Learning Manifold Dimensions with Conditional Variational Autoencoders

Anonymous Author(s)
Affiliation
Address
email

Abstract

Although the variational autoencoder (VAE) and its conditional extension (CVAE) are capable of state-of-the-art results across multiple domains, their precise behavior is still not fully understood, particularly in the context of data (like images) that lie on or near a low-dimensional manifold. For example, while prior work has suggested that the globally optimal VAE solution can learn the correct manifold dimension, a necessary (but not sufficient) condition for producing samples from the true data distribution, this has never been rigorously proven. Moreover, it remains unclear how such considerations would change when various types of conditioning variables are introduced, or when the data support is extended to a union of manifolds (e.g., as is likely the case for MNIST digits and related). In this work, we address these points by first proving that VAE global minima are indeed capable of recovering the correct manifold dimension. We then extend this result to more general CVAEs, demonstrating practical scenarios whereby the conditioning variables allow the model to adaptively learn manifolds of varying dimension across samples. Our analyses are also supported by numerical results on both synthetic and real-world datasets.

1 Introduction

VAEs [6, 14] and CVAEs [17] are poweful generative models as they have achieved competitive results in various domains, for example, image synthesis [21, 5, 13], natural language processing [16], time-series forecasting [9, 19], trajectory prediction [8], etc. Recently, it has been shown that it also achieves state-of-the-art performance for large-scale images given the right architecture [11].

Besides the success, VAEs and CVAEs still have observed behaviors that need precise understanding or theoretical analysis. In particular, when the data lie on or near a low-dimensional manifold, it is meaningful to have a model that learns the dimension of manifold correctly, since it is a necessary but not sufficient condition for the model to produce samples from the true data distribution. In general, the manifold assumption holds for data in real-world [12], and recovering the dimension can help both the generation quality as well as the understanding of data itself. Although it is suggested in prior work [4, 3] that a VAE model can learn the correct manifold dimension when globally optimized, this has only been proven under an assumption that the decoder is linear or approximately linear [2, 4].

Moreover, the ability to learn the correct manifold dimension is less clear when a conditioning variable is introduced. As a set of discrete conditions may correspond to different "slices" of data and each could lie on a manifold with different dimension. For example, in MNIST one may expect the manifold dimension of images labelled "1" is smaller than those of "5". It is also possible to have data expand fully in the ambient space but lie on a low-dimensional manifold when conditioned. Such an example can be on purposely constructed by directly selecting some data dimensions and take corresponding manifold dimensions as the conditioning variable.

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

In addition, the above theoretical properties of VAEs and CVAEs not only help our understanding on the precise behavior on paper, but also directly relevant to the analysis on some common model design choices and assumptions in practice. For example, there has been debate about the choice between a fixed data variance and a learnable one [4, 10, 15, 18, 3]. Another commonly adapted strategy is to share weights for prior and encoder in CVAEs [7, 17]. It may not be obvious, but these design choices are relevant to the ability of learning the manifold dimensions.

Our paper raises and investigates the following four unresolved issues for VAEs and CVAEs:

(i) Besides the extensive observation of VAE being able to learn the correct data manifold dimension, there is no rigorous proof of it yet.

(ii) It remains unclear how to extend the above observation of VAE to conditional VAE, i.e. how are the manifolds related with conditioning variables.

(iii) There has no discussion on the situation where data lies on a union of manifolds.

(iv) The active practices on VAE and CVAE models emerges various model assumptions and design choices, it remains to be discussed that how these assumptions and choices impact the learning of correct manifold dimensions.

In Section 2, we first formally prove that a VAE model can learn the correct data manifold dimension, then extend it to the conditional VAE. In Section 3, some popular model design choices are discussed and analyzed. Section 4 supports our theoretical conclusions with numerical experiments on both synthetic and real-world datasets.

2 Learning the Dimension of Data Manifold

We begin with the definition of data:

Definition 1 (Data lying on a Manifold) *Suppose r and d are two positive integers and $r < d$, $\mathcal{X}$ is a simple r-Riemannian manifold embedded in $\mathbb{R}^d$ when there exists a diffeomorphism φ between $\mathcal{X}$ and $\mathbb{R}^r$. Specifically, for every $x \in \mathcal{X}$, there exists $u = \varphi(x) \in \mathbb{R}^r$, where φ^{-1} exists and both φ and φ^{-1} are invertible and differentiable.*

Given infinite observable variables $x \in \mathcal{X}$, where $\mathcal{X}$ is a r-dimensional manifold embedded in $\mathbb{R}^d$ whose definition is in 1, and latent variable $z \in \mathcal{Z}^\kappa \subseteq \mathbb{R}^\kappa$ which serves as a low-dimensional representation, a VAE model is built to approximate the ground truth measure with a parametric density $p_\theta(x) = \int p_\theta(x|z)p(z)dz$, where prior $p(z) = N(0, I)$. Denote the ground-truth measure on $\mathcal{X}$ as ω_{gt}. The probability mass of an infinitesimal dx on the manifold is $\omega_{gt}(dx)$ and $\int_{\mathcal{X}} \omega_{gt}(dx) = 1$. The canonical cost of VAE model is a bound on the average negative log-likelihood

$$\mathcal{L}(\theta, \phi) = \int_{\mathcal{X}} \{-\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] + \mathbb{KL}[q_\phi(z|x)||p(z)]\}\omega_{gt}(dx) \tag{1}$$

Before starting our main theorems, we first convert the similar idea from [4] and formalize a κ-simple VAE model with explicit Gaussian and Lipschitz assumptions, as well as the way of parameterizations.

Definition 2 (κ-simple VAE) *A κ-simple VAE is defined as a VAE model with dim[z] = κ latent dimensions, the Gaussian encoder $q_\phi(z|x) = N(z|\mu_z(x;\phi), \mathrm{diag}\{\sigma_z^2(x;\phi)\})$, the Gaussian decoder $p_\theta(x|z) = N(x|\mu_x(z;\theta), \gamma I)$, and the prior $p(z) = N(z|0, I)$. Here $\gamma > 0$ is a trainable scalar in θ. The mean functions of the encoder and the decoder, i.e. $\mu_z(x;\phi)$ and $\mu_x(z;\theta)$ are arbitrary L-Lipschitz continuous functions.*

2.1 Learning Manifold Dimension in VAE

We start by analyzing the dimension of manifold learning in VAE models referred to issue (i). As it is pointed out in [4], if γ of a VAE model is trainable and not constrained, γ must be arbitrarily small but explicitly nonzero to minimize the cost and reconstruct data from $\mathcal{X}$. Here we define *active latent dimensions* as the dimensions that have a decaying encoder variance related to γ.

Definition 3 (Active Latent Dimensions in VAE) *In a κ-simple VAE, let $\{\theta^*_\gamma, \phi^*_\gamma\}$ denote any global optimal κ-simple VAE solutions as a function of any fixed γ, then for any $j = 1, \ldots, \kappa$, a dimension j of latent variable z is defined as an active dimension if the optimal encoder variance $\sigma_z(x; \phi^*_\gamma)^2_j \to 0$ as $\gamma \to 0$.*

Theorem 1 (Learning the Data Manifold Dimension Using VAEs) *Suppose training data lies on a r-dimensional manifold embedded in $\mathbb{R}^d$. Then for any $\kappa \geq r$, when a κ-simple VAE model achieves its any global optimums, we have*

(i) $\mathcal{L}(\theta^*_\gamma, \phi^*_\gamma) = (d - r)\log\gamma + O(1)$, and

(ii) The number of active latent dimensions almost surely equals r, and

(iii) The optimal variance of each active dimension satisfies $\sigma_z(x; \phi^*_\gamma)^2_j = O(\gamma)$.

A rigorous proof of Theorem 1 is provided in the appendices. As a summary, our proof follows this path: first, we prove the existence of active dimensions and the rate by contradiction. The conclusion is that there must exist at least r active dimensions whose encoder variances going to zero at a rate of $O(\gamma)$ for reconstruction tasks, otherwise the reconstruction term will grow at a rate of $O(\frac{1}{\gamma})$ as γ goes to zero. Next, we get an upper bound and a lower bound of ELBO and show that their difference is a constant, both in the form of $(d - r)\log\gamma + O(1)$. Last, we get the exact number of active dimensions by showing that adding unnecessary active dimensions decreases the coefficient of $\log\gamma$, which may increase loss value.

This result address issue (i) in Section 1. It implies that we can explicitly obtain the manifold dimension and the state of convergence by viewing the performance of σ_z^2 and γ. Also, both in theory and experiments, the ratio of the norm term in reconstruction and γ is a constant, i.e. the data dimensions d. Specifically, when a VAE model converges to its optimum, γ will go to zero, as well as r dimensions of σ_z^2. To achieve the optimal solution, the active latent dimensions play a crucial role in reconstruction and the inactive ones help to push KL regularization factor to its minimum along these dimensions. By definition, inactive dimensions are with a variance such that $\lim_{\gamma\to 0} \sigma_z(x; \phi^*_\gamma) = C > 0$, where C is some constant. When such latent dimensions are greater than $d - r$, no matter how many inactive dimensions there are, the model cannot perfectly reconstruct even a single manifold dimension.

2.2 Learning Manifold Dimension in Conditional VAE

In this section, we extend the analysis on κ-simple VAE models to κ-simple CVAE models. We first the manifold with condition. Specifically, suppose $\mathcal{X}$ is a Riemannian manifold and there exists a set $\mathcal{C}$ where for each $x \in \mathcal{X}$, there is a corresponding $c \in \mathcal{C}$, and we call that c is the condition of x.

Definition 4 (κ-simple CVAE) *For any $x \in \mathcal{X}$ and its corresponding $c \in \mathcal{C}$, a κ-simple CVAE is an extension of κ-simple VAE with the Gaussian encoder $q_\phi(z|x, c) = N(z|\mu_z(x, c; \phi), \mathrm{diag}\{\sigma_z^2(x, c; \phi)\})$, the Gaussian decoder $p_\theta(x|z, c) = N(x|\mu_x(z, c; \theta), \gamma I)$, and the prior $p_\theta(z|c) = N(z|\mu_z(c; \theta), \mathrm{diag}\{\sigma_z^2(c; \theta)\})$. $\mu_z(x, c; \phi)$ and $\mu_x(z, c; \theta)$ are arbitrarily L-Lipschitz continuous, while $\mu_z(c; \theta)$ can be arbitrary function.*

The canonical CVAE cost with conditional variable being c is

$$\mathcal{L}_c(\theta, \phi) = \int_{\mathcal{X}} \{-\mathbb{E}_{q_\phi(z|x,c)}[\log p_\theta(x|z, c)] + \mathbb{KL}[q_\phi(z|x, c)||p(z|c)]\}\omega_{gt}(dx) \tag{2}$$

When prior is independent of c, i.e. $p(z|c) = p(z)$, the cost still holds.

In a CVAE model, besides samples $x \in \mathcal{X}$, the conditioning variable c may also provide information relevant to x. Thus we define the *effective dimension* to measure the contribution of the conditioning variable.

Definition 5 (Effective Dimension of the Conditioning Variable) *For any $x \in \mathcal{X}$ and its corresponding $c \in \mathcal{C}$, we call integer t the number of effective dimensions of c, if 1) there exists a function $g : \mathcal{C} \to \mathbb{R}^t$ and exists t dimensions of $\varphi(x)$, denoted as $\varphi(x)_t$, such that $g(c) = \varphi(x)_t$, where φ is*

the diffeomorphism in Definition 1, and 2) there doesn't exist such a function g for $t+1$. Further, let $\mathcal{C}_t \subset \mathcal{C}$ be the subset for all c with effective dimensions of t.

Definition 5 shows that any $c \in \mathcal{C}_t$ can reconstruct at most $t \leq r$ dimensions of the manifold.

Also, we extend our definition of *active latent dimensions* to CVAE.

Definition 6 (Active Latent Dimensions in CVAE) *In a κ-simple CVAE, denote $\mathrm{diag}\{\sigma_{z_p}(c;\theta)\}$ and $\mathrm{diag}\{\sigma_{z_q}(x,c;\phi)\}$ as the variance of the prior and encoder. For any $j = 1, \ldots, \kappa$, a dimension j of latent variable z is defined as an active dimension if the ratio $\sigma_{z_q}(x,c;\phi^*_\gamma)^2_j / \sigma_{z_p}(c;\theta^*_\gamma)^2_j \to 0$ as $\gamma \to 0$.*

As special cases, for VAE or CVAE with a standard Gaussian prior we have $\sigma_{z_p}(x)^2_j = 1$, and that indicates Definition 3 is a special case of Definition 6.

Next, we extend Theorem 1 to CVAE models, which describes how the manifold dimension relates to conditioning variables.

Theorem 2 (Learning Manifold Dimension in CVAE) *Suppose $\mathcal{X}$ is a r-dimensional manifold embedded in $\mathbb{R}^d$ with its corresponding condition $c \in \mathcal{C}_t$. Then for any $\kappa \geq r$, when a κ-simple CVAE model achieves its global optimum, we have*

(i) $\mathcal{L}_c(\theta^*_\gamma, \phi^*_\gamma) = (d - r + t)\log\gamma + O(1)$, and

(ii) The number of active latent dimensions almost surely equals $r - t$, and

(iii) The variance of each active dimension satisfies $\sigma_{z_q}(x,c;\phi^*_\gamma)^2_j / \sigma_{z_p}(c;\theta^*_\gamma)^2_j = O(\gamma)$.

This theorem addresses issue (ii) in Section 1. It shows the significance of conditioning variable that it can push the cost to a smaller value than a VAE model without condition, which is favorable. It also points out that the condition will reduce the number of active latent dimensions without disrupting the data fit. It worth mentioning that even though the data is not on the manifold initially, once a conditioning variable with a positive effective dimension is incorporated, the conditioned data then become on a manifold, which can be induced directly in our theorem.

Moreover, to address issue (iii) in Section 1, we have the following results describing CVAE's capacity in learning data lies on a union of manifolds.

Corollary 2.1 (Adaptive Active Dimension) *Suppose there exist $x_1, x_2 \in \mathcal{X}$ with r-dimensional manifold and their corresponding conditioning variables $c_1 \in \mathcal{C}_{t_1}$ and $c_2 \in \mathcal{C}_{t_2}$, i.e. c_1, c_2 have t_1 and t_2 effective dimensions respectively. Given a CVAE model with an optimal solution $\{\theta^*, \phi^*\}$, we have the number of active dimensions of $z_1 \sim q_{\phi^*}(z|x,c_1)$ equals to $r - t_1$, and that of $z_2 \sim q_{\phi^*}(z|x,c_2)$ equals to $r - t_2$.*

This result indicates that CVAE can adaptively learn different data manifolds under conditioning variables with different effective dimensions. When c is discrete, it is equivalent to training separate VAEs where each model corresponds to a discrete condition value. On the other hand, collecting all possible conditions in a single CVAE model enables it to compress the dimension dynamically. For example, c describes the set of manifolds that share the same dimension, thus for samples in each set, the corresponding manifold dimensions are the same. When c is continuous, conditioning variables may contain intact information of certain manifold dimensions, which substitute for corresponding active dimensions. In Section 4, we demonstrate numerical evidence of this behavior.

3 On Common Model Design Choices

In this section, we review some model designs and training practices that are commonly seen in prior works but could potentially negatively impact the model convergence and performance. The following discussions can be seen as important steps towards resolving issue (iv) in Section 1.

3.1 On the Equivalence of Conditioned and Unconditioned Prior

In prior works, it is popular to design a parameterized prior $p_\theta(z|c)$[1] in CVAE [24, 7, 8, 20, 1], which is a trainable module alongside the encoder $q_\phi(z|x,c)$ and the decoder $p_\theta(x|z,c)$.

Remark 1 (Remove conditioning variable from the Prior) *Consider a κ-simple CVAE model with prior $p_\theta(z|c)$, encoder $q_\phi(z|x,c)$ and decoder $p_\theta(x|z,c)$. We can always find another κ-simple CVAE model with prior $p(z) \sim \mathcal{N}(0,I)$, encoder $q_{\phi'}(z|x,c)$ and decoder $p_{\theta'}(x|z,c)$, such that $\mathcal{L}_c(\theta,\phi) = \mathcal{L}_c(\theta',\phi')$.*

Remark 1 indicates that a conditioned prior is not necessary. Specifically, as shown in its proof, we can always explicitly convert an existing κ-simple CVAE model with prior $p_\theta(z|c)$ into another κ-simple CVAE model with prior being $\mathcal{N}(0,I)$. In practice, when we have the flexibility in architecture design, it is equivalent to use a standard Gaussian prior or a parameterized one.

3.2 The Initial γ Impacts Model Convergence and Dimension Learning

Recall in Theorem 1, we have $\mathcal{L}(\theta^*_\gamma, \phi^*_\gamma) = (d-r)\log\gamma + O(1)$, and that implies when $\mathcal{L}(\theta^*_\gamma, \phi^*_\gamma)$ approaches to $-\infty$, γ is expected to be closer to 0. This behavior makes γ important in CVAE training. A direct conclusion is that the model would be compromised if γ is set to be a fixed positive constant, as analysed and empirically studied in literature [4, 10, 15, 18, 3].

Although a model converging towards global optimum would push γ to 0 regardless of its initial value, in practice, we observe that this initial value of γ still significantly impacts the model performance. In the cost function, we have

$$-\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] = \frac{1}{\gamma}\int_{\mathcal{Z}} q_\phi(z|x)||x-\mu_x(z)||_2^2 dz + \log(2\pi\gamma)$$

By this formulation we see that γ acts as the weight between data likelihood and KL-divergence. A large γ downgrades the weight for data likelihood, and encourages the model to optimize the KL-divergence term first, and vice versa. Making γ a learnable parameter is equivalent to adaptively balancing the cost. When the initial value is set to be on an extreme end, the model is at risk of falling into a local minimum, as empirically in Section 4.4.

3.3 Weight Sharing Keeps Model from Global Optimal Convergence

The cost of VAE balances the three modules the encoder, decoder and prior with the two terms, the data likelihood and KL-divergence. It has been argued in literature [3] that such a cost might lead to unstable training dynamics. Therefore, one widely adapted trick[17] in practice is to share model weights between prior and posterior.

Weight sharing is conducted in various forms. For data from a general domain where x and c are differently distributed, the weight sharing couples with the design in which the posterior takes in x and c as two input and process differently at the beginning, and the prior processes c as its input with the same module and weight in the posterior.

In this section we take the sequential data as a representation for the weight sharing issue because it takes a more thorough approach of weight sharing. Sequential data poses the task of using early parts of a sequence to predict the later parts. Formulating in CVAE, we have c being the early parts, and that conditions the to-predict later parts x.

Assuming x is a sequential data, i.e. $\{x_l\}$ with $l > 0$ denoted as the index, and each x_l is paired with its condition $c = x_{<l}$. As a result, the encoder becomes $q_\phi(z_l|x_{\leq l})$, and the prior becomes $p(z_l|x_{<l})$ for any l greater than the observed sequence length. It has been proposed in literature [7, 8] to share weights between the encoder and prior, i.e. $p(z_l|x_{<l}) = q_\phi(z_l|x_{<l})$.

Theorem 3 (Weight Sharing Compromises Performance of Sequential Modeling) *By sharing weights between the encoder and prior, the conditional VAE model has its cost being*

$$\mathcal{L}_c(\theta,\phi) = \int_{\mathcal{X}} \sum_l \{-\mathbb{E}_{q_\phi(z|x_{\leq l})}[\log p_\theta(x_l|z,x_{<l})] + \mathbb{KL}[q_\phi(z|x_{\leq l})||q_\phi(z|x_{<l})]\}\omega_{gt}(\mathrm{d}x) \quad (3)$$

[1]Note that we slightly abused the notation of θ as parameters for both the prior and decoder.

Then we have $\mathcal{L}_c(\theta, \phi) = \Omega(1)$ for any θ and ϕ.

Theorem 3 implies that a fully weight-sharing pair of prior and encoder prevents the model from achieving its global optimum. As the numerical study in Section 4.5, this weight sharing strategy could severely constraint the performance.

4 Experiments

In this section we present numerical results to support our analysis. We first verify model behavior in the synthetic environment for a controllable set of experiments, and extend to real-world datasets to further validate our conclusions.

4.1 Dataset and Metrics

Synthetic Dataset The basic data samples are generated from a mixture of a finite number of Gaussian distributions. Specifically, we first generate a 5-dimensional categorical selecting vector from uniform distribution, and normalize it with norm equaling one. Then we generate two r-dimensional parameter vectors μ_u and $\log \sigma_u$ from a standard Gaussian distribution in $\mathbb{R}^r$. By repeating the second step 5 times, we get 5 independent Gaussian distributions. Our synthetic data $u \in \mathbb{R}^r$ is generated by selecting one of the 5 candidate distributions with a probability of the selection vector, and next generate samples from the selected distribution. To get the manifold $\mathcal{X}$ in $\mathbb{R}^d$, we design a function $h : \mathbb{R}^r \to \mathbb{R}^d$, and the data sample $x = h(u)$. In our experiments, $h(u) = \text{Sigmoid}(G_x u)$, where $G_x \in \mathbb{R}^{d \times r}$ is randomly initialized controlled by a certain random seed. The conditioning variable c is transformed by another transform $h'(u_{1:t}) = G' u_{1:t}$, where $G' \in \mathbb{R}^{t \times t}$. Given a fixed seed, the generation results is controlled by parameters r, d and t. For all the experiments, the training size of synthetic data is 100,000.

Real-world dataset Besides synthetic dataset, we also investigate the model behavior on MNIST[23] and Fashion MNIST [22], two image datasets[2]. These two datasets contain x with complex mappings between the manifold and ambient space. At the same time, they are not too complex thus the training are made reasonably easy with common neural architecture, otherwise our focus could be driven to specific architecture choices other than VAE and CVAE model behavior. By carefully tuning the pair (x, c), we can control how c impacts the model.

Metrics The main metric that we use to evaluate VAEs and CVAEs is the negative Evidence Lower BOund (ELBO), as in Eq 1 and Eq 2, and note that it is reported on convention and not for comparison. There are also auxiliary metrics that we use to compare and diagnose models, including Active Dimensions (AD), Reconstruction error $\mathbb{E}||x - \mu_x(z)||^2$ (Recon), the KL-divergence ($\mathbb{KL}$), and the data variance γ.

4.2 Manifold Dimensions Learning in VAE

Here, we provide direct numerical support to Theorem 1. Specifically, we show that the model captures the correct dimensions of manifold regardless of the ambient data dimensions. To indicate how the active dimension is determined, we provide an example in Section B of the appendix. Although in numerical experiments a variance cannot be exact zero, we can still observe from a converged model that each variance of encoder is either close to 1, or close to zero.

With $d = \kappa = 20$ and varying r, we demonstrate that the VAE model can learn the active latent dimensions when γ converges to a small value, regardless of the value of r. The results are showed in Table 1 and all the numbers are values at convergence.

Experimental results on MNIST and Fashion MNIST are reported in Table 2. Without knowing the true manifold dimensions, a small latent dimension discourages the model from catching up the information for every dimension, while once given enough latent dimensions, the number of active dimensions equals to data's intrinsic dimension and the rest is for the regularization. The two parts of latent dimensions cooperate to push the VAE model to its global optimum.

[2]MNIST is under the CC BY-SA 3.0 license, and Fashion MNIST is MIT-Licensed.

Table 1: VAE latent compression in synthetic dataset. The first three blocks of rows varied d and r. VAE shows powerful capacity to learn the manifold dimension. With a surplus in latent dimension, i.e. $\kappa > r$, the number of active dimensions is exactly r and a small negative ELBO is achievable. As showed in the last row, given $\kappa < r$, model capacity is constrained since the active dimensions are not enough for reconstruction, which is indicated in the proof of Theorem 1.

κ	d	r	AD	Recon	$\mathbb{KL}$	γ	-ELBO
20	10	2	2	3×10^{-4}	18.31	1.625×10^{-5}	-58.26
		4	4	2.6×10^{-3}	24.22	5.654×10^{-5}	-29.83
		6	6	9.2×10^{-3}	24.14	3×10^{-4}	-17.39
		8	7	1.27×10^{-2}	27.91	1.4×10^{-3}	-10.38
		10	8	5.99×10^{-2}	16.39	2.5×10^{-3}	-6.40
	20	2	2	1.6×10^{-3}	17.98	5.052×10^{-5}	-114.52
		4	4	1.75×10^{-2}	23.11	2×10^{-4}	-60.90
		6	6	3.09×10^{-2}	28.96	6×10^{-4}	-43.75
		8	8	3.42×10^{-2}	33.83	1.2×10^{-3}	-36.82
		10	10	4.74×10^{-2}	35.81	1.1×10^{-3}	-28.34
	30	2	2	2.6×10^{-3}	18.42	7.221×10^{-5}	-176.74
		4	4	2.73×10^{-2}	24.60	2×10^{-4}	-100.28
		6	6	4.74×10^{-2}	31.89	9×10^{-4}	-76.46
		8	8	5.68×10^{-2}	37.28	1.6×10^{-3}	-65.66
		10	10	1.13×10^{-1}	35.13	2.5×10^{-3}	-47.00
5	20	6	5	1.299×10^{-1}	22.53	2.1×10^{-3}	-36.97
		8	5	3.719×10^{-1}	16.618	8.8×10^{-3}	-22.60
		10	5	3.564×10^{-1}	15.966	1.113×10^{-2}	-16.96

Table 2: VAE latent compression in real dataset. Support of Theorem 1.

Dataset	κ	AD	Recon	-ELBO
MNIST	5	5	14.899	-842.286
	16	12	9.749	-1065.83
	32	13	7.469	-1224.37
Fashion MNIST	5	5	13.163	-935.127
	16	9	9.026	-1216.68
	32	9	7.820	-1327.26

4.3 CVAE Adaptive Active Dimensions Varied With c

In this section, we design experiments to show that a CVAE model can learn the dimension of manifold with respect to the effective latent dimensions of the condition. Further, it can also learn the adaptive active dimension on a union of manifolds.

By Theorem 2 and Remark 1, a CVAE model with an unconditioned prior and condition $c \in \mathcal{C}_t$ is equivalent to a general one in design, whose active dimensions equals $r - t$. In this experiments, we investigate the behavior of a CVAE model when data lies on a single manifold and its relation to condition is static. Specifically, we set c to be the first t dimensions of u, i.e. take G' as I_t, and let $d = \kappa = 20$, $r = 10$. Table 3 shows the results.

On MNIST and Fashion MNIST, we train CVAEs with the class label of each image as the conditioning variable, and the results with $\kappa = 32$ are in Table 4. By comparing it to Table 2, we notice that the CVAE model for MNIST has a lower active dimension and better ELBO, while for FashionMNIST the CVAE performs similarly. This implies that when data points in MNIST are conditioning on the label, they lie on a lower-dimensional manifold, while for data points in FashionMNIST where among classes the manifold dimensions are similar, the class label doesn't reduce the manifold dimension. This characterization of the dataset complexity between classes is consistent with their intuitive visual

Table 3: CVAE latent compression showing AD $= r - t$ exactly, which supports our Theorem 2. c is the first t dimensions of u, i.e. $G' = I_t$, $d = \kappa = 20$, $r = 10$.

t	-ELBO	Recon	$\mathbb{KL}$	γ	AD
1	-31.41	4.61×10^{-2}	33.26	2.4×10^{-3}	9
3	-36.67	4.66×10^{-2}	27.78	2.4×10^{-3}	7
5	-42.78	4.86×10^{-2}	20.81	2.6×10^{-3}	5
7	-52.39	4.29×10^{-2}	13.72	2.2×10^{-3}	3
9	-62.25	3.84×10^{-2}	6.07	2×10^{-3}	1

Table 4: CVAE latent compression in real datasets. $\kappa = 32$. AD is averaged over all the classes. This result supports Theorem 2.

	AD	Recon	$\mathbb{KL}$	γ	-ELBO
MNIST	12	6.044	81.672	0.0063	-1489.42
Fashion MNIST	9	8.773	54.552	0.0102	-1239.09

complexity. We also provide an example of the encoder variance on the MNIST dataset in Section B of the appendix.

Discrete Condition To verify Corollary 2.1, we train a CVAE model on d-dimensional data that lie on a union of manifolds with different dimensions, with the conditioning variable c being an indicator of the source manifold. Table 5 shows that the model successfully captures the active latent dimensions with an attention layer in the decoder, while without the attention layer the model learns a static active dimension of 4. The result indicates two aspects of CVAE: 1) CVAE models have capacity to capture a union of manifolds simultaneously and converge on each manifold (indicated by -ELBO); 2) In practice, we need careful model architecture design to help the model to distinguish manifolds. In this experiment $d = 20$ and $\kappa = 40$.

Continuous Condition When c is a continuous variable, we show that the model can also adaptively learn the active dimensions. In our setting, a continuous condition $c \in \mathcal{C}_t$ contains t-dimensional information of samples, thus under different conditions the data may lie on different manifolds. We set $r = 12$, $d = 20$ and $\kappa = 90$, then we set the first $2, 4, 6, 8, 10$ dimensions of u as the conditions. Since the model needs dimension of c to be consistent, we padding them to $\mathbb{R}^{10}$ with 0. With the attention layer, the model can use the condition to collapse certain dimensions instead of activating all the possible dimensions and increasing unnecessary loss in KL term. The results are included in Table 6.

4.4 Initialization of γ Impacts Model Training

Here we check how the initialization of γ would impact model convergence, as mentioned in Section 3.2. Our results are in Table 7. With different initial $\log\gamma$ values, we observe the convergence of VAE is affected significantly, as a result, the model can't capture the correct manifold dimension. In comparison, CVAE models are less sensitive, as CVAEs have more parameters and the conditioned

Table 5: Adaptively learnt active dimension with discrete c. $d = 20, \kappa = 40$.

r	True AD	AD without attention	AD with attention	-ELBO without attention	-ELBO with attention
1	1	4	1	-102.25	-114.22
2	2	4	2	-62.42	-99.81
3	3	4	3	-60.13	-74.28
4	4	4	4	-28.06	-50.36
5	5	4	5	-7.58	-59.25

Table 6: Adaptively learnt active dimension with continuous c. $r = 12, d = 20, \kappa = 90$.

t	True AD	AD without attention	AD with attention	-ELBO without attention	-ELBO with attention
2	10	10	10	-9.69	-31.62
4	8	10	8	-33.71	-37.31
6	6	10	6	-27.10	-42.76
8	4	10	4	-50.70	-68.67
10	2	10	2	-61.77	-61.92

Table 7: Active dimensions under different initial γ. $d = 20, r = 10, t = 5, \kappa = 20$.

Init $\log\gamma$	VAE		CVAE $p(z)$		CVAE $p(z\|c)$	
	AD	-ELBO	AD	-ELBO	AD	-ELBO
-20	10	-28.39	5	-41.78	5	-40.72
-10	9	-28.57	5	-41.97	5	-45.25
0	8	-27.56	5	-42.69	5	-45.2
10	3	-13.89	5	-40.827	5	-43.66
20	1	-1.7	3	-29.71	4	-37.85

data lie on a lower-dimension manifold thus it actually poses an easier task. Still, when $\log\gamma = 20$, they converge to local optima. In this experiment, $d = 20, \kappa = 20, r = 10$. In CVAE, $t = 5$.

4.5 Shared Weights Between Encoder and Prior

The experiments in this section is to support Theorem 3, i.e. sharing weights between the encoder and prior is not a satisfactory way for training in sequential data, we generate a sequence by AutoRegressive-Moving-Average (ARMA) model with fixed parameters, then use rolling windows to make our samples. Specifically, the window size is 5, i.e. each sample we use the past 5 points as our condition, and reconstruct the sixth one which is viewed as current sample. The results in Table 8 show that when weight shared, the norm term of reconstruction cannot go down, which may result in a restriction in the capacity of the model, and this observation is coherent to the key contradiction used in our proof of Theorem 3.

5 Conclusion

In this paper we provide insights on the behavior of VAEs and CVAEs for data lying on low-dimensional manifolds. In the theoretic analysis, we show that VAE has the ability to learn the manifold dimension. More interestingly, we extend the conclusion to Conditional VAE that works with conditioned data. This additional conditioning term impacts the relation between data and manifold, as a result it also affects the model design choices in practice. With the theoretic results, we further examine some design choices. A dedicated section of experiments supports our statements and conclusions numerically on both synthetic and real-world data. Finally, we believe this work can in general help to improve the design and diagnose techniques of VAEs and CVAEs.

Limitations Our work may still have limitations in empirical verification for real-world datasets at a larger scale or from other modalities than images. Although our contributions are theoretically grounded, in practice it still takes non-trivial efforts to design model architectures that both align with our guideline and fit the data.

Table 8: Weight sharing on sequential data between the encoder and prior.

Weight Shared	-ELBO	Recon	$\mathbb{KL}$	γ
True	-2.49	0.374	18.09	0.012
False	-45.015	1.81×10^{-5}	175.99	7.252×10^{-7}

References

[1] Matthias Bauer and Andriy Mnih. Resampled priors for variational autoencoders. In *International Conference on Artificial Intelligence and Statistics*, 2019.

[2] Bin Dai, Yu Wang, John Aston, Gang Hua, and David Wipf. Hidden talents of the variational autoencoder. *arXiv preprint arXiv:1706.05148*, 2018.

[3] Bin Dai, Li Wenliang, and David Wipf. On the value of infinite gradients in variational autoencoder models. *Advances in Neural Information Processing Systems*, 34, 2021.

[4] Bin Dai and David Wipf. Diagnosing and enhancing VAE models. In *International Conference on Learning Representations*, 2019.

[5] Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taïga, Francesco Visin, David Vázquez, and Aaron C. Courville. Pixelvae: A latent variable model for natural images. In *5th International Conference on Learning Representations*, 2017.

[6] Diederik Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

[7] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pages 2688–2697. PMLR, 2018.

[8] Longyuan Li, Jian Yao, Li Wenliang, Tong He, Tianjun Xiao, Junchi Yan, David Wipf, and Zheng Zhang. Grin: Generative relation and intention network for multi-agent trajectory prediction. *Advances in Neural Information Processing Systems*, 34, 2021.

[9] Sindy Löwe, David Madras, Richard Zemel, and Max Welling. Amortized causal discovery: Learning to infer causal graphs from time-series data. In *Proceedings of the First Conference on Causal Learning and Reasoning*. PMLR, 2022.

[10] Pierre-Alexandre Mattei and Jes Frellsen. Leveraging the exact likelihood of deep latent variable models. *Advances in Neural Information Processing Systems*, 31, 2018.

[11] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchical vq-vae. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10775–10784, 2021.

[12] Phil Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. In *International Conference on Learning Representations*, 2021.

[13] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[14] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

[15] Danilo Jimenez Rezende and Fabio Viola. Taming VAEs. *arXiv preprint arXiv:1810.00597*, 2018.

[16] Iulian Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[17] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[18] Hiroshi Takahashi, Tomoharu Iwata, Yuki Yamanaka, Masanori Yamada, and Satoshi Yagi. Student-t variational autoencoder for robust density estimation. In *International Joint Conference on Artificial Intelligence*, 2018.

[19] Binh Tang and David S Matteson. Probabilistic transformer for time series analysis. *Advances in Neural Information Processing Systems*, 34:23592–23608, 2021.

[20] Jakub Tomczak and Max Welling. VAE with a VampPrior. In *International Conference on Artificial Intelligence and Statistics*, 2018.

[21] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[22] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[23] Corinna Cortes Yann LeCun. THE MNIST DATABASE of handwritten digits. `http://yann.lecun.com/exdb/mnist/`.

[24] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *Advances in neural information processing systems*, 30, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 4 and Appendix.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] See Section 4
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]