

# Variational Autoencoded Regression: High Dimensional Regression of Visual Data on Complex Manifold

YoungJoon Yoo<sup>1</sup> Sangdoon Yun<sup>2</sup> Hyung Jin Chang<sup>3</sup> Yiannis Demiris<sup>3</sup> Jin Young Choi<sup>2</sup>

<sup>1</sup>Graduate School of Convergence Science and Technology, Seoul National University, South Korea

<sup>2</sup>ASRI, Dept. of Electrical and Computer Eng., Seoul National University, South Korea

<sup>3</sup>Personal Robotics Laboratory, Department of Electrical and Electronic Engineering

Imperial College London, United Kingdom

<sup>1</sup>yjyoo3312@gmail.com <sup>2</sup>{yunsd101, jychoi}@snu.ac.kr <sup>3</sup>{hj.chang, y.demiris}@imperial.ac.uk

## Abstract

This paper proposes a new high dimensional regression method by merging Gaussian process regression into a variational autoencoder framework. In contrast to other regression methods, the proposed method focuses on the case where output responses are on a complex high dimensional manifold, such as images. Our contributions are summarized as follows: (i) A new regression method estimating high dimensional image responses, which is not handled by existing regression algorithms, is proposed. (ii) The proposed regression method introduces a strategy to learn the latent space as well as the encoder and decoder so that the result of the regressed response in the latent space coincide with the corresponding response in the data space. (iii) The proposed regression is embedded into a generative model, and the whole procedure is developed by the variational autoencoder framework. We demonstrate the robustness and effectiveness of our method through a number of experiments on various visual data regression problems.

## 1. Introduction

Regression of paired input and output data with an unknown relationship is one of the most crucial challenges in data analysis. In diverse research fields, such as trajectory analysis, robotics, the stock market, etc. [15, 44, 44, 28, 22], target phenomena are interpreted as a form of paired input / output data. In these applications, a regression algorithm is usually used to estimate the unknown response for a given input by using the information obtained from the observed data pairs. Many vision applications can also be expressed as such input / output data pairs. For example, in Fig. 1 (a), the sequence of the motion images can be described by input / output paired data, where the input can be defined as the relative order and the output response is defined as the

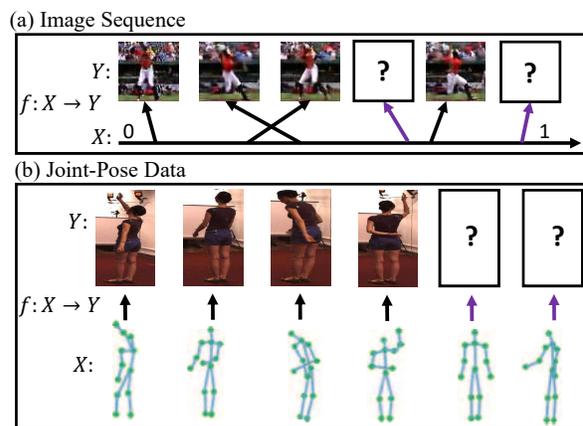


Figure 1. Examples of paired data in vision applications. (a) For the image sequence, the domain can be defined by the space representing relative orders of image sequences. (b) For joint-pose data pairs, the joint vector space can be a possible domain.

corresponding image. The motion capture data and their corresponding images in Fig. 1 (b) are another example. The input data are 3D joint positions, and their responses will be the corresponding posture images. If we can model the implicit function representing the given image data pairs via regression, we can estimate unobserved images that correspond to the input data.

However, applying existing multiple output regression algorithms [1, 3, 2, 39] to these kinds of visual data applications is not straightforward, because the visual data are usually represented in high dimensional spaces. In general, high dimensional visual data (such as image sequences) are difficult to be analyzed with classical probabilistic methods because of their limited modeling capacity [17, 18]. Thus, regression of visual data to estimate visual responses requires a novel approach.

In handling high dimensional complex data, recent at-

tempts [12, 27, 37, 24, 36] using a deep generative network, such as the variational autoencoder (VAE) [24], have achieved significant success in reconstructing images. In VAE, a latent variable is defined to embed the compressed information of the data, and an encoder is trained to map a data space into its corresponding latent space. A decoder is also trained to reconstruct images from a sampled point in the latent space. The projection for the input data into the latent space (via the encoder) captures the essential characteristics of the data and allows execution of the regression task using a much lower dimensional space. The regression in the latent space is done together with the training of the encoder and decoder. However, a naive combination of regression and the VAE is not particularly effective because the decoder and the latent space are not designed in a way that permits the result of the regressed response in latent space, and the corresponding response in data space, to coincide. Therefore, a new method to simultaneously train the latent space and the encoder/decoder is required to achieve coincidence between the regressed latent vector and the reconstructed image.

In this paper, we solve this problem by combining the VAE [24] and Gaussian process regression [35]. The key idea of this work is to do regression in the latent space instead of the high-dimensional image space. The proposed algorithm generates the latent space that compresses the information of both the domain and output image using the VAE, and projects the data pairs to the latent space. Then, regression is conducted for the projected data pairs in latent space, and the decoder is trained so that the regression in latent space and image space coincide. To the best of our knowledge, it is the first attempt to apply the VAE framework to solve the regression problem. The whole process, including the loss function, is designed as the generative model, and a new mini-batch composition method is presented for training the decoder to satisfy our purpose.

All connection parameters of the encoder / decoder are inferred by the end-to-end stochastic gradient descent method as described in [23]. The proposed regression method is validated with two different examples: sports sequences and motion image sequences with skeletons. The first example presents a regression case of simple domain to complex codomain, and the second example presents the complex domain to complex codomain case.

## 2. Related Work

**Deep Generative Network:** Classical probabilistic generative models [34, 8, 42, 19, 29, 40, 5] have proven to be successful in understanding diverse unsupervised data, but their descriptive ability is insufficient to fully explain complex data such as images [18]. Recently, as in other works in the vision area [16, 31], deep layered architectures have been successfully applied to solve this problem

with powerful data generation performance. Among these architectures, generative adversarial network (GAN) [12] and generative moment matching networks (GMMN) [27] directly learn the generator that maps latent space to data space. Meanwhile, the variants of the restricted Boltzmann machine (RBM) [17, 18, 38, 37] and probabilistic autoencoders [24, 30, 11] learn the encoder that defines the map from data to latent space and the generator (decoder) simultaneously. The former methods, and especially variants of GAN [12, 9, 33], are reported to describe the edges of generated images more sharply than the latter methods. However, the applicability of these methods is restricted due to the difficulty of discovering the relationships between data and latent space. This innate nature makes it difficult to use adversarial networks for designing the regression. Therefore, this paper adopts the variational autoencoder framework [24], which is also more suitable than RBM families to expand the regression model.

**Variational Autoencoder:** Since Kingma *et al.* [24] first published the variational autoencoder (VAE), numerous applications [32, 43] have been presented to solve various problems. Yan *et al.* [43] proposed conditional VAE in order to generate the image conditioned on the attribute given in the form of sentences. Furthermore, recent work [14, 25, 13] has demonstrated that a sequence in latent space would be mapped back to the sequence of data. Hence these methods embedded dynamic models such as recurrent neural networks [14] and the Kalman filter [25, 21] into the VAE framework. These algorithms [14, 25, 25, 13] successfully show the ability of dynamic models in a latent space to capture the temporal changes of relatively simple objects in images. In this paper, we apply the VAE for the regression task in a relatively complex manifold.

**Regression:** Regression of paired data is theoretically well established and analytic solutions for the infinite dimension of the basis function [4, 35] have been derived for the last century. In non-parametric cases, Gaussian process [35, 26] provides a general solution by expanding Bayesian linear regression using kernel metrics and a Gaussian process prior. By using this method, we can estimate the output data as a Gaussian posterior composed of given data pairs and input data to the unobserved target outputs. However, applying the algorithms for the high-dimensional output data is difficult because the kernel metric has limited capacity to express the complicated high dimensional data. The variants of multiple output regression algorithms [2, 3, 1, 39] are proposed to deal with multi-dimensional output responses. Still, these algorithms focus on handling relatively low dimensional output responses and are not able to sufficiently describe complicated data, such as that of an image. In this paper, we construct a regressed latent space by using a variational autoencoder to handle complex data.

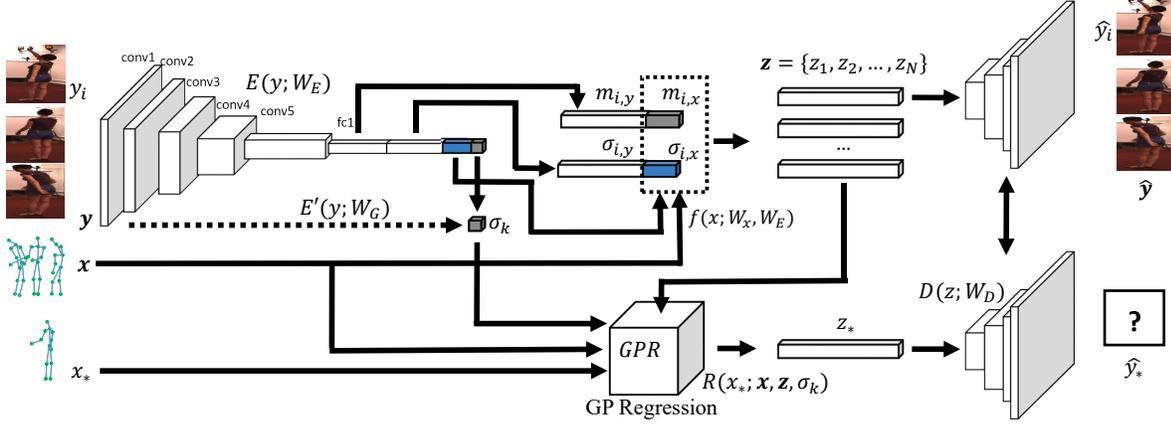


Figure 2. Overall scheme of the proposed method. For observed data pairs  $\mathbf{x} = \{x_1, x_2, \dots, x_N\}$  and  $\mathbf{y} = \{y_1, y_2, \dots, y_N\}$ , the proposed autoencoder reconstructs  $\hat{y}_i \simeq y_i$  as shown in the top right. For the unobserved  $y_*$  to given  $x_*$ , it is impossible to obtain  $z_*$  by using an encoder with  $W_E$ , because we do not have information about  $y_*$ . Thus to estimate  $y_*$ , we obtain  $z_*$  using regression from  $\mathbf{x}, \mathbf{z}$  and  $x_*$ , and estimate the response  $\hat{y}_*$  from  $z_*$ .

### 3. Proposed Method

#### 3.1. Overall Scheme

Given the target data pair  $(x_i, y_i), i = 1, \dots, N, x_i \in \mathcal{X}, y_i \in \mathcal{Y}$ , our goal is to find the unknown response  $y_*$  for a new input  $x_*$ . In this paper, the response  $y \in \mathcal{Y}$  is defined as an image and the corresponding input  $x \in \mathcal{X}$  is defined accordingly based on the applications as in Fig. 1.

As shown in Fig. 2, for the observed data pairs  $(x_i, y_i), i = 1, \dots, N$ , the encoder/decoder produces  $\hat{y}_i$  which is the reconstruction of an observed image  $y_i$ . For the observed data, the encoding network  $E(\cdot)$  produces mean and variance for a part of the latent vector  $z_i$ , that is,  $[m_{i,y}, \sigma_{i,y}] = E(y_i; W_E)$  which compresses  $y_i$  to a latent variable with Gaussian mean  $m_{i,y}$  and variance  $\sigma_{i,y}$ . The remaining part of  $z_i$  is modeled by  $[m_{i,x}, \sigma_{i,x}] = f(x_i, W_x)$  which represents mean and variance of  $z_i$ . Thus, the Gaussian distribution of  $z_i$  is described by  $m_i = [m_{i,y}, m_{i,x}]$  and  $\sigma_i = \text{diag}[\sigma_{i,y}, \sigma_{i,x}]$ . For the unobserved image  $y_*$  for a newly given  $x_*$ , the proposed method produces  $\hat{y}_*$ , which is an estimator of  $y_*$ .

Using  $z_i$  sampled from  $\mathcal{N}(m_i, \sigma_i)$ , the decoding network reconstructs the output response  $\hat{y}_i$ , that is,  $\hat{y}_i = D(z_i; W_D)$ . Note that if  $(W_E, W_x, W_D)$  is well trained by the training scheme in Section 3.4,  $\hat{y}_i$  should be similar to  $y_i$ . However, for an unobserved  $y_*$  to a given  $x_*$ , it is impossible to obtain  $z_*$  from  $E(\cdot; W_E)$  because we do not have any information on  $y_*$ . To estimate  $y_*$ , we obtain  $z_*$  by using regression from  $\mathbf{x} = \{x_1, x_2, \dots, x_N\}, \mathbf{z} = \{z_1, z_2, \dots, z_N\}$  and  $x_*$ . For this regression,  $z_i$  is sampled from  $\mathcal{N}(m_i, \sigma_i)$  for each observed response  $y_i \in \mathbf{y} = \{y_1, y_2, \dots, y_N\}$ . Then, we estimate  $z_*$  using Gaussian process (GP) regression  $z_* \sim R(x_*; \mathbf{x}, \mathbf{z}, \sigma_k)$  to be described in Section 3.2, where  $\sigma_k$  is a kernel parameter of the GP regression, which can be produced by an additional

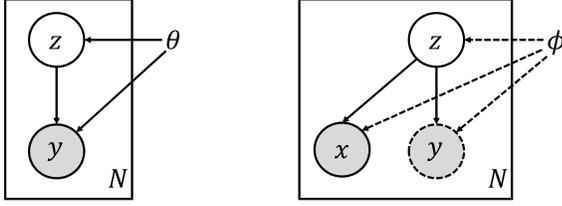
encoder  $\sigma_k = E'(y, W_G)$ ; with  $\sigma_k = [\sigma_{k,1}, \dots, \sigma_{k,N}]$ . In this paper, for computational simplicity, we combine this kernel encoder with the encoder network  $E(y; W_E)$  and change the outputs into  $[m_{i,y}, \sigma_{i,y}, \sigma_{i,k}] = E(y_i, W_E)$ .

After  $z_*$  is estimated, the response  $\hat{y}_*$  is reconstructed from  $z_*$  by using the decoding network  $D(z_*; W_D)$ . Note that the  $D(z_*; W_D)$  should reconstruct not only  $\hat{y}$  from the  $z_i$  sampled by  $\mathcal{N}(m_i, \sigma_i)$ , but also  $y_*$  from the  $z_*$  which is the regression result obtained from  $x_*, \mathbf{x}$ , and  $\mathbf{y}$ . The whole procedure is designed as a generative framework with joint distribution  $p(x_*, y_*, \mathbf{x}, \mathbf{y}, W_E, W_x, W_D)$ , and hence can be derived by the VAE algorithm.

#### 3.2. Variational Autoencoded Regression

The proposed scheme (depicted in Fig. 2) is derived from the directed graph model in Fig. 3. The diagram in Fig. 3 (a) represents the generative model describing a typical reconstruction problem, and the diagram in Fig. 3 (b) is the variational model which not only approximates the generative model in Fig. 3 (a), but also performs the regression for the estimation of unobserved  $y$  by utilizing an information variable  $x$  related to  $y$ . The joint distribution  $p_\theta(y, z)$  can be expressed by the likelihood function  $p_\theta(y|z)$  and the prior distribution  $p_\theta(z)$ , where  $\theta$  refers to the set of all parameters related to the generation of the response  $y \in \mathcal{Y}$  from the latent variable  $z$ . In our method, the prior distribution of  $z$  is defined as zero mean Gaussian distribution, as in typical variants of VAE [43, 24]. Also, the likelihood function  $p_\theta(y|z)$  depicts the decoding process in the proposed scheme. Below, it is shown that  $\theta$  is realized by the parameter  $W_D$  of the decoding network.

Once the joint distribution  $p_\theta(y, z)$  is defined, the posterior  $p_\theta(z|y)$  can be theoretically derived from the Bayes theorem, but the calculation is intractable. Therefore, the variational distribution  $q_\phi(z|x, y)$  is introduced to approxi-



(a) Generative model  $p_\theta(y|z)p_\theta(z)$  (b) Variational model  $q_\phi(z|x,y)$ . Figure 3. The directed graphical model of the proposed method. (a) Generative model for  $y$  and the latent variable  $z$ . (b) Variational distribution to approximate the posterior  $p_\theta(z|y)$  of the generative model.  $y$  is not observed for the newly given input  $x = x_*$ .

mate the true posterior distribution  $p_\theta(z|y)$ . Unlike  $p_\theta(z|y)$ ,  $x$  is introduced for the variational distribution  $q_\phi(z|x,y)$  to sample  $z_* \sim R(x_*, \mathbf{x}, \mathbf{z}, \sigma_k)$ , which is the result of the GP regression for the unknown  $y_*$ .  $q_\phi(z|x,y)$  represents the overall encoding procedure generating the latent variable  $z$  from the input data pair  $(x,y)$  and correspondingly, the variational parameter  $\phi$  is realized by the parameters  $W_E, W_x$  as described in Section 3.1. Importantly,  $q_\phi(z|x,y)$  should be able to explain both cases: 1) an observed image  $y_i$ , and 2) an unobserved image  $y_*$  which requires regression as mentioned previously. For the first case, the variational distribution is defined as  $z_i \sim q_\phi(z|x = x_i, y = y_i)$ . For the latter case, the variational distribution is defined as  $z_* \sim q_\phi(z|x = x_*, y \in \emptyset)$  which represents the GP regression procedure for estimating latent  $z_*$  for the input  $x_*$ .

In order to estimate the parameters  $\theta$  and  $\phi$  which minimize the distance between  $p_\theta(z|y)$  and  $q_\phi(z|x,y)$ , we minimize the Kullback–Leibler divergence  $D_{KL}(p_\theta(z|y)||q_\phi(z|x,y))$ . Following the derivation in [6, 24], the minimization procedure  $\{\theta^*, \phi^*\} = \arg \min_{\{\theta, \phi\}} D_{KL}(p_\theta(z|y)||q_\phi(z|x,y))$  is converted to  $\{\theta^*, \phi^*\} = \arg \min_{\{\theta, \phi\}} L(\theta, \phi)$ , where

$$L(\theta, \phi) = -D_{KL}(q_\phi(z|x,y)||p_\theta(z)) + \sum_{i=1}^N \log p_\theta(y_i|z_i) + \sum_{j=1}^M \log p_\theta(\hat{y}_{*,j}|z_{*,j}). \quad (1)$$

$z_{*,j}$  and  $y_{*,j}$  represents the  $M$  number of latent codes and output responses for  $x_{*,j}, j = 1, \dots, M$ , to be regressed. In (1),  $\hat{y}_{*,j} = D(z_{*,j}; W_D)$  is the reconstructed response from  $z_{*,j}$  by the decoding network as depicted in Fig. 2. The parameters  $\theta$  and  $\phi$  are realized by the connection parameters of the encoding network with regression for  $q_\phi(z|x,y)$ , and the decoding network for  $p_\theta(z|y)$  (see Section 3.3). To minimize the loss in (1), we propose a method for mini-batch learning (see Section 3.4). The Adam optimizer [23] is used for stochastic gradient descent training.

### 3.3. Model Description

For the encoding part, we define  $q_\phi(z|x,y)$  which maps the data pair  $(x,y)$  into the latent space  $\mathcal{Z}$ . For

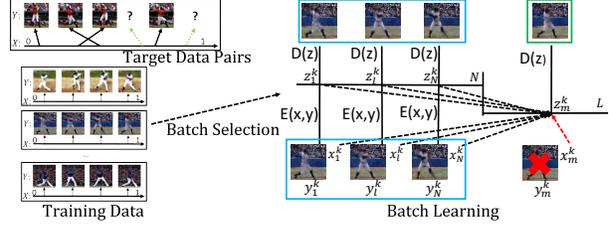


Figure 4. Training strategy of the proposed method. The mini-batch is generated from the sampled training data sequences.

both, observed and unobserved images,  $q_\phi(z|x,y)$  is defined by Gaussian distribution as in (2) and it enables us to analytically solve the KL-divergence term  $D_{KL}(q_\phi(z|x,y)||p_\theta(z))$  in (1) following [24]:

$$q_\phi(z|x,y) = \mathcal{N}(z|m(x,y), \sigma(x,y)). \quad (2)$$

The variational parameter  $\phi$  consists of the Gaussian mean function  $m(x,y)$  and the variance function  $\sigma(x,y)$ . The  $m(x,y)$  and  $\sigma(x,y)$  are produced in different ways depending on the input data. When the input data is given by  $x = x_i \in \mathbf{x}$ , the encoder yields  $m(x,y) = [m_{i,y}, m_{i,x}]$  and  $\sigma(x,y) = \text{diag}[\sigma_{i,y}, \sigma_{i,x}]$ , where  $\text{diag}[\cdot]$  refers to a diagonal matrix. When the input data is given by  $x = x_{*,j} \in \mathbf{x}_*$ ,  $m(x,y)$  and  $\sigma(x,y)$  are determined by the mean and variance  $(m_{*,j}, \sigma_{*,j})$  estimated by GP regression from  $\mathbf{z}, \mathbf{x}$  and  $x_{*,j}$ , where

$$m_{*,j} = K_{*,j}K^{-1}\mathbf{Z}, \sigma_G = (K_{**,j} - K_{*,j}K^{-1}K_{*,j}^T)I. \quad (3)$$

$\mathbf{Z}$  refers to the matrix  $[z_1; z_2; \dots; z_N] \in \mathcal{R}^{N \times D}$ , and  $I \in \mathcal{R}^{D \times D}$  is the identity matrix, where  $D$  is the dimension of  $z \in \mathcal{Z}$ . The matrices  $K, K_{**,j}$  and  $K_{*,j}$  are defined as

$$K = \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k(x_N, x_1) & \dots & k(x_N, x_N) \end{bmatrix}, \quad (4)$$

$$K_{**,j} = k(x_{*,j}, x_{*,j}), \quad (5)$$

$$K_{*,j} = [k(x_{*,j}, x_1), k(x_{*,j}, x_2), \dots, k(x_{*,j}, x_N)]. \quad (6)$$

For the kernel  $k(\cdot, \cdot)$ , we use a simplified version of SE-kernel [35], where  $k(x_i, x_j) = \sqrt{\sigma_i \sigma_j} \exp \{-\|x_i - x_j\|^2\}$ . Eventually, the variational parameter  $\phi$  is realized by the weight matrices  $(W_x, W_E)$  of the encoder network. In summary, for the given data  $\mathbf{x}, \mathbf{y}$ , and  $\mathbf{x}_*$ ,  $q_\phi(z|x,y)$  in (2) is given as

$$q_\phi(z|x,y) = \begin{cases} \mathcal{N}(m_i, \sigma_i) & x = x_i, y = y_i. \\ \mathcal{N}(m_{*,j}, \sigma_{*,j}) & x = x_{*,j}, y \in \emptyset. \end{cases} \quad (7)$$

For the decoding procedure, we define the likelihood function  $p_\theta(y|z) = p(y|D(z; W_D))$ , where  $p(y|D(z; W_D))$

is defined as a Gaussian distribution with mean  $D(z; W_D)$  and fixed variance. Since the prior of  $z$  is defined with zero mean Gaussian and identity covariance matrix, the weight  $W_D$  represents the generative model parameter  $\theta$ . Correspondingly, the meanings of the second term and third term in (1) are interpreted as follows. Since the negative log-likelihood  $(-\log(p_\theta(y|z)))$  is defined as  $l_2$  distance  $\|y - D(z; W_D)\|^2$  in our algorithm, the second term represents the reconstruction error for the given data pair  $(x_i, y_i)$  to  $y_i$ , and the third term denotes the estimation error for  $y_{*,j}$  via regression from the given input data  $x_{*,j}$  and the observed data  $(x_i, y_i), i = 1, \dots, N$ .

### 3.4. Training

To train the parameters of the proposed model, a sufficient number of the training datasets is required. In our algorithm, a total of  $V$  different training sequences  $(x_i^v, y_i^v), v = 1, \dots, V, i = 1 \dots, N_v$  are used, as shown in Fig. 4. These training data pairs share similar semantics to the target (test) data pair  $(x_i, y_i)$ . If the target data pair is a golf swing sequence, the training data pairs will be different golf swing sequences obtained in different situations. Once the parameters are trained by the training dataset consisting of diverse golf swings, the proposed method can complete the target image sequence via regression from the incomplete test sequence on a golf swing. After training the model with the mini-batch, we fine-tune the parameters with observed data pairs in target regression.

**Mini-Batch Training:** The work in [7] reports that the composition of the mini-batch is critical when using variants of stochastic gradient descent methods [23, 10] to train the parameters. To generate the batch, in this paper,  $K$  sequences of a total  $V$  sequences are randomly selected. For each selected training sequence  $k = 1 \dots K$ , we randomly pick  $L$  data pairs  $(x_l^k, y_l^k), l = 1, \dots, L$ , where  $L = (M + N)$ . For the earlier  $N$  data pairs  $(x_n^k, y_n^k), n = 1, \dots, N$ , we get the latent space vector  $z_n^k$  from the encoder function  $E(y_n^k; W_E)$ , and  $f(x_n^k; W_x)$  to train  $W_E, W_x$ , and  $W_D$ . Alternatively, for the latter  $M$  data pairs  $(x_m^k, y_m^k), m = (N + 1), \dots, L$ , we obtain the latent  $z_m^k$  by regression (Section 3.3) from  $\{z_1^k, \dots, z_N^k\}, \{x_1^k, \dots, x_N^k\}$  and  $x_m^k$ . The responses  $\{y_{N+1}^k, \dots, y_L^k\}$  are assumed to be unknown in the encoding process. This data set is used to train the decoder network  $D(z; W_D)$  to reconstruct the proper responses not only for the  $z_n^k$  from the data pair  $(x_n^k, y_n^k)$ , but also for the  $z_m^k$  which are obtained from the regression. The corresponding loss from the estimated  $\hat{y}_m^k$  and the actual  $y_m^k$  refers to the the third term in (1). We note that it is possible to calculate the loss term because  $y_m^k$  can be used as ground truth regression response. After constructing the batch, the stochastic gradient [23] for the batch is calculated to train all parameters.

**Parameter Fine-Tuning:** After training the parameters

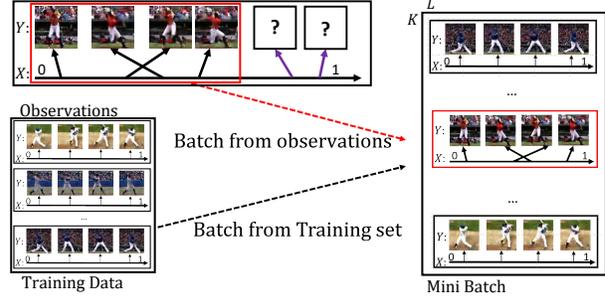


Figure 5. Batch generation for fine-tuning. The batch is composed of observed data pairs (red) and sampled data pairs in training dataset.

$W_E, W_x$  and  $W_D$  using the batch from the training dataset, we further fine-tune the parameters with the observed data pairs  $(x_i, y_i), i = 1, \dots, N$  in the same way as previous regression techniques [35, 26]. Note that the training of the regression part is not done because the ground truth is not available for the test dataset. For the fine-tuning process, mini-batches are composed of the observed test data pairs  $(x_i, y_i)$  and randomly selected  $(K - 1)$  data sequences  $(x_i^k, y_i^k)$  from the training set as in Fig. 5, where  $i = 1, \dots, N_k$  and  $k = 1, \dots, (K - 1)$ . When the total number  $N$  of observed test data pairs is less than  $L$ , we increase the number of samples by allowing repetition. Then, the parameters are fine-tuned with 50 iterations. The detailed implementation is described in the supplementary material.

## 4. Experiments

In the experiments, we evaluated the regression capability of the proposed method via two applications composed of image data: (1) a problem with a simple temporal domain and complicated codomain and (2) a problem with a complicated domain and codomain. For the first application, we used sport data sequences obtained from YouTube. Human pose reconstruction for a given skeleton was tested for the second application.

### 4.1. Sports Data Sequences

**Evaluation Scenario:** In this scenario, we created data sets for three sport sequences: baseball swing, golf swing, and weightlifting. The dataset includes 236 baseball swings, 232 golf swings and 129 weightlifting sequences from YouTube. In the dataset, 1000 – 2000 images are included for each action sequence, and their relative orders are given. The domain is defined as  $\mathcal{X} : [0, 1]$  and a point in  $\mathcal{X}$  is assigned to  $x$  for each image  $y \in \mathcal{Y}$  according to its relative order in the entire sequence. For testing, the golf and the baseball swings were trained with 200 randomly selected sequences and tested with those that remained. The weight lifting scenario was trained with 100 sequences. We executed the regression for each test sequence with 20 observed



Figure 6. Qualitative Results on regression from the sport dataset (best viewed in color). The row (a) in each sport represents the proposed regression results. The images in rows (b) result from the regression with R-VAE. Row (c) is the result from MOGP [2]. The results on the right indicate the samples of reconstruction results for observed images.



Figure 7. experimental results comparing with the NN method. (a) proposed method. (b) NN with the latent space from VAE.

images within all images of each sequence and compared the results with multiple-output GP regression (MOGP) [2], and GP regression combined with vanilla VAE [24] (called R-VAE from here on). For R-VAE, we conducted the fine-tuning process in the same way as the proposed method. For MOGP, we trained the kernel with two-thirds of the images in the given sequences.

**Qualitative Analysis:** Fig. 6 shows the qualitative comparison of image generation results. The sequences in Fig. 6 show samples uniformly picked among the regressed responses from 100 evenly divided points in the range  $[0, 1]$ . As seen in (a), the proposed method generated the most accurate responses compared to the other methods. R-VAE also succeeded in capturing the blunt characteristics of the background and the motions of the actions. However, the generated images in (b) suffer from large amount of noise for some images it is difficult to recognize the motion (circled in red). Demonstrated are also instances in which the order of the image was not matched (circled in blue), and instances in which the background of the image was not matched (circled in green). The images in the box show the samples of reconstruction results for given image pairs. Both, the proposed method and R-VAE successfully reconstructed the images, but the regression performance is largely different. As in (c), MOGP was not successful in describing the motion changes in the image, where every

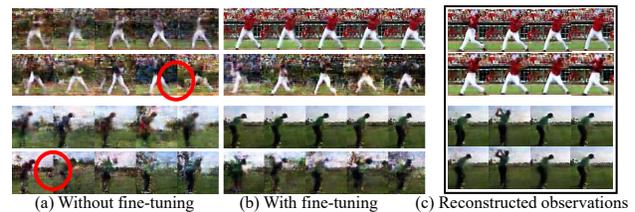


Figure 8. Analysis on the effect of fine-tuning. (a), (b): the regression result of the proposed method is shown in the first row and that of R-VAE in the second row. (c): the images in the box denotes the samples of reconstruction results for observed images.

regression converged to the average of the training images.

We also conducted the experiments comparing with Nearest Neighbor (NN) method results to the proposed method. We investigated the reconstruction results by applying the NN to the latent space after VAE learning. However, the latent space encoded by the vanilla VAE is not appropriate enough to perform regression using the NN (see Fig. 7). This is because the encoding of the background region plays a dominant role in NN as compared to the motion region. This problem is clearly seen at the bottom right sequence in the Fig. 7. Although the background (green and sky) region is relatively similar to the observation, the swinging human regions are not correctly regressed. This clearly shows that the proposed regression in the latent space is well performed achieving expected regression results in the image space. The encoder and decoder are trained to link the regression results in the latent space to the regression results in the image space directly, which is not trivial as shown in Fig. 7 (b). This is the second contribution of the proposed method (Abstract-ii).

Fig. 8 shows the effect of the fine-tuning process. The

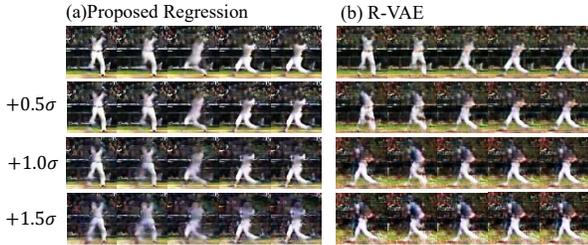


Figure 9. Results from  $+0.5\sigma$ ,  $1.0\sigma$  and  $1.5\sigma$  latent sample.

Table 1. Measure for the results with / without background.

Structural Similarity Index Measure [41] result				
sports	Proposed	R-VAE [24]	MIGP [2]	NN
Baseball	<b>0.610 / 0.607</b>	<b>0.492 / 0.489</b>	0.803 / 0.247	0.215 / 0.210
Golf	<b>0.752 / 0.707</b>	<b>0.578 / 0.543</b>	0.845 / 0.114	0.244 / 0.213
Snatch	<b>0.377 / 0.369</b>	<b>0.207 / 0.205</b>	0.626 / 0.019	0.206 / 0.198

first and second column show the results with and without fine tuning. The result of the proposed method is shown in the first row and that of R-VAE in the second row. Before fine tuning, both methods generated noisy outputs, but the proposed method captured the vast characteristics of the background as well as the change of the motions. In R-VAE, background information was less accurate than the proposed method (circled in red). After the fine-tuning process, both methods accurately reconstructed the given image pairs, as in (c). Nevertheless, the regression performance between the methods varied significantly, as in (b).

Fig. 9 represents the image generation results for different standard deviations. As with the original GP regression, the proposed method estimates the output responses in the form of mean and variance because the latent  $z$  for reconstructing the image is sampled from Gaussian distribution, as in (7). As seen in (a), the proposed algorithm captured the core semantics of the motion in each image despite the deviation change. In R-VAE, the regression results were plausible when the sampled latent  $z$  was close to the mean, but the motion in the image was regressed by a totally different action when adding large amounts of noise (up to  $1.0\sigma$ ). From this result, we can see that R-VAE also has an ability to align the images in the latent space according to their order as reported in previous works [24, 13]. However, the results also show that the learned variance of R-VAE does not represent the motion semantics required for regression well, which is essential for the realization of GP regression in the image space.

**Quantitative Analysis:** The quantitative performance was measured using the Structural Similarity Index Measure (SSIM) [41] which captures the structural similarities between two images. We estimated the 100 images in the test set by using their domain information only, and compared the similarity between the ground truth image and the regression results. Table 1 shows the performance mea-

Table 2. Measure for images from  $+0.5\sigma$ ,  $+1.0\sigma$  and  $+1.5\sigma$ .

SSIM result for different standard deviations				
sports	method	$+0.5\sigma$	$+1.0\sigma$	$+1.5\sigma$
Baseball	proposed	0.6453	0.5980	0.5307
	R-VAE [24]	0.4993	0.4402	0.3825
Golf	proposed	0.7203	0.4839	0.4422
	R-VAE	0.5642	0.4026	0.2417
Snatch	proposed	0.4042	0.3656	0.3629
	R-VAE	0.2700	0.1645	0.0770

asures for generated regression images. For the three different sport sequences, the proposed method generated more similar images to the ground truth (GT) compared to R-VAE. Interestingly, the results of MOGP [2] which converged to the average of the images were measured to be most similar among the tested methods when including the background. This is because the background of the average image is almost the same as the background of the GT when the background of the GT is fixed. When we measured the similarity without the background region, MOGP was not successful and the proposed algorithm achieved the highest performance. Also, as with the result shown in Fig. 7, NN method marked unsuccessful quantitative performance. Table 2 and Fig. 9 show the performance when changing the standard deviation. We confirmed that the proposed method generated more plausible output than R-VAE for all cases.

## 4.2. Human Pose Reconstruction

**Evaluation Scenario:** For this experiment, we have used the human 3.6 million (H3.6m) [20] dataset for generating proper human appearance given the joint positions. The dataset provides 32 joint positions, and thus the input data lie in 96 dimensional space. The dataset includes diverse actions, and each action is repeatedly performed by different actors. Our goal is to estimate the proper image of a new skeleton by utilizing the observed pairs of joint positions and images. In the experiment, we used the ‘greeting’ and ‘posing’ scenarios of the H3.6m dataset. The scenario for each actor was captured in 8 different view-points, resulting in a total of 16 human pose sequences available for each actor. We trained the model with the motions of 4 different actors using 12 sequences from each actor. Then, we picked the observations from the remaining four sequences and conducted the regression. The joint vectors for the regression were selected from the sequences from which the observations were selected. The joint vectors from other actors were also tested. For comparison, we used the recent conditional VAE (C-VAE) [43] method, which generated an image according to a given attribute coupled with the sampled latent code. In this experiment, the joint vector was used as the attribute.

**Qualitative Analysis:** Fig. 10 (A) shows the pose generation result of the proposed algorithm and C-VAE. For

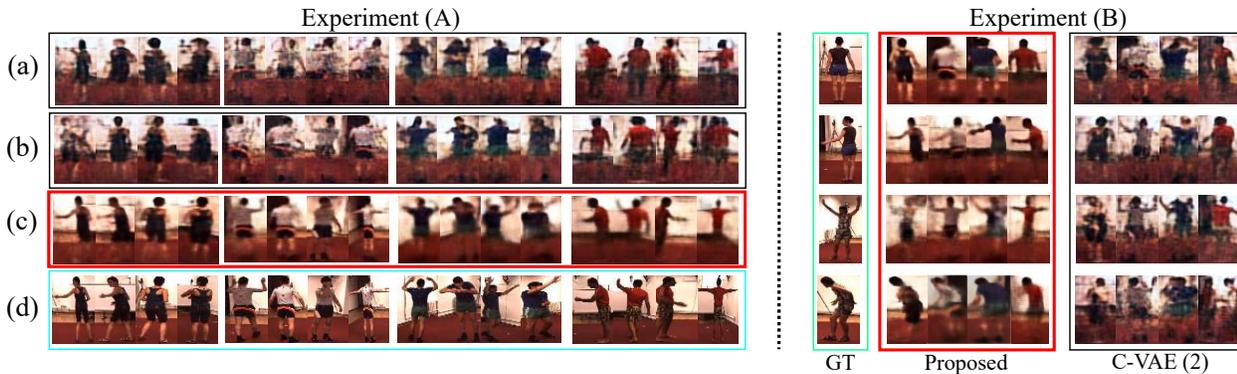


Figure 10. Human pose estimation results from the joint (best viewed in color). The images in row (a) represents the C-VAE (1) results. The images in rows (b) is result from the CVAE (2). The row (c) is the result from proposed method. The row (d) shows the ground truth.

Table 3. Similarity measure for generated human pose image.

Actors	SSIM result for human pose generation			
	proposed(A)	CVAE(A)	proposed(B)	CVAE(B)
#1	0.7402	0.4849	0.5227	0.4059
#2	0.6743	0.4265	0.4775	0.3580
#3	0.7295	0.5094	0.5013	0.4268
#4	0.7671	0.4954	0.5224	0.4198

C-VAE (1), we used randomly sample latent code  $z_y$  as in [43]. For C-VAE (2), the latent code was given by the proposed regression block in Fig. 2. As shown in (c), the image regressed by the proposed method successfully describes the overall motion of each human pose. Also, note that the background of each image was correctly generated according to the view point of the observed data pair. The generated images from C-VAE (2) contain a large amount of noise, but they captured the rough silhouette of the actors. This result is noticeable because C-VAE usually deals with cases in which the attribute is discrete. The result from C-VAE (2) was clearer than the result from C-VAE (1), but the difference was not significant. The result in Fig. 10 (B) shows the output responses when the joint vectors of other actors were given. The images in the blue box refer to the ground truth pose, and the images in the red box are the regression result by the proposed method. This result shows that the proposed method generates poses that resemble those of the input joint vectors while preserving the appearance of the given data pairs via regression. Specifically, when the given pair involves a man wearing white clothes, the generated image illustrates a man wearing the same clothes with a similar pose to the GT image. C-VAE (2) was not successful in generating a corresponding pose for a given joint from other actors.

**Quantitative Analysis:** Table 3 shows the similarities between the generated image and the ground truth image. The first two columns denoted by (A) represent the quantitative results in experiment (A) of Fig. 10. In the experiment, the proposed method achieved a higher score than C-VAE (2). For experiment (B), we compared the similarity between the

regressed image and the original images for the joint vector (green box in Fig. 10). There, our method also achieved a higher score than C-VAE (2).

In this experiment, the input data lay in the high dimensional space and the target joint vectors were selected without considering temporal information. Despite the complicated and non-sequential input domain, the proposed regression method achieved reasonable output responses describing the semantics given in the input and the identity information contained in the observed pairs. It means that the proposed method is available for the temporal input and can also handle more complex and non-sequential input.

## 5. Conclusion

In this paper, we have proposed a novel regression method regarding high dimensional visual output. To tackle the challenge, the proposed regression method is designed so that the result of the regressed response in a latent space and the corresponding response in the data space coincide. Through qualitative and quantitative analysis, it has been verified that our method properly estimates the regressed image responses and offers an approximation of the complicated input-output relationship. This paper discovers meaningful progress in the regression field in that our work introduces a way to combine a deep layered architecture to the regression method in a probabilistic framework.

## 6. Acknowledgment

This work was partly supported by the ICT R&D program of MSIP/IITP (No.B0101-15-0552, Development of Predictive Visual Intelligence Technology), the SNU-Samsung Smart Campus Research Center at Seoul National University, EU FP7 project WYSIWYD under Grant 612139 and the BK 21 Plus Project. We thank the NVIDIA Corporation for their GPU donation.

## References

- [1] M. Alvarez and N. D. Lawrence. Sparse convolved gaussian processes for multi-output regression. In *Advances in neural information processing systems*, pages 57–64, 2009. 1, 2
- [2] M. A. Alvarez and N. D. Lawrence. Computationally efficient convolved multiple output gaussian processes. *Journal of Machine Learning Research*, 12(May):1459–1500, 2011. 1, 2, 6, 7
- [3] M. A. Alvarez, D. Luengo, M. K. Titsias, and N. D. Lawrence. Efficient multioutput gaussian processes through variational inducing kernels. In *AISTATS*, volume 9, pages 25–32, 2010. 1, 2
- [4] Y. Anzai. *Pattern Recognition & Machine Learning*. Elsevier, 2012. 2
- [5] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966. 2
- [6] M. J. Beal. *Variational algorithms for approximate Bayesian inference*. University of London London, 2003. 4
- [7] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009. 5
- [8] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003. 2
- [9] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015. 2
- [10] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011. 5
- [11] M. Germain, K. Gregor, I. Murray, and H. Larochelle. Made: masked autoencoder for distribution estimation. In *International Conference on Machine Learning*, pages 881–889, 2015. 2
- [12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 2
- [13] R. Goroshin, M. F. Mathieu, and Y. LeCun. Learning to linearize under uncertainty. In *Advances in Neural Information Processing Systems*, pages 1234–1242, 2015. 2, 7
- [14] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1462–1471, 2015. 2
- [15] H. He and W.-C. Siu. Single image super-resolution using gaussian process regression. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 449–456. IEEE, 2011. 1
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2
- [17] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002. 1, 2
- [18] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. 1, 2
- [19] P. W. Holland and S. Leinhardt. An exponential family of probability distributions for directed graphs. *Journal of the american Statistical association*, 76(373):33–50, 1981. 2
- [20] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014. 7
- [21] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960. 2
- [22] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka. Stock market prediction system with modular neural networks. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, pages 1–6. IEEE, 1990. 1
- [23] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations*, 2014. 2, 4, 5
- [24] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *Proceedings of the 2nd International Conference on Learning Representations*, 2013. 2, 3, 4, 6, 7
- [25] R. G. Krishnan, U. Shalit, and D. Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015. 2
- [26] N. D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *Advances in neural information processing systems*, 16(3):329–336, 2004. 2, 5
- [27] Y. Li, K. Swersky, and R. Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, pages 1718–1727, 2015. 2
- [28] A. W. Lo and A. C. MacKinlay. Stock market prices do not follow random walks: Evidence from a simple specification test. *Review of financial studies*, 1(1):41–66, 1988. 1
- [29] S. N. MacEachern and P. Müller. Estimating mixture of dirichlet process models. *Journal of Computational and Graphical Statistics*, 7(2):223–238, 1998. 2
- [30] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 2
- [31] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4293–4302, 2016. 2
- [32] Y. Pu, Z. Gan, R. Hénao, X. Yuan, C. Li, A. Stevens, and L. Carin. Variational autoencoder for deep learning of images, labels and captions. In *Advances in Neural Information Processing Systems*, pages 2352–2360, 2016. 2
- [33] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

- [34] C. E. Rasmussen. The infinite gaussian mixture model. In *NIPS*, volume 12, pages 554–560, 1999. [2](#)
- [35] C. E. Rasmussen. Gaussian processes for machine learning. 2006. [2](#), [4](#), [5](#)
- [36] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1278–1286, 2014. [2](#)
- [37] R. Salakhutdinov. *Learning deep generative models*. PhD thesis, University of Toronto, 2009. [2](#)
- [38] R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. In *AISTATS*, volume 1, page 3, 2009. [2](#)
- [39] K. Swersky, J. Snoek, and R. P. Adams. Multi-task bayesian optimization. In *Advances in neural information processing systems*, pages 2004–2012, 2013. [1](#), [2](#)
- [40] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the american statistical association*, 2012. [2](#)
- [41] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. [7](#)
- [42] B.-C. Wei. *Exponential family nonlinear models*, volume 130. Springer Verlag, 1998. [2](#)
- [43] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. *arXiv preprint arXiv:1512.00570*, 2015. [2](#), [3](#), [7](#), [8](#)
- [44] H. Yang, L. Chan, and I. King. Support vector machine regression for volatile stock market prediction. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 391–396. Springer, 2002. [1](#)