

# Constructing High Dimensional Random Functions With Artificial Neural Network

**Junyu Liu<sup>a</sup>**

<sup>a</sup>University of Science and Technology of China,  
Hefei, Anhui 230026, P.R.China

E-mail: [junyu@mail.ustc.edu.cn](mailto:junyu@mail.ustc.edu.cn)

**Abstract.** We present a brief review on several Monte Carlo approaches to practically generate random functions, especially in high dimensional case. A new method with the use of artificial neural network is also originally given. In order to show application in physics, we use the random function to simulate the string landscape (the complicated high-dimensional potential of the vacuum in string theory) and study the inflationary trajectories in the early universe.

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Random Functions and Classification</b>	<b>2</b>
<b>3</b>	<b>Interpolation Trial</b>	<b>3</b>
<b>4</b>	<b>Artificial Neural Network</b>	<b>5</b>
4.1	Introduction	5
4.2	Function Generation	8
<b>5</b>	<b>Application in String Landscape</b>	<b>10</b>
5.1	Inflation, String Phenomenology and Multi-stream Formulism	10
5.2	Monte-Carlo Simulation	13
<b>6</b>	<b>Conclusion and Outlook</b>	<b>15</b>

---

### 1 Introduction

Just like random number generation and its application in a large range of different areas, random function generation, namely choosing functions randomly from some function classes on the bounded (or infinite) domain (eg,  $C^1$  class on the interval  $[0, 1]$ ), plays a crucial role in both mathematics and physics. Although formal study of random function in mathematics [1] is hard and technical because of the non trivial topology of function space [2] (which still appears in Feynman path integral), several methods, including sequential approach [3], Gaussian Wiener integrals [4], Parseval duality [5], p-adics theory [6] and Feynman motive [7].

On the practical side, the most celebrated use of random functions may be the description of disordered systems in solid state physics leading to Anderson localization [8] (often Gaussian random potentials based on a truncated Fourier series are used). Motivated by string landscape [9] (The extremely complicated background in the vacuum of string theory), some cosmologist have studied the random inflation [10], with the use of random function techniques, such as interpolation method [11, 12], Dyson Brownian motion [13, 14], and Gaussian random field [15]. However, as far as we know, neither formal configurations nor these practical approaches can give us a way to generate (pseudo) random functions, especially high dimensional functions, at the polynomial complexity of dimension  $d$ .

Here we originally present two methods to generate random functions at the arbitrary dimension. The first approach stems from the help of artificial neural network (ANN) [16, 17], a powerful function simulator inspired by the sophisticated functionality of human brains where hundreds of billions of interconnected neurons process information in parallel. If we use the feed-forward neural network with one hidden layer to construct a vector-machine-like architecture, we can effectively generate global random functions with the complexity  $\mathcal{O}(hd)$  where  $h$  is the number of neurons in the hidden layer. We find this method is extremely useful in building global random functions.

As an example, we also present simple applications of ANN approach in the simulation of random string inflation theory [20]. In this model, the inflaton (a scalar field which dominates

the dramatically expanding phase of the early universe) is constrained by the Friedmann equation with the random potential from string landscape. If we want to avoid eta problem (the mass instability of inflaton in the whole inflation process), we may need the high dimension field space with the multi-stream consideration [21]. This is the source of high-dimensional random potential. We simulate the inflation trajectory with the ANN approach, which shows a high speed and efficiency especially for high dimension.

In this paper, we manage these contents as following. In Section 2, we present a simple review on random functions and its classification. In Section 3, we use the interpolation approach to introduce the random generation in the low dimension. In Section 4, we present the basic introduction of artificial neural network and show the algorithm to generate random functions in the higher dimension. In Section 5, we present the application of our ANN random functions in a specific physical problem, the inflation trajectory in a string landscape. In Section 6, we conclude our results and give some outlook.

## 2 Random Functions and Classification

What is the definition of random functions?

Qualitatively speaking, random function is the function randomly chosen from a certain function space with a certain functional distribution. For example, if we want to define random numbers uniformly living in  $[0, 1]$ , we should consider a process of choosing variables from  $[0, 1]$ . Each real number in  $[0, 1]$  has the same probability to be chosen. Similarly, we can define random functions uniformly living in the function space  $C^1[0, 1]$ . Each order-one derivative function with the domain  $[0, 1]$  has the same probability to be chosen.

Generally speaking, consider a function space  $\Lambda[X^d]$ , where  $X^d$  is a general subset (often compact or open) of  $d$ -dimensional Euclidean space  $\mathbb{R}^d$ . For  $d$ -dimensional function  $\psi \in \Lambda[X^d]$  We have a *functional distribution*  $P[\psi]$ , which is a functional of  $\psi$  as analogy of random distribution for random variables

$$\int \mathcal{D}\psi P[\psi] = 1$$

This kind of integration is defined as the Feynman path integral in quantum mechanics. The integration measure,  $\mathcal{D}\psi$ , is called Wiener measure [4] defined formally in mathematics as a significant kind of infinite-dimensional integration measure. The functional  $P[\psi]$  labels the probability of  $\psi$ 's neighborhood to be chosen in the function space.

It is extremely hard to study functional distribution formally. But on the practical side, random functions always appear in some differential equations in large range of areas, so we often make the function space approximately  $\Lambda[X^d] = C^m[X^d]$ , the  $m$ th order derivative function, and  $X^d$  is a simple closed interval in  $\mathbb{R}^d$ .

The simplest example is the Langevin equation of a Brownian particle in one-dimensional space

$$m\ddot{x} = -\alpha\dot{x} + F(t)$$

Here  $F(t)$  is a random function living in the interval  $[t_i, t_f] \subset \mathbb{R}$ . However,  $F(t)$  is an arbitrary function without continuous constrain, so it is a Markov process, and can directly solved by difference method with MCMC. However, if we need to assume that  $F(t)$  is a *continuous* function in some systems, things are different.

Moreover, one can consider a special system with order-one derivative function  $F(t)$ , and the friction coefficient is also related to function  $F$  or  $\dot{F}$

$$m\ddot{x} = -\alpha[F, \dot{F}]\dot{x} + F(t)$$

we must use the totally different random function generation for this non-Markov process. This introduction is heuristic, and also not unphysical. In fact, inflation dynamics is a non-Markov process of this kind. We will show the applications of random function in the Section 5.

Some works have been done to generate random functions. One can classify them into two categories. The first of them is called the global method, namely, generate the random functions on the total domain. The most naive but safe method of this kind is interpolation, which is widely used in other researches (eg. [11, 12]) and has no subtleties such as self-crossing. However, this method will lead an exponential computational expense so it is hard to use on high dimensional case. The second method is called the local method. This algorithm fails to generate a random function globally, but generate it locally around a high-dimensional trajectory to study random differential equations. This kind of methods can dramatically decrease the expense we pay in time, but may cause some new problems. For example, the Dyson Brownian motion approach [13, 14] use the Wigner ensemble to truncate patches on a trajectory as Taylor expansion, and glue them together patch by patch. However, it cannot solve problems like trajectory-crossing and uniqueness of solutions because different the same variables at different trajectories must have different random function values. The gaussian field approach (only appearing in this paper [15] as far as we know) is used to study random supergravity, which can avoid self-crossing problem. Nevertheless, it assumes that random field must be Gaussian and function values must be decided point by point. So it can only be used in some special problems with strong constraint. In Section 4, we construct a new method inspired by ANN, which belongs to the global method. We find that this method is extremely useful in random construction.

### 3 Interpolation Trial

In this part, we perform a simple trial to show the random function generation with ordinary interpolation method in the low dimension [11].

The algorithm can be simply described as following. First, we assume the dimension  $d = 1$  for example. Consider smooth functions on the interval  $I = [a, b]$ . First, we split the domain  $[a, b]$  into  $N$  small intervals. For each interval  $I_i$ , we generate a random number  $f_i$  living in some specific distributions (Gaussian or uniform distribution). Finally, we do the interpolation of series  $(a_i, f_i)$ , where  $a_i$  is a characteristic point of  $I_i$  (eg, we can choose  $a_i$  as the middle point of  $I_i$ ). After interpolation, we gain a smooth curve, which can be regarded as a one-dimensional random function construction living in the function space  $C^m[a, b]$  (where  $m$  depends on the interpolation algorithm we choose).

This method can be easily generalized to high-dimensional case, where the interpolation and sub-intervals should be high-dimensional. For instance, we can interpolate a smooth curve as a two-dimensional random function where two-dimensional domain is split to many small boxes. Moreover, we can interpolation on one fixed surface with some random bumps.

For example, we choose the quadratic potential surface on the domain  $[0, \varphi_{\max}]^2 \subset \mathbb{R}^2$

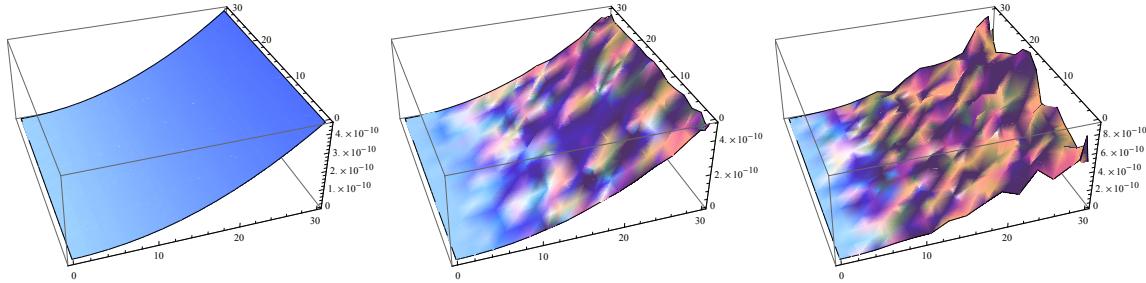
$$V(\varphi_x, \varphi_y) = \frac{1}{2} m^2 \varphi_x^2$$

and after random perturbations, we have the interpolation

$$V_{ij} = \frac{1}{2} (1 + \mathcal{A} \text{Rand}_{ij}) m^2 (i \Delta_x \varphi)^2$$

where  $\Delta_x \varphi = \frac{\varphi_{\max}}{N}$  is the distance for each adjacent pair of two-dimensional intervals,  $\text{Rand}_{ij}$  are random numbers in  $[0, 1]$  chosen for each  $(i, j)$  and belong to one fixed distribution, and  $\mathcal{A}$  is the amplitude of randomness.

Shape of figures with different randomness are shown as Figure 1.



**Figure 1.** Potential surface with null, small and large randomness. We choose  $\Delta_x \varphi = 0.05$ ,  $m = 10^{-6}$ ,  $\varphi_{\max} = 30$ , and  $\mathcal{A} = 0, 0.1, 1$  respectively for the left, middle and right figure. The distribution is uniform for random numbers in the interpolation function, and the interpolation order is  $C^1$ .

Exponential Disaster  $\mathcal{O}(e^d)$  However, this method is practically useless for high-dimensional case. Just see the number of random variables related on the dimension  $d$ . If we choose  $N$  intervals on each direction, we will gain  $N^d$  high-dimensional intervals. As a result, the number of random variables are  $N^d$ . And we must do high-dimensional interpolation on these  $N^d$  random numbers. So the complexity is exponentially blowing up with the dimension  $d$ . The same situation happens on other global methods. Taking classical Fourier series For example, the one-dimensional Fourier expansion is

$$f(x) = \sum_{n=1}^N a_n e^{inx}$$

So we have  $N$  Fourier coefficients after truncation. The two dimensional Fourier expansion is

$$f(x_1, x_2) = \sum_{n_1 n_2}^N a_{n_1}^{(1)} a_{n_2}^{(2)} e^{in_1 x_1} e^{in_2 x_2}$$

So we have  $N^2$  Fourier coefficients after truncation. Generally, the  $d$  dimensional Fourier expansion is

$$f(x_1, x_2, \dots, x_d) = \sum_{n_1 n_2 \dots n_d}^N \prod_{j=1}^d a_{n_j}^{(j)} e^{in_j x_j}$$

So we will have  $N^d$  Fourier coefficients.

When we want to use classical Fourier series to simulate the high-dimensional random function, we must set all the coefficients to be random numbers and the complexity will be  $\mathcal{O}(e^N)$ . As a result, we need a new way to construct high-dimensional random functions.

## 4 Artificial Neural Network

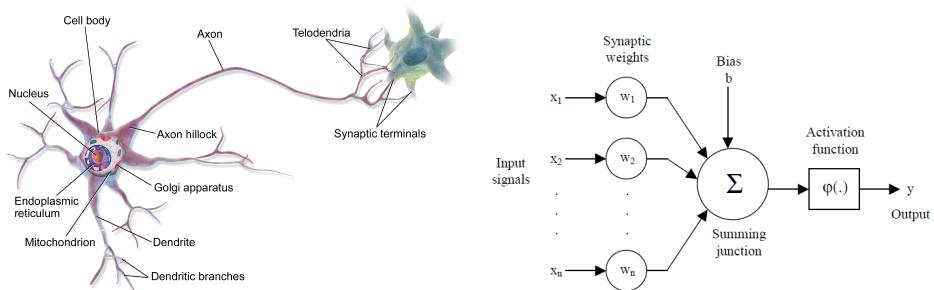
In this section, we will provide a new way construct random functions inspired by the celebrated Artificial Neural Network (ANN) [16, 17]. First, some basic ideas of the simplest ANN configurations will be introduced. Then, we will present our result on random function generation with the help of ANN.

### 4.1 Introduction

Artificial Neural Network (ANN) is a hot topic [16–18] in the machine learning area. Generally speaking, ANNs has been inspired from its inception through calculation and identification of the human brain in a completely different way from the traditional digital computer (Von Neumann). The human brain is a highly *complex, nonlinear and parallel* computer (namely, the information processing system). It has the ability of neurons, in order to perform some specific calculations (such as pattern recognition, perception, and motor control) whose behavior are many times faster than the fastest computers that exist nowadays. In its most general form, an ANN is designed as a kind of way, to perform a specific task or function of interest-model-machine of human’s brain.

ANN is proposed for a technology in neural science, mathematics, statistics, physics, computer science and engineering [17], which is also of benefit to many other areas. ANN has important applications in many fields, including time series analysis, pattern recognition, signal processing, and control theory etc., controlled by a significant property: the capacity to know from data *with or without teachers*.

ANNs also have very strong power in the function approximation area. In multilayer feed-forward architecture, we have the celebrated *universal approximation theorem*, which can be used to approximate arbitrary functions at the arbitrary accuracy. This is our theoretical foundation of function generation. We will leave it at the last part of this section for detailed introduction.



**Figure 2.** Description figures of true biological neuron(left) and ANN mathematical neuron model(right).

asfsd  
fdsf

First let us talk about neurons. A neuron is an information-processing unit, which is extremely crucial to the operation of an ANN or natural NN. The left of Figure 2 shows the natural shape of a neuron, which is the basic picture for ANN. The artificial neurons we use to build our ANNs are truly primitive in comparison to those found in the brain.

An artificial neuron has several inputs but only one output besides the following four important parts:

- (1). *Synapses*, which are labelled by some weights. To specify, a signal  $x_j$  at input of synapse  $j$  linked to a neuron is timed by the corresponding weight  $w_j$ .
- (2). An *adder*  $\Sigma$  for addition of the input signals.
- (3). An *activation function*  $\varphi$  for influencing the norm or shape of the output of a neuron.
- (4). An external *bias*, denoted by  $b$ , which has the dramatic influence of the input contribution to the activation function.

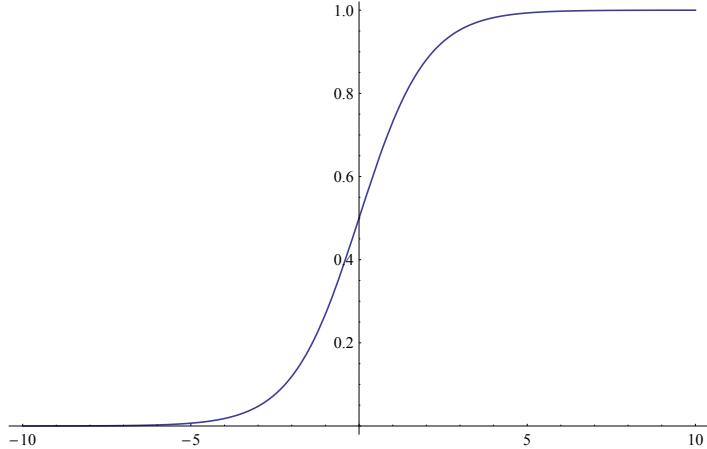
In mathematical words, for  $d$ -dimensional signal vector  $\{x_j\}_{j=1}^d$ , we may describe a neuron output  $y$  by

$$y = \varphi \left( \sum_{j=1}^d w_j x_j + b \right)$$

Generally, we can choose the activation function  $\varphi : \mathbb{R} \mapsto \mathbb{R}$  as a bounded function with different finite limit at positive and negative infinity. However, if we need to construct the random functions on the function space  $C^k[X^d]$ , or more strong constraint,  $C^\infty[X^d]$ , we often choose the *Logistic function*

$$\varphi(x) = \frac{1}{1 + e^{-x}}$$

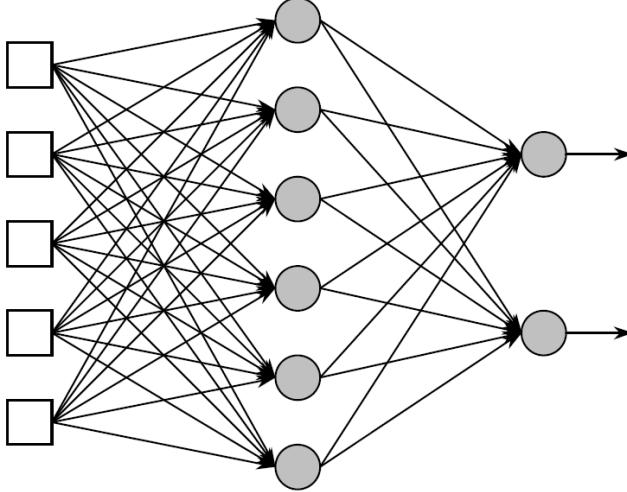
This is the most common form of activation function.



**Figure 3.** Graph of Logistic function.

Now we introduce the *multilayer feed-forward architecture*. In the organizational structure of a layered neural network of neurons, we have at least two layers: an input and an output layer. The layers between the input and the output layer are called *hidden layers*. The original nodes in the input of the ANN provide respective elements of the input, which form the information to the second layer, also the first hidden layer. The output information of the second layer is used as input to the third layer, which is the same as the rest of the ANN. A layer of nodes projects onto the next layer of neurons, but not vice versa. In other words, this kind of ANN is formally a *feed-forward* type, which has only one direction. The neurons in every layer of the network have as their inputs the output information of the preceding layer only. The set of output information of the neurons in the ultimate layer of the network forms the global reflection of the ANN to the stimulation pattern provided by the original

nodes in the first layer. An ANN with this kind of architecture is also called as a *multilayer perceptron*.



**Figure 4.** A fully connected feed-forward neural network with one hidden layer.

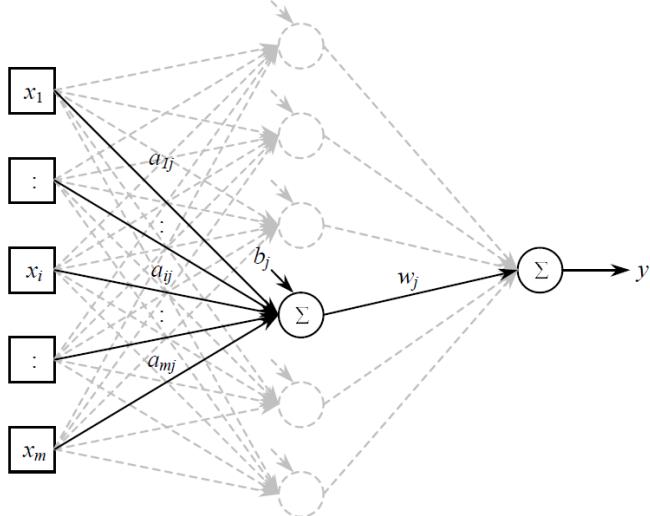
The function of the hidden layer neurons are some useful network between the intervention of external input and output. By adding one or more hidden layers, so the network can extract higher order statistics. To obtain the global perspective in a relatively loose significance in the network, although we still have more local connections because of the extra dimensions of extra synaptic and neural interactions. The neural network is said to be fully connected in the meaning of mathematics and graph theory, that each node of the network links to each other node in the neighborhood's forward layer. Otherwise the network is called partially connected. Figure 4 shows the pattern of a fully connected feed-forward neural network with one hidden layer.

In practice, we need to construct the random function with only one output, so we formalise the output of a feedforward neural networks with one hidden layer and with one linear output unit as a function of the inputs as in Figure 5. Namely, if we have a multilayer perceptron neural network with one hidden layer that consists of  $h$  hidden units, and the network has  $d$  inputs, then we can formalise the output of the network as a function of the inputs by

$$A_h(x_1, x_2, \dots, x_d) = \sum_{j=1}^h w_j \varphi \left( \sum_{i=1}^d a_{ij} x_i + b_j \right)$$

where  $a_{ij}$  is the weight of the synapse that goes form the input  $x_i$  to the  $j$ th hidden neuron,  $b_j$  is the bias of the  $j$ th hidden neuron,  $\varphi$  is the activation function, and  $w_i$  is the weights of the synapse that goes to the output neuron to form the  $j$ th neuron.

For this configuration, we have the simplest *universal approximation theorem*. The universal approximation theorem claims that the standard multilayer feed-forward networks with a single hidden layer that contains finite number of hidden neurons, and with arbitrary activation function are universal approximators. This property is not the specific choice of the activation function, but rather the multilayer feed-forward architecture itself which gives neural networks the potential of being universal approximators [23]. However, for our motivation we can choose the activation function to be the Logistic function. We have,



**Figure 5.** A feedforward neural network with one hidden layer.

Let  $\varphi$  be the Logistic function. Let  $X^d \subset \mathbb{R}^d$  and  $X^d$  be compact. Then for  $\forall k = 1, 2, \dots, \infty$ , for  $\forall f \in C^k[X^d]$ , for  $\forall \epsilon > 0$ :  $\exists h \in \mathbb{N}$ ,  $a_{ij} \in \mathbb{R}$ ,  $b_j \in \mathbb{R}$ ,  $w_j \in \mathbb{R}$ ,  $i \in \{1, 2, \dots, d\}$ ,  $j \in \{1, 2, \dots, h\}$ :

$$\forall \vec{x} = (x_1, x_2, \dots, x_d) \in X^d : |f(\vec{x}) - A_h(\vec{x})| < \epsilon.$$

Consider the limitation of space, we cannot perform the proof. However, this proof can be checked from thousands of ANN study on the formal side (eg,[23]) for much more general case. This result could be our theoretical foundation in the random function construction.

#### 4.2 Function Generation

In this subsection we will present our new random function generation approach. From the universal approximation theorem mentioned in the last subsection, one can approximate any arbitrary continuous or  $k$ th order continuous function by choosing different parameters  $h, a_{ij}, b_j, w_j$ . So as a result, if we choose these parameters randomly, we can construct a smooth random function living in some functional distribution. This is the basic idea of our random generation method.

Some comments should be given. First, the number of neurons in the hidden layer  $h$ , should be fixed in practical generation. The reason is,  $h$  can be roughly regarded as the *degree of randomness*. One can directly see that if  $h$  is small, the function  $A_h$  we made is extremely simple. While if  $h$  is large enough, many kinds of Logistic functions as the linear basis will give us a very complicated random phase. Different problems need different degree of randomness, and  $h$  could be given as an adjustable variable for users.

Second, the range of parameters  $a_{ij}, b_j, w_j$  should be bounded. Indeed, we cannot generalize random variables living in the full  $\mathbb{R}$ , and practically, we only consider the function bounded on a compact set. So points at the infinity are useless for our purpose. For instance, one can assume that

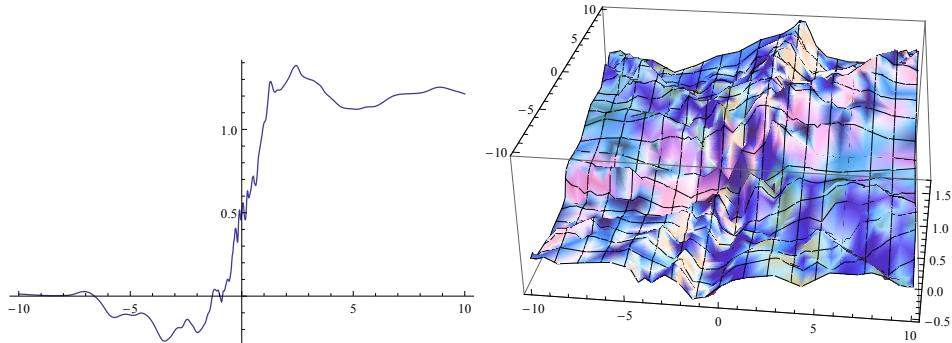
$$a_{ij} \in [-\alpha, \alpha] \quad b_j \in [-\beta, \beta] \quad w_j \in [-\omega, \omega]$$

and each parameter is living in the uniform distribution. In the following numerical experiments, we always use this constrain.

Third, the complexity is  $\mathcal{O}(hd)$ , because we will use only  $hd+2h$  random numbers. To specify, in a symbolic computation system (like Mathematica), if we assume that when we compute  $\varphi(x)$  once for some  $x$ , we need the time  $t_1$ ; when we compute the addition or multiplication once, we need the time  $t_2$ ; when we generate a random number, we need  $t_3$ , then for a total construction process of one random function, we need time

$$hd(2t_2 + t_3) + h(t_1 + 2t_2 + 2t_3) - t_2 = \mathcal{O}(hd)$$

This is a rough estimation because it depends on which symbolic system we use, so the complexity may larger than  $\mathcal{O}(hd)$ . However, we should believe that the random function can be generated in most of symbolic systems within polynomial time, which is impossible for other global methods. In Figure 6 we show some slices of a 100-dimensional random function generated by ANN. This result shows the strong power of ANN in random function generation.



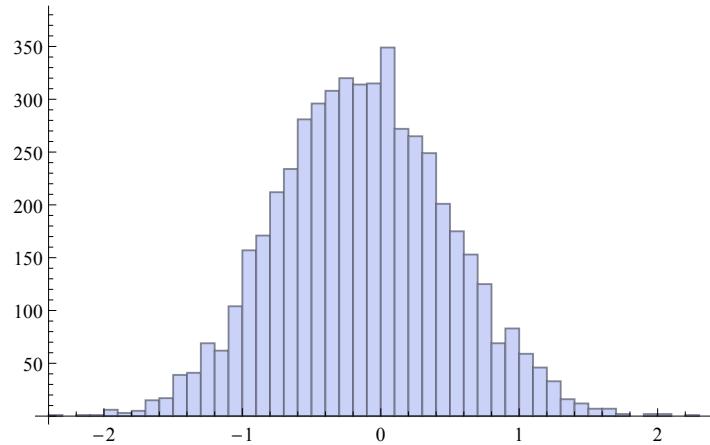
**Figure 6.** Some slices of a 100-dimensional random function. We generate a 100-dimensional random function with parameters  $\alpha = 50$ ,  $\beta = 50$ ,  $\omega = 0.05$ ,  $h = 1000$ . The left graph is the function  $A_h(0, 0, \dots, x, x)$  (The first 98 components of the variable vector are all zero and the last two components are chosen to be the same), while the right graph is the two-dimensional function  $A_h(0, 0, \dots, x, y)$  (The first 98 components of the variable vector are all zero and the last two components are chosen to be  $x$  and  $y$ ).

Another aspect, about the distribution. How to formally derive the functional distribution of the ANN-generating functions? As far as I know, our work is the first work in ANN-random function generation, so there is no formal study before. Consider the extreme hardness of functional integration in mathematical rigor, we leave this work for future research. However, at least we should study the *range* of random functions. In fact we even cannot find the analytic integration of

$$E(A_h)(x_1, x_2, \dots, x_d) = \int da_{ij} dw_j b_j \sum_{j=1}^h w_j \varphi \left( \sum_{i=1}^d a_{ij} x_i + b_j \right)$$

which is the *expectation value function* of random functions, even related to the Feynmann diagrams in the celebrated M-theory in string theory [24]. So we cannot control the range *formally*. However, we can deal with it *numerically* through Monte Carlo simulation before we generate some random functions. We find that because the basis function, the Logistic

function has a local sharp threshold around the origin. So we only need to consider the range near the origin. So for each parameter choice, the random function scale can be constrained in such a way. For example, we choose the parameters in Figure 6. Namely,  $\alpha = 50$ ,  $\beta = 50$ ,  $\omega = 0.05$ ,  $h = 1000$  for the 100-dimensional random function generation. To determine which kind of range scale it could have, we generate 100 times and get 100 100-dimensional random functions. For each function, we choose 50 random vectors randomly living in  $[-10, 10]^{100}$  with uniform distribution as Figure 7. One can find the range was constrained in  $[-2, 2]$ . Each kind of ANN-random functions can be used in such a way before doing simulation with it.



**Figure 7.** Range distribution of random functions. We choose  $d = 100$ ,  $\alpha = 50$ ,  $\beta = 50$ ,  $\omega = 0.05$ ,  $h = 1000$ . We generate 100 different functions and for each function we randomly choose 50 points living in uniform  $[-10, 10]^{100}$ . The horizontal axis of the histogram shows the number of function value, while the vertical axis shows the frequency.

To conclude, we find a way that has the strong power in random smooth function generation especially high dimensional, which is impossible for other alternative methods. In the next chapter, we will use it in a specific physical problem as a simple use of this ANN-random function strategy.

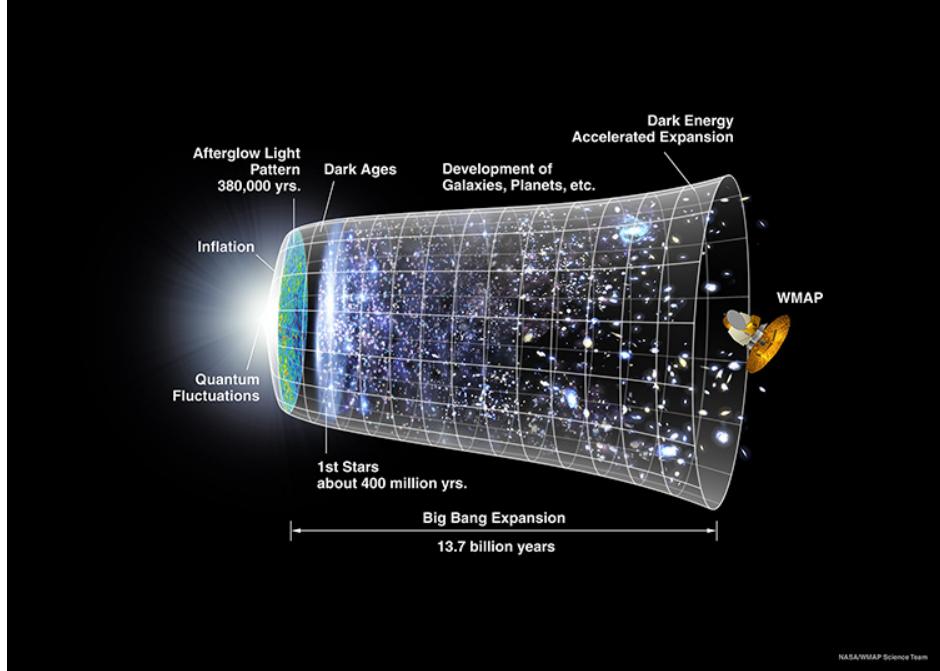
## 5 Application in String Landscape

In this section we will use our ANN-random functions to simulate the inflationary process with a random potential motivated by string landscape. In the first subsection, we will make a brief introduction on the string inflation theory. Then, a Monte-Carlo study will be presented.

### 5.1 Inflation, String Phenomenology and Multi-stream Formulism

Nowadays, inflation, the dramatic expansion of scale factor  $a(t)$  (which denoted the radius of universe) in the early universe, has become the most promising paradigm in the modern cosmology [20]. Historically, inflation theory was built by Alan Guth and his collaborator Henry Tye to solve the horizon and flatness problem in the classical cosmology [25]. Now this theory can explain almost all aspects of early universe, even formation of large scale structure, and even why we exist, and agreed with all observations of WMAP/Planck satellites and BICEP collaboration. In Figure 8 made by NASA we can see the evolution of scale factor in

the historical ages of universe.



**Figure 8.** The scale factor's evolution and the historical events in our universe.

The inflation theory, basically the slow-roll inflation, believes that the inflation process is dominated by a *scalar field*  $\varphi$ , namely, the *inflaton*, whose background (the classical part without the quantum fluctuations) satisfying the following *Friedman equation*

$$\ddot{\varphi} + 3H\dot{\varphi} + V'(\varphi) = 0$$

where the dot denotes the time derivative, while the prime denotes the  $\varphi$  derivative, and  $H$  is the celebrated *Hubble parameter*,  $H(t) = \frac{\dot{a}}{a}$ ,

$$H(t) = \frac{\dot{a}}{a}$$

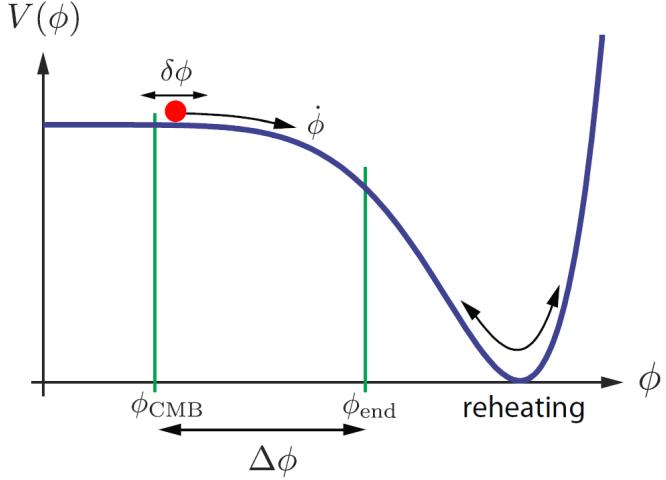
and  $V(\varphi)$  is the *slow-roll potential*, a positive function of inflaton  $\varphi$ . This equation describes the dynamics of inflation. The slow roll inflation  $V(\varphi)$  is model dependent. Indeed, we use this function to denote different inflation models.

If we want to satisfy the inflation condition (the exponential expansion of universe), we must assume that the derivative of slow-roll potential should be extremely small. This is called the *slow-roll inflation*. One can describe the process of inflation as a ball rolling down along a small-slope curve, as Figure 9.

In the slow-roll limit, we have

$$H = \sqrt{\frac{V(\varphi)}{3}}$$

when we set the Planck mass  $M_p = 1/\sqrt{8\pi G} = 1$ . So Friedman equation is a closed equation now when we assume the shape of potential and the initial condition is already given. However, in this model we have only one inflaton which will cause the instability of inflaton



**Figure 9.** The scale factor's evolution and the historical events in our universe.

mass. This is called *eta-problem*. In order to solve it, we should consider many inflatons  $\varphi_i$ , where one of them dominates the inflation process, called  $\varphi_1$ , while the others contributes much smaller. This is called the *multi-stream formulism* [21]. Now the Friedman equation becomes

$$\ddot{\varphi}_i + 3H\dot{\varphi}_i + \partial_i V(\varphi_1, \varphi_2, \dots, \varphi_d) = 0$$

where  $i = 1, 2, \dots, d$  for  $d$  inflatons. Notice that this equation is only background behavior of inflationary trajectory. In the early universe there are only fundamental particles, so we must use the quantum correction. Starobinsky [26] proved that, the quantum correction can be contained after addition a random source on the R.H.S of Friedman equation

$$\ddot{\varphi}_i + 3H\dot{\varphi}_i + \partial_i V = \frac{3}{2\pi} H^{\frac{5}{2}} \eta_i(t)$$

where the  $\eta_i$  term is the random source to denote the quantum corrections and follows independent Gaussian distribution. We construct the normalization condition to constrain the  $\eta_i$  term,

$$\langle \eta_i(t)\eta_j(t') \rangle = \delta(t-t')\delta_{ij}$$

such that during a Hubble time and averaged on a Hubble volume, the quantum fluctuation on each field direction is normalized as  $H/(2\pi)$ . In numerical calculation, the discrete time interval  $\Delta t = t_n - t_{n-1}$  has to be used. While replacing differential equations into difference equations, the Dirac delta function shall be replaced by

$$\delta(t-t') \rightarrow \frac{\delta_{ij}}{\Delta t}$$

So the difference strategy is

$$\frac{\varphi_i(n+1) + \varphi_i(n-1) - 2\varphi_i(n)}{2\Delta t^2} + 3\sqrt{\frac{V(\varphi_j(n))}{3}} \frac{\varphi_i(n+1) - \varphi_i(n)}{\Delta t} + \partial_i V(\varphi_j(n)) = \frac{3H^{5/2}}{2\pi} \eta_i$$

where  $\eta_i(n) \sim N(0, 1/\sqrt{\Delta t})$ , and Gaussian distributions are independent for different  $i$ . Now the only thing remaining is, what is actually the potential  $V$  like? Indeed, there are

at least 40 different kinds of inflation model with different  $V$  appeared in Arxiv, and the inflation mechanism is still a mystery. Some people believe that, the celebrated string theory, the most promising quantum gravity theory should give us an explanation because the inflation process with dramatic expansion and large energy scale around Planck mass  $M_p$  where string interaction appears. So many cosmologists use the string theory to build the inflation model and the effective potential  $V$  after dimensional reduction [20], including the first string inflation model, brane inflation [27], and the most successful model appearing recently, axion monodromy inflation [28]. These string theory models all stems from the topological structure of the stringy vacuum.

However, people nowadays have realized that the vacua of string theory is extremely complicated. The number of vacuum solution of string theory has at least  $\mathcal{O}(10^{100})$  (larger than total atoms in our universe) [29]. All of these ground states are degenerated, which is called the *string landscape* [9]. If each solution has a corresponding effective slow-roll potential  $V$ , we have  $\mathcal{O}(10^{100})$  different but equivalent potentials. Moreover, solutions in the landscape will skip to other solutions at each second because of quantum effect. As a result, the effective potential  $V$ , will be a *highly random function*. That is why random function comes to the picture. The idea of inflation in a landscape, is called the *random inflation*, has studied in many papers (see [30] for example).

When we come to multi-stream understanding of inflation, we must consider the high-dimensional random functions. However, because people cannot generate high-dimensional random functions owing to the exponential disaster as below (for example, [11] only considers the two-dimensional case, while [12] reduces the question into two-dimensional case in order to prove the non-uniqueness of solutions in the high-dimensional landscape), high-dimensional field space is almost impossible to study. However, with the ANN strategy as before we can solve the equation directly. We will show the details in the next subsection.

## 5.2 Monte-Carlo Simulation

Let us come to the algorithm we use. As the last subsection tells, we use the difference strategy

$$\frac{\varphi_i(n+1) + \varphi_i(n-1) - 2\varphi_i(n)}{2\Delta t^2} + 3\sqrt{\frac{V(\varphi_j(n))}{3}}\frac{\varphi_i(n+1) - \varphi_i(n)}{\Delta t} + \partial_i V(\varphi_j(n)) = \frac{3H^{5/2}}{2\pi}\eta_i$$

where  $\eta_i(n) \sim N(0, 1/\sqrt{\Delta t})$ , and Gaussian distributions are independent for different  $i$ . The slow-roll potential is given by

$$V = V_0(\varphi_1) + V_{\text{rand}}(\varphi_1, \varphi_2, \dots, \varphi_d)$$

$$V = V_0(\varphi_1)$$

where  $V_0$  is chosen to be the quadratic potential which dominates in the inflation process as *large field inflation*

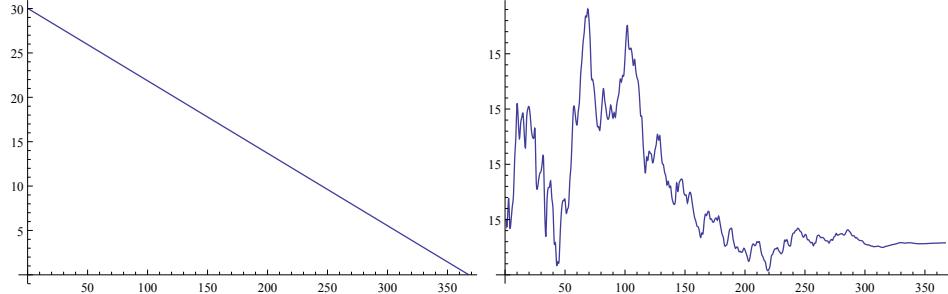
$$V_0(\varphi_1) = \frac{1}{2}m^2\varphi_1^2$$

$$V_0(\varphi_1) = \frac{1}{2}m^2\varphi_1^2$$

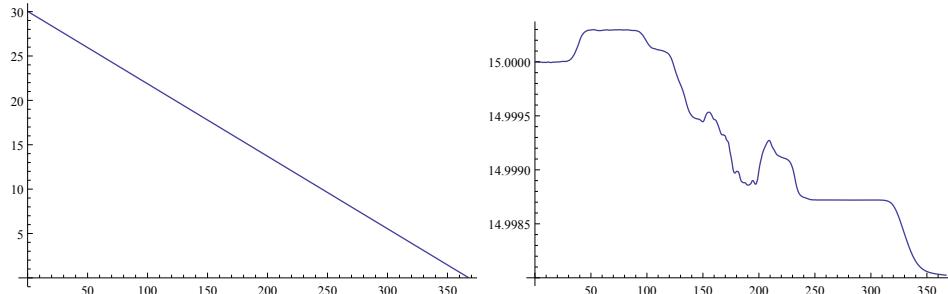
The random part  $V_{\text{rand}}$ , is generated by ANN before our MC simulation with amplitude  $\mathcal{A}$

$$V_{\text{rand}} = \mathcal{A}A_h(\varphi_1, \varphi_2, \dots, \varphi_d)$$

We choose the parameters as  $M_p = 1$  (the Planck mass is normalized),  $d = 20$  (for 20-dimensional field space),  $m = 10^{-6}$ ,  $\Delta\phi = 0.05$  (the feature size for concrete field values),  $\varphi_{\max} = 30$  (the initial value and bounded value of fields),  $\Delta t = 10^5$ ,  $S = 1000$  (the maximal step in the Monte Carlo simulation). For ANN we choose  $\omega = 0.01$ ,  $\alpha = 50$ ,  $\beta = 50$ ,  $h = 1000$ . The initial value of fields are:  $\varphi_1 = \varphi_{\max} - \Delta\varphi$ ,  $\varphi_{i \neq 1} = \varphi_{\max}/2$ . About the randomness, we should assume that the random behavior of  $V$  should be not very small, but should be larger and less-dramatic than quantum fluctuations. In the MC simulation, if field one field drops out of the range  $[0, \varphi_{\max}]$ , we think it as the end of inflation and break the recursion.



**Figure 10.** The MC simulation of field trajectory in dimension 20. We choose the random amplitude  $\mathcal{A} = 10^{-20} \sim 0.01m^3$ , which is the smaller randomness case. The left figure shows evolution of  $\varphi_1$  with step, while the right figure shows evolution of  $\varphi_2$  with step.



**Figure 11.** The MC simulation of field trajectory in dimension 20. We choose the random amplitude  $\mathcal{A} = 10^{-14} \sim 0.01m^2$ , which is the larger randomness case. The left figure shows evolution of  $\varphi_1$  with step, while the right figure shows evolution of  $\varphi_2$  with step.

we can see some expected but interesting features in these figures. First, the behavior of  $\varphi_1$  are linear, namely the characteristic slow-roll inflation with the quadratic potential, which supports that in the random multi-stream inflation, the quadratic potential dominates the whole process of inflation. Second, when the random amplitude  $\mathcal{A}$  are different, things are totally different for sub-leading  $\varphi_{i \neq 1}$ . When  $\mathcal{A} = \mathcal{O}(m^3)$ , the field trajectory is dominated by the quantum fluctuation, whose randomness is larger but the amplitude of change is smaller. When  $\mathcal{A} = \mathcal{O}(m^2)$ , the field trajectory is dominated by the random function, whose randomness is chosen to be smaller but the amplitude of change is larger, as desired.

## 6 Conclusion and Outlook

In this paper we first make a brief review on random function both on theoretical and computational side. An interpolation trial is given to show the random function generation in a two-dimensional surface. Then we introduce the artificial neural network (ANN), a celebrated area in machine learning, and come up a new strategy to generate random functions with the help of ANN. After numerical simulation, we show the strong power of ANN in function generation. Finally, we introduce the basic string inflation theory and present a simple application of ANN function generation in the high-dimensional string landscape. Some work related can be done for future research. First, we only studied the basic shape of inflationary trajectory, and have not discuss anything. One can use this result to study some interesting topic in recent cosmology, such as reheating, bifurcation, stuck, and topological defect in early universe. Second, one can use this method to study other aspect of computational physics, especially the Anderson localization, i.e., in condensed matter physics.

## Acknowledgments

JL will thank Prof. Yi Wang and Dr. Siyi Zhou's discussion in the Hong Kong University of Science and Technology.

## References

- [1] Pugachev, Vladimir Semenovich. *Theory of Random Functions: And Its Application to Control Problems*. Elsevier, 2013.
- [2] Albeverio, S., Hahn, A. and Sengupta. *Rigorous Feynman Path Integrals, with Applications to Quantum Theory, Gauge Fields, And Topological Invariants*. SAMP, 2002.
- [3] Grosche, Christian, and Frank Steiner. *Handbook of Feynman Path Integrals*. Vol. 1. 1998.
- [4] Johnson, Gerald W., and Michel L. Lapidus. *The Feynman Integral And Feynman's Operational Calculus*. Oxford: Clarendon Press, 2000.
- [5] Ito, Kiyosi. "Wiener Integral And Feynman Integral." *Proceedings of the 4th Berke ley Symposium on Mathematical Statistics and Probability*. Vol. 2. 1961.
- [6] Dragovich, Branko. "Adeles in Mathematical Physics." arXiv preprint arXiv:0707.3876 (2007).
- [7] Greenfield, Mark, Matilde Marcolli, and Kevin Teh. "Type III Sigma-spectral Triples And Quantum Statistical Mechanical Systems." arXiv preprint arXiv:1305.5492 (2013).
- [8] Lugan, Pierre, et al. "One-dimensional Anderson Localization in Certain Correlated Random Potentials." *Physical Review A* 80.2 (2009): 023605.
- [9] Arkani-Hamed, Nima, et al. "The String Landscape, Black Holes and Gravity as The Weakest Force." *Journal of High Energy Physics* 2007.06 (2007): 060.
- [10] Huang, Qing-Guo, and S-H. Henry Tye. "The Cosmological Constant Problem and Inflation in The String Landscape." *International Journal of Modern Physics A* 24.10 (2009): 1925-1962.
- [11] Duplessis, Francis, Yi Wang, and Robert Brandenberger. "Multi-Stream Inflation in A Landscape." *Journal of Cosmology and Astroparticle Physics* 2012.04 (2012): 012.
- [12] J. Liu, Y. Wang and S. Zhou, "Non-Uniqueness of Classical Inflationary Trajectories on a High-Dimensional Landscape." arXiv preprint arXiv:1501.06785(2015).
- [13] Marsh, MC David, et al. "Charting An Inflationary Landscape with Random Matrix Theory." *Journal of Cosmology and Astroparticle Physics* 2013.11 (2013): 040.

- [14] Battefeld, Thorsten, and Chirag Modi. "Local random potentials of high differentiability to model the Landscape." arXiv preprint arXiv:1409.5135 (2014).
- [15] Bachlechner, Thomas C. "On Gaussian Random Supergravity." *Journal of High Energy Physics* 2014.4 (2014): 1-33.
- [16] Dreiseitl, Stephan, and Lucila Ohno-Machado. "Logistic Regression And Artificial Neural Network Classification Models: A Methodology Review." *Journal of biomedical informatics* 35.5 (2002): 352-359.
- [17] Yao, Xin. "A Review Of Evolutionary Artificial Neural Networks." *International journal of intelligent systems* 8.4 (1993): 539-567.
- [18] Balazs Csanad Csaji. "Approximation with Artificial Neural Networks." MSc Thesis in Eindhoven University of Technology, The Netherlands.
- [19] Edelman, Alan, and N. Raj Rao. "Random Matrix Theory." *Acta Numerica* 14 (2005): 233-297.
- [20] Baumann, Daniel, and Liam McAllister. "Inflation And String Theory." arXiv preprint arXiv:1404.2601 (2014).
- [21] Li, Miao, and Yi Wang. "Multi-stream Inflation." *Journal of Cosmology and Astroparticle Physics* 2009.07 (2009): 033.
- [22] White, Halbert. "Artificial Neural Networks: Approximation And Learning Theory." Blackwell Publishers, Inc., 1992.
- [23] Kurt Hornik. "Approximation Capabilities Of Multilayer Feedforward Networks." *Neural Networks*, vol. 4, 1991.
- [24] E. Brezin, C. Itzykson, G. Parisi and J. B. Zuber, "Planar Diagrams." *Commun. Math. Phys.*, 35, 1978.
- [25] A. H. Guth, "The Inflationary Universe: A Possible Solution To The Horizon And Flatness Problems." *Phys. Rev. D*, 23, 347, 1981.
- [26] A. A. Starobinsky, "Stochastic De Sitter (inflationary) Stage In The Early Universe." *Lect. Notes Phys.*, 246, 107, 1986.
- [27] G. R. Dvali and S. H. H. Tye, "Brane Inflation." *Phys. Lett. B*, 450, 72, 1999.
- [28] F. Marchesano, G. Shiu and A. M. Uranga, "F-term Axion Monodromy Inflation." *JHEP*, 1409, 184, 2014.
- [29] T. Friedmann and R. P. Stanley, "The String Landscape: On Formulas for Counting Vacua." *Nucl. Phys. B*, 869, 74, 2013
- [30] R. Kallosh and A. D. Linde, "Landscape, the scale of SUSY breaking, and inflation." *JHEP*, 0412, 004, 2004