and Yoshinobu Kawahara. Many-body approximation for non-negative tensors. *Advances in Neural Information Processing Systems*, 36, 2024.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

Henry Han, Wentian Li, Jiacun Wang, Guimin Qin, and Xianya Qin. Enhance explainability of manifold learning. *Neurocomputing*, 500:877–895, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Alex Hernández-García and Peter König. Further advantages of data augmentation on convolutional neural networks. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*, pp. 95–103. Springer, 2018.

Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. In S. Becker, S. Thrun, and K. Obermayer (eds.), *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002.

Taiwan Economic Journal. Taiwanese Bankruptcy Prediction. UCI Machine Learning Repository, 2020. DOI: https://doi.org/10.24432/C5004D.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Yann LeCun. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

John M Lee. *Smooth manifolds*. Springer, 2012.

Adrien Marie Legendre. *Mémoire sur l'intégration de quelques équations aux différences partielles*. Imprimerie royale, 1787.

Bohan Li, Yutai Hou, and Wanxiang Che. Data augmentation approaches in natural language processing: A survey. *Ai Open*, 3:71–90, 2022a.

Chao Li, Kelu Yao, Jin Wang, Boyu Diao, Yongjun Xu, and Quanshi Zhang. Interpretable generative adversarial networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(2):1280–1288, Jun. 2022b. doi: 10.1609/aaai.v36i2.20015.

Kiran Maharana, Surajit Mondal, and Bhushankumar Nemade. A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, 3(1):91–99, 2022.

L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*, February 2018.

Marina Meilă and Hanyu Zhang. Manifold learning: What, how, and why. *Annual Review of Statistics and Its Application*, 11, 2024.

Alhassan Mumuni and Fuseini Mumuni. Data augmentation: A comprehensive survey of modern approaches. *Array*, 16:100258, 2022.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech 2019*, Sep 2019. doi: 10.21437/interspeech.2019-2680. URL http://dx.doi.org/10.21437/Interspeech.2019-2680.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A Mann. Data augmentation can improve robustness. *Advances in Neural Information Processing Systems*, 34:29935–29948, 2021.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

Terry Sejnowski and R. Gorman. Connectionist Bench (Sonar, Mines vs. Rocks). UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C5T01Q.

Rui She, Pingyi Fan, Xiao-Yang Liu, and Xiaodong Wang. Interpretable generative adversarial networks with exponential function. *IEEE Transactions on Signal Processing*, 69:3854–3867, 2021.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. Text data augmentation for deep learning. *Journal of big Data*, 8(1):101, 2021.

Nikolay Metodiev Sirakov, Tahsin Shahnewaz, and Arie Nakhmani. Training data augmentation with data distilled by principal component analysis. *Electronics*, 13(2):282, 2024.

Gilbert W Stewart. On the early history of the singular value decomposition. *SIAM review*, 35(4): 551–566, 1993.

Mahito Sugiyama, Hiroyuki Nakahara, and Koji Tsuda. Information decomposition on structured space. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pp. 575–579, 2016. doi: 10.1109/ISIT.2016.7541364.

Mahito Sugiyama, Hiroyuki Nakahara, and Koji Tsuda. Tensor Balancing on Statistical Manifold. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3270–3279. PMLR, 2017.

Mahito Sugiyama, Hiroyuki Nakahara, and Koji Tsuda. Legendre decomposition for tensors. *Advances in Neural Information Processing Systems*, 31, 2018.

Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimension reduction. *Science*, 290(5500):2319–2323, 2000. doi: 10.1126/science.290.5500.2319.

Brandon Trabucco, Kyle Doherty, Max A Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ZWzUA9zeAg.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

Max Vladymyrov and Miguel Á Carreira-Perpinán. Locally linear landmarks for large-scale manifold learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*, pp. 256–271. Springer, 2013.

Jason Wang, Luis Perez, et al. The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit*, 11(2017):1–8, 2017.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Christopher Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. *Advances in neural information processing systems*, 13, 2000.

Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

Sebastien C Wong, Adam Gatt, Victor Stamatescu, and Mark D McDonnell. Understanding data augmentation for classification: when to warp? In *2016 international conference on digital image computing: techniques and applications (DICTA)*, pp. 1–6. IEEE, 2016.

Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *International conference on machine learning*, pp. 2635–2644. PMLR, 2016.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

A PROJECTION THEORY IN INFORMATION GEOMETRY

We will assume some familiarity with the basic terminologies for manifold (Lee, 2012, Chapter 1, 4). In particular, in this section, we explain the main concepts of information geometry used in this study, including natural parameters, expectation parameters, model flatness, and convexity of optimization. In the following, we consider only discrete probability distributions for simplicity and refer to Amari (2016) for more general cases.

A.1 (θ, η)-COORDINATE AND GEODESICS

Consider $\mathcal{S}$ as the space of discrete probability distributions, which is a non-Euclidean space with the Fisher information matrix G as the metric. This metric measures the distance between two points, i.e., discrete probability distributions, in $\mathcal{S}$. In Euclidean space, the shortest path between two points is a straight line, while in a non-Euclidean space, such a shortest path is called a *geodesic*. In the space $\mathcal{S}$, two kinds of geodesics can be introduced: e-geodesics and m-geodesics. For two points $p_1, p_2 \in \mathcal{S}$, e- and m-geodesics are defined as

$$\{r_t \mid \log r_t = (1-t)\log p_1 + t \log p_2 - \phi(t), 0 \leq t \leq 1\}, \quad \{r_t \mid r_t = (1-t)p_1 + tp_2, 0 \leq t \leq 1\},$$

respectively, where $\phi(t)$ is a normalization factor to keep r_t to be a distribution.

We can parameterize distributions $p \in \mathcal{S}$ by parameters known as *natural parameters*. In Section 3.2, we have described the relationship between a distribution p and a natural parameter vector $\theta \in \mathbb{R}^{D-1}$ for a discrete probability distribution over a sample space of D elements in the log-linear model. The natural parameter θ serves as a coordinate system of $\mathcal{S}$, since any distribution in $\mathcal{S}$ is specified by determining θ. Furthermore, we can also specify a distribution p by its expectation parameter vector $\eta \in \mathbb{R}^{D-1}$, which corresponds to expected values of the distribution and an alternative coordinate system of $\mathcal{S}$. More explicitly, the definition of the expectation parameter η is defined as $\eta_x = \sum_{y \geq x} p(y)$ for $x \in \Omega$, and $\eta_\perp = 1$, where $p(x)$ is the probability mass function of p over the sample set Ω, which is assumed to be a poset. The θ-coordinates and η-coordinates are orthogonal with each other, which means that the Fisher information matrix G has the following property, $G_{u,v} = \partial \eta_u / \partial \theta_v$, and $(G^{-1})_{u,v} = \partial \theta_u / \partial \eta_v$. e- and m-geodesics can also be described using these parameters as follows:

$$\{\theta_t \mid \theta_t = (1-t)\theta^{p_1} + t\theta^{p_2}, 0 \leq t \leq 1\}, \quad \{\eta_t \mid \eta_t = (1-t)\eta^{p_1} + t\eta^{p_2}, 0 \leq t \leq 1\},$$

where θ^p and η^p are θ- and η-coordinate of a distribution $p \in \mathcal{S}$.

A.2 FLATNESS, PROJECTION, AND ITS OPTIMIZATION

A subspace is called *e-flat* when any e-geodesic connecting two points in a subspace is included in the subspace. The vertical descent of an m-geodesic from a point $p \in \mathcal{S}$ onto e-flat subspace $\mathcal{B}_e$ is called m-projection. Similarly, e-projection is obtained when we replace all e with m and m with e. The flatness of subspaces guarantees the uniqueness of the projection destination. The projection destination $\overline{p}$ or $\widetilde{p}$ obtained by m- or e-projection onto $\mathcal{B}_e$ or $\mathcal{B}_m$ minimizes the following KL divergence

$$\overline{p} = \operatorname*{arg\,min}_{q \in \mathcal{B}_e} D_{\mathrm{KL}}(p, q), \quad \widetilde{p} = \operatorname*{arg\,min}_{q \in \mathcal{B}_m} D_{\mathrm{KL}}(q, p),$$

where the KL divergence from discrete distributions $p \in \mathcal{S}$ to $q \in \mathcal{S}$ is given as

$$D_{\mathrm{KL}}(p, q) = \sum_{x \in \Omega} p(x) \log \frac{p(x)}{q(x)}, \tag{3}$$

where $p(x)$ and $q(x)$ are the probability mass functions of p and q, respectively. A subspace with some of its natural parameters fixed at 0 is e-flat (Amari, 2016, Chapter 2), which is obvious from the definition of e-flatness. More generally, any subspace $\mathcal{B}$ resulting from linear constraints on the natural parameter is e-flat. Similarly, any subspace $\mathcal{B}$ resulting from linear constraints on the expectation parameter is m-flat. When a space is e-flat and m-flat at the same time, we say that the space is *dually-flat*. The set of discrete probability distributions $\mathcal{S}$ is dually-flat.

Both e- and m-flatness guarantee that the cost functions to be optimized in Eq. (3) are convex. Therefore, m- and e-projection onto an e- or m-flat subspace can be implemented by a gradient

method using a second-order gradient. This second-order gradient method is known as the *natural gradient method* (Amari, 1998). The Fisher information matrix G appears by second-order differentiation of the KL divergence. For instance, given p and an e-flat subspace $\mathcal{B}_e$, the optimization problem $\overline{p} = \arg\min_{q \in \mathcal{B}_e} D_{\mathrm{KL}}(p, q)$ can be efficiently solved via gradient descent with second-order derivative by the update rule $\theta_{t+1} = \theta_t - G^{-1}(\eta_t - \eta^p)$, where $G \in \mathbb{R}^{D \times D}$ is the Hessian matrix, and $\partial D_{\mathrm{KL}}(P, Q)/\partial\theta = \eta - \eta^p$ is the derivative of the KL divergence. The updated natural parameters θ_{t+1} can then be used to construct $q_{t+1} \in \mathcal{B}_e$ that is closer to the destination $\overline{p}$ along with the e-geodesic from q_t to $\overline{p}$. By repeating this process until convergence, we can always find the global optimal solution. A similar algorithm can be implemented for the other case, i.e., $\widetilde{p} = \arg\min_{q \in \mathcal{B}_m} D_{\mathrm{KL}}(q, p)$.

We make some remarks on the optimization of many-body approximation (Ghalamkari et al., 2024) that we omit in Section 4.4, which is a specific case of the above discussion.

Example A.1 (Many-body approximation)**.** *For $\mathcal{B}_e = \mathcal{M}_\ell$ defined in Eq.*(1)*:*[4]

1. Convexity and uniqueness*: The solution of many-body approximation is always unique, and the objective function of many-body approximation is convex (Ghalamkari et al., 2024, Theorem 1). In particular, the many-body approximation is a maximum likelihood estimation that approximates a non-negative tensor, which is regarded as an empirical distribution, by an extended Boltzmann machine without hidden variables.*
2. Computational complexity*: The computational complexity of the many-body approximation for $\mathcal{B}_e = \mathcal{M}_\ell$ with B many non-fixed indexes (i.e., ℓ-body parameters) is $O(T|B|^3)$, where T is the number of iterations of the optimization. This is because the overall complexity is dominated by the update of θ, which includes matrix inversion of G, and the complexity of computing the inverse of an $n \times n$ matrix is $O(n^3)$.*
   Note that this complexity can be reduced if one reshapes tensors so that the size of each mode becomes small. We explore this idea further in Appendix C.5.

B Omitted details from Section 4

We provide the pseudocode for the proposed algorithm in Algorithms 1 and 2.

Algorithm 1: Backward projection

Data: A representation $w^* \in \mathcal{B}$, φ-embedded dataset $\{z'_i\}_{i=1}^n$ with projection $\{w_i\}_{i=1}^n$ on $\mathcal{B}$, $k \in \mathbb{N}$
Result: Backward projected data z'^*

$N \leftarrow$`Nearest-Neighbor(`k, w^*, $\{w_i\}_{i=1}^n$`)`
$\mathcal{D} \leftarrow$`Construct-Sub-Manifold(`$\{z'_i\}_{i \in N}$`)`
$z'^* \leftarrow$`Proj(`w^*, $\mathcal{D}$`)`
return z'^*

Algorithm 2: Pseudo-non-linear data augmentation

Data: A dataset $\{z_i\}_{i=1}^n$, embedding $\varphi\colon \Omega_{\mathbb{R}} \to \mathcal{S}$, $k \in \mathbb{N}$, flat base sub-manifold $\mathcal{B}$, size $m \in \mathbb{N}$
Result: A generated dataset $\{z_j^*\}_{j=1}^m$ of size m

for $i = 1, \ldots, n$ **do** `// Encoding`
  $z'_i \leftarrow \varphi(z_i)$
  $w_i \leftarrow$`Proj(`z'_i, $\mathcal{B}$`)` `//` $w = \mathsf{Enc}(z) = \mathrm{Proj}_{\mathcal{B}} \circ \varphi(z)$
for $j = 1, \ldots, m$ **do** `// Augmenting`
  $w_i^* \leftarrow$`Augment (`$\{w_i\}_{i=1}^n$, $\mathcal{B}$`)`
for $j = 1, \ldots, m$ **do** `// Decoding`
  $z'^*_j \leftarrow$`Back-Proj(`w_j^*, $\{z'_i\}_{i=1}^n$, $\{w_i\}_{i=1}^n$, k`)` `// Algorithm 1`
  $z_j^* \leftarrow \varphi^{-1}(z'^*_j)$ `//` $z^* = \mathsf{Dec}(w^*) = \varphi^{-1} \circ \mathrm{Proj}_{\mathcal{B}}^{-1}(w^*)$
return $\{z_j^*\}_{j=1}^m$

[4] One can consider $\mathcal{B}_m$ with Eq.(1) being defined w.r.t. the η-coordinate system, and similar remarks hold.

C OMITTED DETAILS FROM SECTION 5

In Appendix C.1, we provide the details of experimental setup omitted in Section 5, and Appendices C.3 to C.5 consists of additional experiments.

C.1 DETAILS OF EXPERIMENTAL SETUP

In this section, we provide the details of each dataset and other experimental setups with relevant explanations.

Datasets. First, we summarize the details of each dataset in Table 2 and relevant parameters for applying the log-linear model on posets and Algorithm 2.

Table 2: Summary of each dataset and the corresponding setups of Algorithm 2.

	Dataset					
	MNIST	CIFAR-10	Speech Commands	Connectionist Bench	Taiwanese Bankruptcy	Wine Quality
Train Size	60,000	60,000	84,848	166	5,455	5,197
Test Size	10,000	10,000	4,890	42	1,364	1,300
Augment Size	10,000	10,000	7,000	32	1,090	1,036
Class	10	10	35	2	2	7
Feature	784	3,072	16,000 $\searrow$ 4,000	60	95	11
Poset Ω	$\mathbb{R}_{\geq 0}^{7^2\times 2^4}$	$\mathbb{R}_{\geq 0}^{2^{10}\times 3}$	$\mathbb{R}_{\geq 0}^{2^5\times 5^3}$	$\mathbb{R}_{\geq 0}^{2^2\times 3\times 5}$	$\mathbb{R}_{\geq 0}^{5\times 19}$	$\mathbb{R}_{\geq 0}^{2^2\times 3}$
Base $\mathcal{B}$ (dim)	$\mathcal{M}_1$ (17)	$\mathcal{M}_5$ (1,410)	$\mathcal{M}_2$ (136)	$\mathcal{M}_1$ (9)	$\mathcal{M}_1$ (23)	$\mathcal{M}_2$ (10)
Local Data $\mathcal{D}$ (dim)	$\mathcal{M}_1^*$ (767)	$\mathcal{M}_4^*$ (2,334)	$\mathcal{M}_3^*$ (3,430)	$\mathcal{M}_2^*$ (30)	$\mathcal{M}_1^*$ (72)	$\mathcal{M}_1^*$ (7)
Bandwidth	0.05	0.05	0.05	0.05	0.05	0.05
Neighbor k	8	3	3	2	5	10

Note that MNIST holds a CC BY-SA 3.0 license, CIFAR-10 is released with a MIT license, and Speech Commands is released with a CC BY 4.0 license. Finally, all the UCI datasets (Connectionist Bench, Taiwanese Bankruptcy, and Wine Quality) are licensed under CC-BY 4.0. We now break each group down and explain it in detail:

1. The first group consists of basic dataset statistics. The first three datasets (MNIST, CIFAR-10, and Speech Commands) come with a default train/test split; for the last three UCI datasets, since there is no default train-test split, we take 80% of the whole dataset as the training set, and the remaining 20% as the test dataset. **Augment Size** reports the size of the augmented data, which is roughly 20% of **Train Size**, off by some rounding errors since we assume we augment the same amount of data for each class.
   In all cases, the full training set is used to train the classifier when evaluating the classification performance in Section 5.3, and also to train our data augmentation baseline (i.e., autoencoder) for comparison. The only exception is that for MNIST and Speech Commands, we choose an equal number of samples for every class when doing our pseudo-non-linear data augmentation for implementation convenience.
   Finally, due to the extremely high dimensionality of Speech Commands (16000), we down-sampled each data to 4000 dimensions in the entire experiment due to the computational constraint: solving 16000-dimension convex programs is infeasible in terms of the memory requirement.
2. The second group consists of the poset structure we impose on each dataset when applying the log-linear model for positive tensors. Since our ultimate goal is to utilize the many-body approximation (Eqs. (1) and (2)), by reshaping the feature vector into a high-order tensor, a finer hierarchy of projection can be obtained. Hence, in all experiments, we reshape the feature w.r.t. the prime-number factorization of the number of features. For instance, an MNIST image is in $\mathbb{R}_{\geq 0}^{28\times 28}$, and we reshape it into a tensor of shape $(7, 2, 2, 7, 2, 2)$, giving it a 6^{th}-order tensor structure $\mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2}$. For notation convenience, in **Poset** Ω, we overload $\mathbb{R}_{\geq 0}^{D}$ to indicate the natural poset structure introduced in Example 4.1, and compress the repeated prime factors in the exponent. We note that for the Wine Quality dataset, since the feature dimension is originally 11, which is a prime, we artificially add 1 dimension by padding 0's, so we get a non-trivial prime factorization.
   Next, **Base** $\mathcal{B}$ and **Local Data** $\mathcal{D}$ report the corresponding construction of the base sub-manifold and the local data sub-manifold using either Eq.(1) or Eq.(2) w.r.t. the given poset structure of that

dataset. The number in the parentheses reports the corresponding dimension of the constructed sub-manifolds.

3. Finally, **Bandwidth** reports the bandwidth parameter we used when fitting the kernel density model M in the generating step, and **Neighbor** k reports the number of nearest neighbors used in Algorithm 1.

Classification Performance. In Section 5.3, we evaluate the augmentation methods by training classifiers and evaluating their prediction accuracy. For MNIST and three UCI datasets, we consider the simple 2-layer MLP, while we use the ResNet-18 for the other two datasets. All models are trained by Stochastic Gradient Descent (SGD) (Ruder, 2016) with learning rate 0.1, momentum 0.9, and weight decay 5×10^{-4}. A step size scheduler is utilized to reduce the learning rate by a factor of 0.1 every 30 epochs until convergence.

For the two baseline data augmentation methods:

- **Standard method (STD)**: The standard image augmentation methods include random horizontal flipping and cropping. For other modalities, the standard augmentation method corresponds to adding Gaussian noise, which is sampled from $\mathcal{N}(0, \sigma^2)$, where σ is chosen to be $1/4$ of the minimum standard deviations over each feature dimension to make sure that the noise level is reasonable.
- **Autoencoder (AE)**: For UCI datasets, we consider a simple two-layer MLP encoder-decoder architecture, where the dimension of the hidden layer is the same as $\dim(\mathcal{B})$ indicated in Table 2, with a bottleneck dimension being 3. For the image datasets, both the encoder and the decoder are based on convolutional layers. The encoder uses a series of convolutional layers with increasing feature map sizes to progressively downsample the input image, while the decoder mirrors this structure with transposed convolutional layers to reconstruct the image from the latent representation.
  In all experiments, the autoencoder is trained with the Adam optimizer with a learning rate 10^{-3} under the mean squared error until convergence.

C.2 Impact of the size of augmentation

In this section, we conduct additional experiments on the impact of the ratio between the augmented dataset size on the performance. Specifically, we consider the two image datasets (MNIST and CIFAR-10) and the audio dataset (Speech Commands), and vary the size of the original size of the augmented dataset, which is 20% of the original dataset size. The results are presented in Tables 3 to 5.

Table 3: Impacts on the sizes of the augmentation dataset for MNIST.

AG	25%	50%	75%	100%
None	98.14%	98.08%	98.10%	98.12%
STD	92.08%	92.00%	92.02%	92.16%
AE	97.91%	97.86%	97.90%	97.97%
PNL	98.01%	98.07%	97.93%	97.55%

Table 4: Impacts on the sizes of the augmentation dataset for CIFAR-10.

AG	25%	50%	75%	100%
None	89.06%	89.32%	88.63%	88.83%
STD	90.42%	92.00%	90.60%	90.92%
AE	88.56%	88.76%	88.42%	86.49%
PNL	88.72%	88.32%	88.53%	88.67%

Table 5: Impacts on the sizes of the augmentation dataset for Speech Commands.

AG	25%	50%	75%	100%
None	83.68%	83.26%	84.72%	84.16%
STD	84.34%	83.84%	84.98%	84.25%
AE	82.30%	83.49%	84.72%	85.21%
PNL	84.20%	84.61%	84.56%	84.43%

C.3 SENSITIVITY AND ROBUSTNESS

We examine our proposed method's robustness and sensitivity of the *bandwidth* used when fitting the kernel density model, and also the *number k of the nearest neighbors* used in Algorithm 1. For an easier visual inspection, we use MNIST in this section.

Bandwidth of Kernel Density Estimation Model. Consider varying the bandwidth we use when fitting the kernel density model, ranging among $\{0.01, 0.05, 0.1, 0.2, 0.5\}$. The results are shown in Figure 5. We observe that Algorithm 2 is robust under different bandwidths when working with the kernel density estimation model in the generating step.

(a) Result with bandwidth 0.01.

(b) Result with bandwidth 0.05.

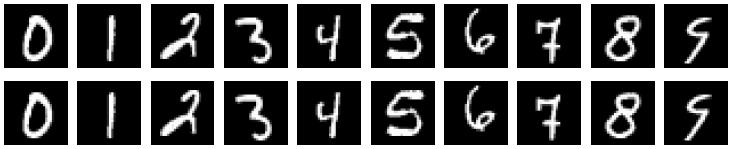
(c) Result with bandwidth 0.1.

(d) Result with bandwidth 0.2.

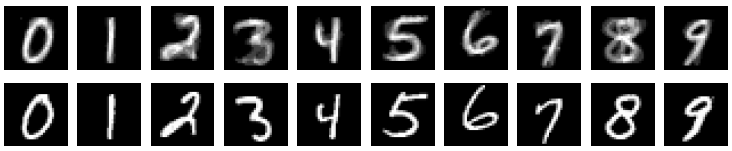
(e) Result with bandwidth 0.5.

Figure 5: Augmented data via Algorithm 2 with different kernel density estimation bandwidths.

Number of Nearest Neighbors. Next, we consider ranging k among $\{1, 4, 8, 16\}$. The results are shown in Figure 6. Observe that when k is small, e.g., 1, the result of Algorithm 2 tends to overfit since the local sub-manifold $\mathcal{D}$ in Algorithm 1 is defined using only the nearest neighbor. When k goes up, a non-trivial augmentation emerges, robust across different k's.

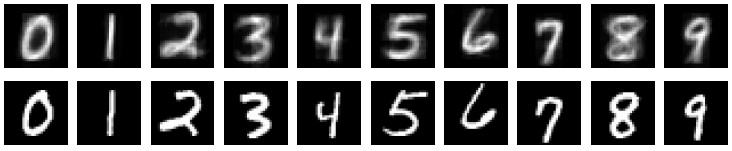


(a) Result with $k = 1$.



(b) Result with $k = 4$.

(c) Result with $k = 8$.



(d) Result with $k = 16$.

Figure 6: (*Top*) Augmented data via Algorithm 2 with different k's for Algorithm 1. (*Bottom*) The closest training data.

C.4 Necessity of dimension reduction

We demonstrate that dimension reduction, a key building block of our proposed method based on the intuition we have from autoencoder-like models, is necessary for Algorithm 2 to work. For an easier visual inspection, we use MNIST in this section.

Direct Fitting. Naive perturbation-based data augmentation methods fall short of high-dimensional data due to the sparsity of the data. Figure 7 shows the results of directly fitting a kernel density estimation model on MNIST with 1000 samples.

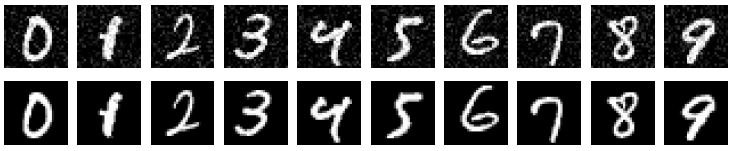
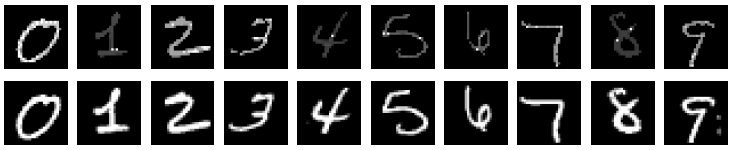


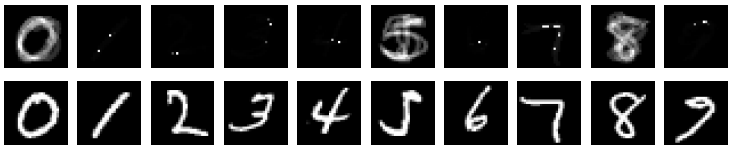
Figure 7: (*Top*) Augmented data via directly fitting a kernel density estimation model with a bandwidth 30. (*Bottom*) The closest training data.

Observe that even with a large bandwidth (30) to introduce variability, we only see a meaningless noisy perturbation on one of the exact training samples, indicating overfitting.

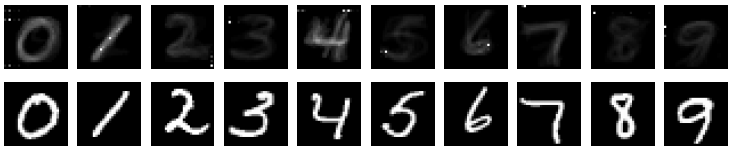
Local Data Sub-manifold. A potential problem related to the necessity of dimension reduction is that, if $\mathcal{D}$ captures too much local information about the data (i.e., with low dimension), backward projecting a random latent representation $w^* \in \mathcal{B}$ might already suffice to augment the data in a non-trivial way, without the need for knowing the latent representations of the training dataset. To this end, consider sampling uniformly random latent representations within the empirical range we observed from the latent representations of the training data and perform Algorithm 1. The results are shown in Figure 8.



(a) Result with $k = 1$.



(b) Result with $k = 4$.



(c) Result with $k = 8$.

(d) Result with $k = 16$.

Figure 8: (*Top*) Augmented data on random latent representations via Algorithm 2 with different k's for Algorithm 1. (*Bottom*) The closest training data.

For $k = 1$, Figure 8(a) shows that, similar to Figure 6(a), it is possible to overfit one of the training data (i.e., the nearest neighbor of the randomly sampled latent representation). This is not surprising since the base sub-manifold is only of dimension 17, as the random latent representation is sufficiently close to one of the representations of the training data in $\mathcal{B}$, their backward projection result should not deviate too much. Furthermore, we observe the *fading effect*, which intuitively corresponds to *misspecification of the energy*, indicating that the sampled latent representation is fundamentally different from that of the dataset.

As k increases, the benefit of getting informative and meaningful latent representations from the original dataset becomes clear. Specifically, we start to see *degeneration*: from unclear overlappings to collapsing (i.e., only a few pixels are showing). Intuitively speaking, it is because the random latent representation's nearest neighbors appear to be significantly different, hence failing to provide a consistent local data sub-manifold. For instance, in the extreme case when $k = 16$, the local data sub-manifold is completely not informative, resulting in collapsing. Overall, without dimension reduction, we will lose the reference of *realistic latent representations* provided by the original dataset, which leads to bad performance once we are beyond the trivial overfitting regime.

C.5 Choices of tensor structure and construction of sub-manifolds

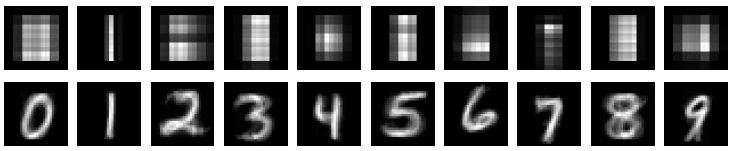
In Section 5.2, we consider varying ℓ for $\mathcal{B} = \mathcal{M}_\ell$ with the tensor structure $\mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2}$ on the MNIST dataset. In this section, we further vary the tensor structure as well: in particular, we consider the tensor structure of the MNIST image being $\mathbb{R}_{\geq 0}^{28\times 28}$, $\mathbb{R}_{\geq 0}^{7\times 4\times 7\times 4}$, and $\mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2}$.

Remark C.1. *For notation convenience, we also write their corresponding poset structures as $\mathbb{R}_{\geq 0}^{28\times 28}$, $\mathbb{R}_{\geq 0}^{7\times 4\times 7\times 4}$, and $\mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2}$, and further write the many-body approximation sub-manifold (Eq.(1)) as $\mathcal{M}_\ell(\Omega)$ and its dual (Eq.(2)) as $\mathcal{M}_\ell^*(N, \Omega)$ for a particular poset Ω to emphasize the dependency.*

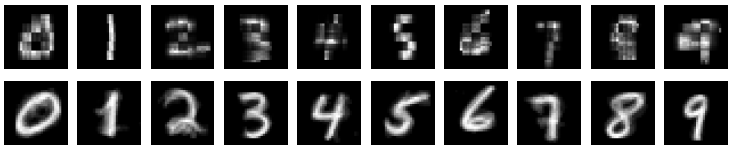
Finally, we consider ranging ℓ from 1 to at most 4 where we neglect the degenerate case: for instance, in the case of $\mathbb{R}_{\geq 0}^{28\times 28}$, $\mathcal{M}_2(\mathbb{R}_{\geq 0}^{28\times 28}) = \mathcal{S}$ as there are only two modes for a matrix, therefore degenerates to direct fitting which is not of interest (see Appendix C.4). Note that throughout this section, we fix the default local data sub-manifold to be $\mathcal{D} = \mathcal{M}_1^*(N, \mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2})$ for consistency.

The results for the finest structure, $\mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2}$, are shown in Figure 9. As ℓ grows, the forward projection results (*Top*) preserve the structure of the data better, subsequently improving the quality of the augmented data (*Bottom*). Similar trends can be found in the case of $\mathbb{R}_{\geq 0}^{7\times 4\times 7\times 4}$, as shown in Figure 10.

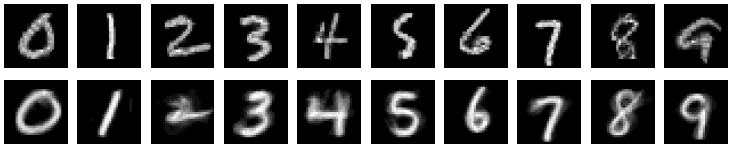
If we look at the results when using the original matrix structure $\mathbb{R}_{\geq 0}^{28\times 28}$ (Figure 11), some interesting comparisons can be made. Firstly, if we compare the augmentation results for $\mathcal{B} = \mathcal{M}_1(\mathbb{R}_{\geq 0}^{28\times 28})$ (Figure 11 (*Bottom*)) with the finer structures counterparts, e.g., (Figure 9(a) (*Bottom*)) for $\mathcal{M}_1(\mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2})$, one can observe that the results are worse. However, the former requires more dimension ($\dim(\mathcal{M}_1(\mathbb{R}_{\geq 0}^{28\times 28})) = 55 > 17 = \dim(\mathcal{M}_1(\mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2}))$) for the base sub-manifold. Similarly, the augmentation results with $\mathcal{B} = \mathcal{M}_1(\mathbb{R}_{\geq 0}^{7\times 4\times 7\times 4})$ (Figure 10(a) (*Bottom*)) also achieve better performance with a lower base sub-manifold dimension.



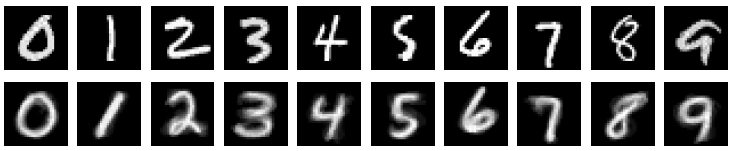
(a) Result with base sub-manifold $\mathcal{B} = \mathcal{M}_1(\mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2})$ ($\dim(\mathcal{B}) = 17$).



(b) Result with base sub-manifold $\mathcal{B} = \mathcal{M}_2(\mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2})$ ($\dim(\mathcal{B}) = 107$).

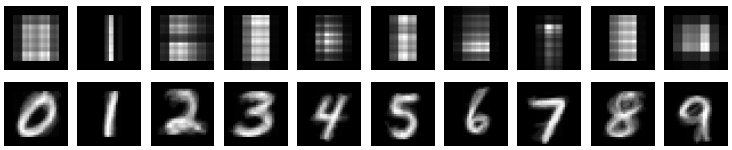


(c) Result with base sub-manifold $\mathcal{B} = \mathcal{M}_3(\mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2})$ ($\dim(\mathcal{B}) = 327$).

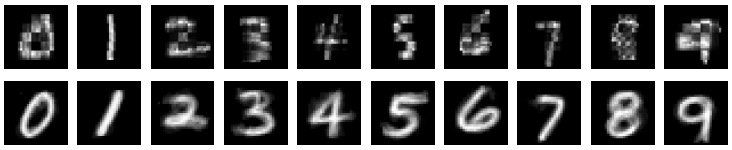


(d) Result with base sub-manifold $\mathcal{B} = \mathcal{M}_4(\mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2})$ ($\dim(\mathcal{B}) = 592$).

Figure 9: (*Top*) Forward projection on $\mathcal{B} = \mathcal{M}_\ell(\mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2})$. (*Bottom*) Backward projection on $\mathcal{D} = \mathcal{M}_1^*(\mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2})$.



(a) Result with base sub-manifold $\mathcal{B} = \mathcal{M}_1(\mathbb{R}_{\geq 0}^{7\times 4\times 7\times 4})$ ($\dim(\mathcal{B}) = 19$).



(b) Result with base sub-manifold $\mathcal{B} = \mathcal{M}_2(\mathbb{R}_{\geq 0}^{7\times 4\times 7\times 4})$ ($\dim(\mathcal{B}) = 136$).

Figure 10: (*Top*) Forward projection on $\mathcal{B} = \mathcal{M}_\ell(\mathbb{R}_{\geq 0}^{7\times 4\times 7\times 4})$. (*Bottom*) Backward projection on $\mathcal{D} = \mathcal{M}_1^*(\mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2})$.

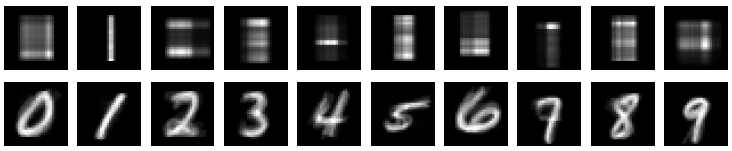


Figure 11: (*Top*) Forward projection on $\mathcal{B} = \mathcal{M}_\ell(\mathbb{R}_{\geq 0}^{28\times 28})$. (*Bottom*) Backward projection on $\mathcal{D} = \mathcal{M}_1^*(\mathbb{R}_{\geq 0}^{7\times 2\times 2\times 7\times 2\times 2})$.