'Which model is best?' Composite Relative goodness-of-fit testing with kernels

VMFS3

Supervisor : Dr François-Xavier Briol

Department of Statistical Science
University College London

September 17, 2022

Contents

1 Introduction

Broader context of the problem:

There is an old saying by George Box: "All models are wrong, but some are useful". Suppose there is a *perfect model* that can be perfectly described the truth pattern of a given real-world problem, people may never be able to find it since all models have errors (e.g. discriminative error) given the statistical noise in the data. Instead, people always tend to choose a *optimal models* that is not perfect but good enough for the given data and problems (e.g. discriminative problem or generative problem) among a set of candidate models.

The *Model selection* and *Model assessment* are two of the most important steps in statistic and machine learning modeling. Model selection is the process of choosing proper models from some candidate models (by considering a proper level of flexibility, maintainability, and available resource) whereas evaluating or assessing candidate models to choose the best one is known as model assessment (Gareth et al. (2013)).

Model assessment problem is not only being considered in *discriminative model* but also in *generative model* problem. One of the main differences between them is their attitudes to *overfitting* problems. For discriminative modeling problems such as regression (for continuous random variables) or classification (for discrete random variables), while the models are being trained on a training dataset, they need to avoid the problem of overfitting to get a good performance on the testing dataset. However, for the generative modeling problem that aims to train a model which able to fits the given dataset as well as possible, the concern of overfitting is not required. For different types of problems, the techniques of model assessment are different. In this paper, we focus on the model assessment problems that evaluate and compare the generative probabilistic models: **Given two sets of probabilistic generative models (parametric distribution families), determined which one can fit the given dataset better.**

What people currently do in the literature:

As we wish to determined which model fits the dataset better, two types of *hypothesis testing* is being widely used: *goodness-of-fit testing* (GOF) and *two-sample testing* (TST).

The first method of GOF is the Chi-square testing which was proposed by Pearson (1900) in order to determine whether a given model can fit the given dataset or not. This original type of goodness-of-fit can be extended to the *composite goodness-of-fit testing* (C-GOF) case: testing a given set of models instead of a single model. Another type of extension of GOF is known as *relative goodness-of-fit testing* (R-GOF) which aims to test whether a model can fit the given dataset better than another model or not. The combination of them is known as *composite relative goodness-of-fit testing* (CR-GOF).

The classical GOF methods such as the Chi-square test (Pearson (1900)), K-S test (Kolmogorov (1933b)), and C-V test (Cramer (1946)) have a common problem: cannot be applied for the models that only known up to a normalizing constant term, such as the Boltzmann machine (Sutskever et al. (2008)) and other graphical models. To solve this problem, a GOF method with *Kernelized Stein discrepancy* (KSD) have been proposed by Liu et al. (2016) and Chwialkowski et al. (2016). Later, some extensions of this method were proposed, see Fernandez and Gretton (2019), Xu (2021), Schrab et al. (2022), Xu (2022).

With these GOF methods, the researchers are able to test whether a model can fit the given data or not, even if the model is only known up to a normalizing constant term. However, if the candidate's model is a set of models instead of a single model, the C-GOF is needed.

For the C-GOF problem, some methods which are not based on KSD have been proposed for specific parametric families such as the normal distributions family (Betsch and Ebner (2020); Henze and Visagie (2020)) and gamma distribution family (Henze et al. (2012)). Recently, a method of composite goodness-of-fit testing by using the KSD is being proposed by Key et al. (2021). This method combines the KSD-based goodness-of-fit that was proposed in Liu et al. (2016) and the minimum-KSD estimator (Barp et al. (2019)), resulting in an efficient method of C-GOF for the models that contain normalizing constant terms.

Different from GOF (or C-GOF) which aims to test whether a model (or a set of models) can fit the data well or not, the problem of R-GOF is more complicated since it needs to compare two models. Recall the attitude of the generative model is to learn the distribution of the given dataset. Given two models, the better one should be "closer" to the distribution of the data. Here, "closer" means the statistical discrepancy (such as KSD) between the model and data is smaller.

For the R-GOF problem, a method that uses the difference of two KSDs for relative goodness-of-fit testing problem given two latent variable models has been proposed by Kanagawa et al. (2019). In Lim et al. (2019), a method of GOF with KSD has been applied in multiple model comparison (finite model).

When the model is represented by its sample, goodness-of-fit testing reduces to two-sample testing (TST). The two-sample testing aims to determine whether two given samples follow the same distribution or not. The first two-sample test is the *t-test* (Stein (1945)). However, its unrealistic assumptions make it hard to be applied to many real-world problems (e.g. high dimensional problems). To fix this problem, a two-sample test combine with *Maximum Mean discrepancy* (MMD) was proposed by Gretton et al. (2012). Later in Bounliphone et al. (2015), this kind of two-sample test was used to build a method for the relative goodness-of-fit test by using the difference between two MMDs.

However, as Kanagawa et al. (2019) mentioned, there're two main drawbacks of the two-sample testing-based methods: 1. the need for sampling from the model makes them computationally expensive. 2. They don't take the advantage of the information involved in the models such as the dependencies between variables.

Issues with existing methods:

To sum up, our goal is to determine which set of models can fit the given dataset better, given two sets of probabilistic generative models. This problem is known as the *composite relative testing problem* (C-R testing problem). The classical goodness-of-fit testing (GOF) cannot be applied for the model that takes the form of unnormalized and differentiable density functions, while the two-sample testing (TST) based methods own their drawback in the sampling procedure. Also, as we mentioned above, the existing C-GOF and R-GOF methods do not perfectly match the C-R testing problem.

To date, this problem remains unresolved. This paper proposes a novel goodness-of-fit testing method with KSD to solve the above C-R testing problem, named: "KSD-based composite relative goodness-of-fit testing", in short: KCR-GOF.

KCR-GOF: A novel goodness-of-fit test that achieve SOTA performance:

Our KCR-GOF is motivated by the KSD-based method for C-GOF (Key et al. (2021)) and the KSD-based R-GOF method for comparing two latent variable models (Kanagawa et al. (2019)).

Recall our goal is to determine which set of models can fit the given dataset better, given two sets of parametric generative models. The KCR-GOF is a *two-stage testing* that included

estimation and **testing**. The estimation procedure can be regarded as selecting a model from a set of parametric models. Usually, the classical statistical inference method such as Maximum likelihood inference is used. However, it's hard to compute the likelihood when the models are only known up to a normalization constant term. Hence the minimum-KSD estimator is being applied. The basic idea of it is select a model from a set of models by minimizing the KSD between model and dataset. By using this method, we selected two models from two parametric distribution sets respectively. For the testing part, we use the difference between the two KSDs to build a test statistic.

Here is a figure to provide an intuitive understanding of our method. The formal description of our method will be left to Section 3.

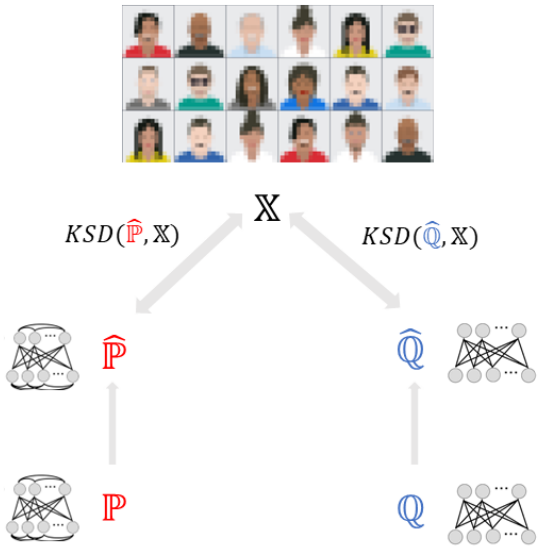

Figure 1: $\mathbb{X}$ is the population diatribution of the given dataset (images of human face). $\mathbb{P}$ and $\mathbb{Q}$ represent two sets of candidate parametric models $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$ and $\{\mathbb{Q}_\phi\}_{\phi\in\Phi}$ respectively (Θ and Φ are the paramter space). We wish to determined whether $\mathbb{P}$ or $\mathbb{Q}$ can fit the dataset better. The first step is select two optimal models $\hat{\mathbb{P}}, \hat{\mathbb{Q}}$ from two model sets $\mathbb{P}$ and $\mathbb{Q}$ by minimum the Kernelized Stein discrpancy (KSD) between the model and dataset. Then we use the difference between $KSD(\hat{\mathbb{P}}, \mathbb{X})$ and $KSD(\hat{\mathbb{Q}}, \mathbb{X})$ to build a hypothesis test. If $KSD(\hat{\mathbb{P}}, \mathbb{X}) < KSD(\hat{\mathbb{Q}}, \mathbb{X})$ then $\mathbb{P}$ fits the dataset better than $\mathbb{Q}$. Otherwise, $\mathbb{Q}$ fits better.

Our contributions: We did a literature review of the goodness-of-fit testing and find that there do not exist a proper method for the relative composite goodness-of-fit problem. Then we proposed a novel method to solve this problem, known as "KSD-based composite relative goodness-of-fit testing" (KCR-GOF). Also, we provide a efficient and flexible code for our method.

Structure of this paper: In section 2, all the the background knowledge you may need to know (e.g. Kernel method, Stein's method, KSD ect) will be fully introduced. In section 3, the formal description of KCR-GOF as well as the related discussion will be introduced. In section 4, two experiment and related discussion are given. In the Appendix, some other background material and proof is given.

2 BACKGROUND MATERIAL

This section introduces all the related background knowledge for our project. Recall the KCR-GOF that we mentioned before is a two-stage testing (include etimation and testing).

For estimation part, the procedure of using statistical discrepancy to measure the similarity between two distribution is neccessary. In this paper, we choose the Kernelized Stein discrepancy (KSD) as statistical discrepancy since it perfectly fit the model that only known up to a normalizing constant term. The KSD owe its name from Kernel method (**section 2.1**) and Stein's method **section 2.3**. Also, the idea of KSD is mapping the data into a *Reproducing Kernel Hilbert space* (RKHS) (**section 2.2**) in order to measure the similarity of two distribution in a high dimensional space. The detailed of KSD will be given in **section 2.4**. Finally, a brief introduction of minimum-KSD estimator is shown in **section 2.7**.

For the testing part, the background material of goodness-of-fit test (**section 2.5**) and relative goodness-of-fit test (**section 2.6**) are provided.

2.1 Kernel method

The Kernel method owes its name to the use of *kernel functions*, which enable people to obtain the result of a high dimensional inner product without explicit computation. With a proper choice of the kernel function, we're able to map the data into a high dimensional space (known as Reproducing kernel Hilbert space) and then do the goodness-of-fit test. This idea benefits from the advantage that measuring the similarity of two distributions by using the high dimensional features is better than just using the low dimensional features such as mean and variance.

The Kernel method is widely used in statistics and machine learning. For many learning algorithms such as perceptron [Rosenblatt (1958)] and support vector machines (SVMs) [Cortes and Vapnik (1995)], the data only being employed by pair (such as $\mathbf{x},\mathbf{x}' \in \mathcal{X}$ where $\mathcal{X}$ is a non-empty set) with inner product $\langle \mathbf{x},\mathbf{x}' \rangle$. It can be regards as the similarity measure between $\mathbf{x}$ and $\mathbf{x}'$ [Muandet et al. (2020)]. For these algorithms, we can replace the inner product $\langle \mathbf{x},\mathbf{x}' \rangle$ with a kernel function $k(\mathbf{x},\mathbf{x}')$. For example, we can set kernel function to simplest the one: $k(\mathbf{x},\mathbf{x}') = \langle \mathbf{x},\mathbf{x}' \rangle^1$.

However, for some problems known as *linearly inseparable problem*, data are not separable in low dimensional space. For this reason, a kernel function is needed. The kernel function firstly maps the data into a higher dimensional space where the data are linearly separable then applied the inner product. That is, the kernel function should involved a mapping function ϕ such that $k(\mathbf{x},\mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$. The mapping function ϕ is defined as below:

Definition 2.1.1(feature map). A feature map $\phi(\mathbf{x})$ is a function that map a data point $\mathbf{x}$ to a feature space $\mathcal{F}$: $\mathbf{x} \mapsto \phi(\mathbf{x})$ where $\mathbf{x} \in \mathcal{X}, \phi(\mathbf{x}) \in \mathcal{F}$. Here the $\mathcal{X}$ is the data space(or input space) and $\mathcal{F}$ is the feature space.

For many cases, the input data $\mathbf{x}$ could be a finite d-dimensional vector $\mathbf{x} \in \mathcal{X}^d$. However, we need not make assumptions other than $\mathcal{X}$ being a non-empty set. For example, we could consider a set of discrete objects, such as strings. [Schölkopf et al. (2002)]

Note that for some mapping function ϕ, the $\phi(\mathbf{x})$ could be a finite vector with a higher dimension. However, for some commonly used mapping such as the mapping involved in a gaussian kernel, the $\phi(\mathbf{x})$ is an infinite vector that can be viewed as a function $k(\mathbf{x}, \cdot)$. The idea of viewing the infinite vector as function may not intuitive at first, but after we give an introdction

to kernel function it will become more clear and you will know the fact that people are more likely to choose a specific kernel function rather than choose a specific mapping function during modeling. Before we define what is a kernel function, we first introduce a useful operation in mathematics: Inner Product.

Definition 2.1.2 (Inner product). Let $\mathcal{V}$ be a vector space over $\mathbb{R}$. A function $\langle \cdot, \cdot \rangle_{\mathcal{V}} : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ is an inner product on $\mathcal{V}$ if it's satisfied three properties:

1.Linear: $\langle \alpha_1 f_1 + \alpha_2 f_2, g \rangle = \alpha_1 \langle f_1, g \rangle_{\mathcal{V}} + \langle f_2, g \rangle_{\mathcal{V}}$.

2.Symmetric: $\langle f, g \rangle_{\mathcal{V}} = \langle g, f \rangle_{\mathcal{V}}$.

3.$\langle f, f \rangle_{\mathcal{V}} \geqslant 0$ and $\langle f, f \rangle_{\mathcal{V}} = 0$ if and only if $f = 0$.

From the above definition, the inner product can be viewed as a function that map two vectors in $\mathcal{V}$ into a scalar in $\mathbb{R}$. The inner product is important for the kernel method since all the kernel functions are being defined on the *Hilbert spcae* which is a space on which an inner product is defined. With the inner product, we can give the definition of Hilbert space as below:

Definition 2.1.3 (Hilbert space). Let $\langle \cdot, \cdot \rangle : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ be an inner product on a real or complex vector space $\mathcal{H}$. Also, let $|| \cdot || : \mathcal{H} \to \mathbb{R}$ be the associated norm defined as $||v|| = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}$ where $\mathbf{v} \in \mathcal{H}$. Then the vector space $\mathcal{H}$ is a Hilbert space if it is complete with respect to this norm.

Intuitively, if the l^2- norm of a vector is not an infinite number, then this vector should belong to a Hilbert space $\mathcal{H}$. A typical example is a Cauchy sequence in $\mathcal{V}$ which has a finite norm.

With the deifinition of Hilbert space $\mathcal{H}$, we can now define the kernel function k by using feature map ϕ and inner product on Hilbert space: $\langle \cdot, \cdot \rangle_{\mathcal{H}}$.

Definition 2.1.4 (kernel function). Given a Hilbert space $\mathcal{H}$ and a non-empty set $\mathcal{X}$ as well as a map function $\phi(x) : \mathcal{X} \to \mathcal{H}$. Suppose $\mathbf{x}$, $\mathbf{x}' \in \mathcal{X}$, then a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel function that

$$k(\mathbf{x},\mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}} \tag{2.1}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the inner product of $\mathcal{H}$.

From the definition of kernel function we can see that, in order to compute the result of a kernel function, we may need to know the feature map $\phi(x)$. So you may wonder how can we choose a proper feature mapping. There's some commonly used feature maps such as *polynomial feature map*. Here is an example in 2-dimension: $\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_2x_2)$ where $\mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^2$. In this case:

$$\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}} = x_1^2 x_1'^2 + x_2^2 x_2'^2 + 2x_1x_2x_1'x_2' = (x_1x_1' + x_2x_2')^2 = \left\langle \mathbf{x}, \mathbf{x}' \right\rangle_{\mathcal{H}}^2 \tag{2.2}$$

In order to build a more complex and flexible model, people tend to choose the kernel function $k(\mathbf{x},\mathbf{x}')$ rather than choose the feature map ϕ because the dimension of ϕ could be very large or even infinite for some problems. In that cases, it's impossible to compute the inner product of $\phi(\mathbf{x})$ and $\phi(\mathbf{x}')$ directly. However, if we compute the inner product by choosing a proper kernel function, we can bypass this problem.

There's some commonly used kernels such as Gaussian and Laplace kernels:

$$k(\mathbf{x},\mathbf{x}') = e^{-\frac{||\mathbf{x} - \mathbf{x}'||_2^2}{2\sigma^2}}, k(\mathbf{x},\mathbf{x}') = e^{-\frac{||\mathbf{x} - \mathbf{x}'||}{\sigma_1}} \tag{2.3}$$

However, after we choose or design a kernel function k, how can we guarantee there is a feature map ϕ involved in k? In other words, how can we guaranteed computing the kernel function $k(\mathbf{x}, \mathbf{x}')$ is equivalent to computing a inner product with a mapping function: $\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}}$? To answer this question, we firstly introduce the concept of positive definite:

Definition 2.1.5 (positive definite function). A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive definite if $\forall n \geqslant 1, \forall (a_1, ..., a_n) \in \mathbb{R}^n, \forall (x_1, ..., x_n) \in \mathcal{X}^n$,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(x_i, x_j) \geqslant 0 \tag{2.4}$$

The function $k(\cdot, \cdot)$ is strictly positive definite if for mutually distinct x_i, the equality holds only when all the a_i are zero.

Then we have the following lemma that solves the above question:

Lemma 2.2.1 Let $\mathcal{H}$ be any Hilbert space, $\mathcal{X}$ a non-empty set and $\phi : \mathcal{X} \to \mathcal{H}$. Then a kernel function $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$ is a positive definite function. And the reverse direction also holds.

The proof of Lemma 2.2.1 is shown in Appendix A. With Definition 2.1.5 and Lemma 2.1.1, we know that if the kernel function k is valid, then k should be positive definite so that there always exists $\phi : \mathcal{X} \to \mathcal{H}$ for which $k(\mathbf{x},\mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}}$.

More detail for kernel choice is left to Section 4 and Appendix C.

2.2 Reproducing Kernel Hilbert spaces

Recall that the KSD is used in composite relative goodness-of-fit (CR-GOF) testing. The *Reproducing Kernel Hilbert space (RKHS)* is one of the most fundamental concepts for KSD. In section 2.3 you will see that the main idea of the Stein method is to apply a *Stein class* onto a proper *Stein class*. For KSD, the stein class is a unit-ball in RKHS.

As we have introduced the concept and notation of feature map ϕ and kernel function k on feature spaces $\mathcal{H}$. We are now in a proper position to give the introduction of RKHS. In a sentence, RKHS is a Hilbert space that contain a function with special property known as *reproducing property*.

To elaborate the conpect of RKHS, we firstly recall the definition of Hilbert space that we mentioned before: Suppose $\langle \cdot, \cdot \rangle : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ is a inner product on $\mathcal{V}$ where $\mathcal{V}$ is a real or complex vector space. A Hilbert space $\mathcal{H}$ is defined on $\mathcal{V}$ with the associated norm defined as $||v|| = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}$ where $\mathbf{v} \in \mathcal{V}$.

According to Lemma 2.2.1, if we have a positive definite function k, there always exists one (or more) feature map ϕ for which the kernel defines the inner product. Before the formal definition of RKHS is given, here is an example of how RKHS is applied in the XOR problem.

Example 2.2 (XOR problem in RKHS) For the XOR problem, the data is linearly inseparable in $\mathbb{R}^2$ so that we need to map them into a higher dimension space (e.g. $\mathbb{R}^3$) where the data are linearly separable. Suppose we use a feature map ϕ:

$$\phi(\mathbf{x}) = (x_1, x_2, \sqrt{2}x_1 x_2)^T \in \mathbb{R}^3, \text{where } \mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^2 \tag{2.5}$$

result in a kernel function:

$$k(x,y) = \begin{bmatrix} x_1 \\ x_2 \\ \sqrt{2}x_1x_2 \end{bmatrix}^T \begin{bmatrix} y_1 \\ y_2 \\ \sqrt{2}y_1y_2 \end{bmatrix} = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}} \in \mathbb{R}^1 \tag{2.6}$$

Let's define a function $f : \mathbf{x} \in \mathbb{R}^2 \to f(\mathbf{x}) \in \mathbb{R}^1$ with the feature $(x_1, x_2, \sqrt{2}x_1x_2)^T \in \mathbb{R}^3$ where $\mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^2$ as below:

$$f(x) = ax_1 + bx_2 + c\sqrt{2}x_1x_2$$

The function f is an element of a space of functions $\mathcal{F}$ that every elements $f \in \mathcal{F}$ mapping $\mathcal{X} = \mathbb{R}^2$ to $\mathbb{R}^1$ since different value of a, b, c result in different functions f. For example: $f_1(x) = a_1x_1 + b_1x_2 + c_1x_1x_2$, $f_2(x) = a_2x_1 + b_2x_2 + c_2x_1x_2$, where $a_1 \neq a_2, b_1 \neq b_2, c_1 \neq c_2$.

Then we can define an equivalent representation for f which just using its coefficient:

$$f(\cdot) = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

The function space where all function $f(\cdot) = (a, b, c)^T$ belong to is a Reproducing Kernel Hilbert Spaces (RKHS), denote as $\mathcal{F}$. Then it's interesting to note that for all the functions $f \in \mathcal{F}$ we have a property below, known as the "reproducing property" :

$$f(x) = \left\langle (a, b, c)^T, (x_1, x_2, \sqrt{2}x_1x_2) \right\rangle_{\mathcal{H}} = \langle f(\cdot), \phi(x) \rangle_{\mathcal{H}} \tag{2.7}$$

In equation(2.6) we have: $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$. Then we have $k(x, y) = \langle k(\cdot, y), k(x, \cdot) \rangle_{\mathcal{H}}$ if we denote $\phi(x) = k(\cdot, y), \phi(y) = k(x, \cdot)$. Then the equation (2.7) can be equally rewritted into $f(x) = \langle f(\cdot), k(x, \cdot) \rangle_{\mathcal{H}}$. This property is known as reproducing property and we say a Hilbert space $\mathcal{F}$ where all the element $f(\cdot) \in \mathcal{F}$ has this property is a Reproducing Kernel Hilbert space.

We sometimes write f rather than $f(\cdot)$, when there is no ambiguity. Now we can give a formal definition of RKHS as below:

Definition 2.2.2 (reproducing kernel Hilbert space (RKHS)). A Hilbert space $\mathcal{H}$ of functions is a reproducing kernel Hilbert space (RKHS) if $1.\forall x \in \mathcal{X}, k(\cdot, x) \in \mathcal{H}, 2.\forall x \in \mathcal{X}, \forall f \in \mathcal{H}, \langle f(\cdot), k(\cdot, x) \rangle_{\mathcal{H}} = f(x)$.

Intuitively, given $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), ..., \phi_d(\mathbf{x}))^T$, we can regard the function $f \in \mathcal{F}$ where $\mathcal{F}$ is a RKHS as a d-dimensional real-number vector: $f(\cdot) = (f_1, f_2, ..., f_d)^T$. When f is evaluated on a specific point $\mathbf{x}$ (which is a vector or a scalar) we have: $f(\mathbf{x}) = f(\cdot)^T\phi(\mathbf{x}) = \langle f(\cdot), \phi(\mathbf{x}) \rangle_{\mathcal{H}}$.

2.3 Stein's method

In this section, we give an introduction to *Stein's method* that is being used in our composite relative goodness-of-fit problem. Stein's method is the foundation of the Kernel Stein Discrepancy (KSD) that we will introduce in the next section.

The key idea of Stein's method is using a class of linear operators called *Stein's operator* to characterizes a distribution, providing an elegant way for comparing distribution. It was firstly introduced by Charles Stein [Stein (1972)] for comparisons to the normal distribution, then extended to Poisson approximation and other approximation problem. Stein's method also provides a powerful solution for deriving explicit bounds on distributional distances for the model which contains dependence structures [Anastasiou et al. (2021)]. The technique of combining Stein's identity with RKHS was first developed by [Oates et al. (2014)] for variance reduction. Among all the applications of Stein's method, the one which closest connects with our composite relative goodness-of-fit testing is the application in goodness-of-fit testing [Liu et al. (2016); Chwialkowski et al. (2016); Key et al. (2021)]. For more applications of Stein's method, see Anastasiou et al. (2021). In this report, we only focus on:

1. How does Stein's operator help us to bypass the computation of intractable likelihoods?
2. How to build a statistical discrepancy with Stein's operator.

In this section, we answer the first question while the second one is discussed in the next section.

As we mentioned in the introduction part, the first stage of the composite relative goodness-of-fit test is parameter estimation. The procedure of parameter estimation is usually involved the calculation of likelihoods. However, the mordern learning techniques increasingly involve complex probabilistic models with computationally intractable likelihoods such as deep generative models [Kingma and Welling (2013); Goodfellow et al. (2014)] and model criticism [Kim et al. (2016)]. Although Markov chain Monte Carlo(MCMC) or variational methods such as Variational Auto-encoder(VAE) [Kingma and Welling (2013)] can be used to approximate the likelihood, the approximation error is often large and hard to control.

Around 2015, the stein method attracted attention from researchers of machine learning and computational statistics. It thrives since it allows people to handle the model that only known up to a normalizing constant term. In detailed, a typical form of probability model can be written as: $p(x) = f(x)/Z$ with $Z = \int_{x\in\mathcal{X}} f(x)dx$ being the normalizing constant term. This type of model is usually shown in the graphical model and one famous example is the Boltzmann machine [Sutskever et al. (2008)]. In practice, the computation of Z may be difficult when there involves high dimensional integration. Stein's method is the key to bypassing this problem.

To elaborate on how Stein's operator work in this problem, we start from the concept of *Score function*. Here we fix some notations. $\mathcal{X}$ is a non-empty subset of $\mathbb{R}^d$ and has a continuous probability differentiable density $p(x)$. Then the (Stein) *Score function* of p can be defined as follow:

Definition 2.4.1(Score function) Assume that $\mathcal{X}$ is a non-empty subset of $\mathbb{R}^d$ and $p(x)$ is a smooth density of $\mathcal{X}$, the (Stein) Score function of p is defined as

$$s_p = \bigtriangledown_x \log p(x) = \frac{\bigtriangledown_x p(x)}{p(x)} \tag{2.8}$$

In statistics, the score function $s(\theta) = \bigtriangledown_\theta \log \mathcal{L}(\theta)$ is the gradient of the log-likelihood function. Here, the score function can be used to define the *Langevin Stein operator* of p which is a kind of Stein's operator. Before we give a definition of the Stein operator, we introduce the *Stein class* of p as follows:

Definition 2.4.2(Stein class) A function $f : \mathcal{X} \to \mathbb{R}$ is in the Stein class $\mathcal{F}$ of p if f is smooth and satisfies

$$\int_{x\in\mathcal{X}} \bigtriangledown_x (f(x)p(x))dx = 0 \tag{2.9}$$

From the above definition, the Stein class $\mathcal{F}$ of probability density p can be viewed as a collection of the function f which satisfy the above property. You may wonder why we need f to satisfied the above property. Before we explain the reason for that, we firstly give a definition of the stein operator, and a popular choice of it, known as *Langevin Stein operator*.

Definition 2.4.3 (Stein operator) Given a target probability distribution $\mathbb{P}$ on some set $\mathcal{X}$ and its stein class $\mathcal{F}$, suppose $\mathcal{A}_p$ is a linear operator acting on any function $f \in \mathcal{F}$. We call $\mathcal{A}_p$ is a stein operator of P if the below equation is held:

$$\mathbb{E}_{X\sim p}[\mathcal{A}_p f(X)] = 0 \tag{2.10}$$

The most popular choice of Stein operator is the Langevin Stein operator. For this kind of stein operator, the information of target distribution $\mathbb{P}$ only contain in the score function s_p that we defined in equation (2.8).

Definition 2.4.4 (Langevin Stein operator) The Stein's operator of p is a linear operator acting on the function f where $f \in \mathcal{F}$ and $\mathcal{F}$ is the Stein class of p. Suppose s_p is the score function of p, the stein operator on f can be defined as:

$$\mathcal{A}_p f(x) = s_p(x)f(x) + \bigtriangledown_x f(x) \tag{2.11}$$

For the above equation, the score function s_p and the stein operator $\mathcal{A}_p f$ can be viewed as a function that map from $\mathcal{X}$ to $\mathbb{R}^d$. And the function $f(x) = [f_1(x), ..., f_d(x)]$ is a $d-$dimensional *vector-valued function* which in $\mathcal{F}$ the Stein class of $\mathbb{P}$. Note that all $f_i, \forall i \in [d]$ is in the Stein class of $\mathbb{P}$. As the derivative of the function $f(x)$ is a $d \times d$ matrix, the stein operator can be rewrite as $\mathcal{A}_p f(x) = s_p(x)f(x)^T + \bigtriangledown_x f(x)$ which is also a $d \times d$ matrix-value function. Here the "vector-valued function" means given a input x, it will return a real-value vector while "matrix-valued function" will return a real-value matrix.

After the Langevin Stein operator of p is defined, now it's the best time to explain why the function $f \in \mathcal{F}$ needs to satisfy the property that we mentioned before. The Lemma 2.4.1 show that we can rewrite Stein's operator into another form:

Lemma 2.4.1 Suppose $p(x)$ is a smooth density support by $\mathcal{X}$, $s_p(x)$ is the score function of p, The Stein's operator $\mathcal{A}_p f(x) = s_p(x)f(x) + \bigtriangledown_x f(x)$ can be rewrited into:

$$\mathcal{A}_p f(x) = \frac{1}{p(x)}\frac{d}{dx}(f(x)p(x)) \tag{2.12}$$

Proof. $\mathcal{A}_p f(x) = \frac{1}{p(x)}\frac{d}{dx}(f(x)p(x)) = \frac{1}{p(x)}(p(x)\frac{1}{dx} + f(x)\frac{d}{dx}p(x)) = \frac{d}{dx}f(x) + f(x)\frac{1}{p(x)}\frac{d}{dx}p(x) = \frac{d}{dx}f(x) + f(x)\frac{d}{dx}\log p(x) = \mathcal{A}_p f(x) = s_p(x)f(x) + \bigtriangledown_x f(x)$

After we rewrite the Stein operator into the new form, let's take the expectation of Stein's operator $\mathcal{A}_p f(x)$ under p:

$$\begin{aligned}\mathbb{E}_p[\mathcal{A}_p f(x)] &= \int [\frac{1}{p(x)}\frac{d}{dx}(f(x)p(x))]p(x)dx \\ &= \int [\frac{d}{dx}(f(x)p(x))]dx \\ &= 0\end{aligned} \tag{2.13}$$

Note that for the equation (2.13), $\int [\frac{d}{dx}(f(x)p(x))]dx = 0$ only holds when:

$$\lim_{||x||\to\infty} f(x)p(x) = 0 \tag{2.14}$$

As Liu et al. (2016) mentioned, the above condition can be checked using divergence theorem or integration by parts.

From equation (2.13) you can see that, when the function $f \in \mathcal{F}$ satisfied the property that we mentioned before, then the expectation of Stein's operator under p is zero. In this case, the *Stein's Identity* can be defined as below:

Definition 2.4.4 (Stein's Identity) Assume $p(x)$ is a smooth density support by $\mathcal{X}$, then the Stein's identity is defined as below for any function $f \in \mathcal{F}$ where $\mathcal{F}$ the Stein class of p:

$$\mathbb{E}_p[\mathcal{A}_p f(x)] = \mathbb{E}_p[s_p(x)f(x)^T + \bigtriangledown f(x)] = 0 \tag{2.15}$$

Intuitively, the above equation reveals that if we take the expectation for Stein's operator $\mathcal{A}_p f(x)$ under p then the result is zero. For the target distribution $\mathbb{P}$, if we take the expectation under other distribution such as q rather than the same distribution p, the result (Ley & Swan(2013)) given below can be viewed as a measure of similarity between p and q which a convenient tool for the derivation of Kernelized Stein discrepancy(KSD).

Lemma 2.4.1 (Ley and Swan (2013)). Assume $p(x), q(x)$ are two smooth density support on $\mathcal{X}$ and $f(x)$ is the function that on the Stein class of p. Then

$$\mathbb{E}_p[\mathcal{A}_q f(x)] = \mathbb{E}_p[(s_q(x) - s_p(x))f(x)^T] \tag{2.16}$$

Proof. From Definition 2.4.4 we have $\mathbb{E}_p[\mathcal{A}_p f(x)] = 0$. Hence $\mathbb{E}_p[\mathcal{A}_q f(x)] = \mathbb{E}_p[\mathcal{A}_q f(x) - \mathcal{A}_p f(x)] = \mathbb{E}_p[(s_q(x)f(x)^T + \bigtriangledown f(x)) - (s_p(x)f(x)^T + \bigtriangledown f(x))] = \mathbb{E}_p[(s_q(x) - s_p(x))f(x)^T]$.

The above result is important since the later Kernelized Stein discrepancy (KSD) can be constructed by using the Stein discrepancy in this form. Intuitively, the expectation of the Stein's operator to q under p is the $f(x)$-weighted expectation of the score function difference $(s_q(x) - s_p(x))$ [Liu et al. (2016)]. If the two smooth densities $p(x)$ and $q(x)$ supported on $\mathbb{R}$ are identical then $\mathbb{E}_p[\mathcal{A}_q f(x)]$ will turn to $\mathbb{E}_p[\mathcal{A}_p f(x)]$ which equal to zero. Note that the score function is a $d \times 1$ vector-valued function while $f(x)^T$ is a $1 \times d$ function, the $\mathbb{E}_p[\mathcal{A}_q f(x)]$ would be a $d \times d$ matrix. Hence we can take its trace to give a scalar that represent the similarity of distribution p and q:

$$\mathbb{E}_p[\text{trace}(\mathcal{A}_q f(x))] = \mathbb{E}_p[(s_q(x) - s_p(x))^T f(x)] \tag{2.17}$$

From Lemma 2.4.1 we know that, with the Langevin Stein operator $\mathcal{A}_p f(x) = s_p(x)f(x) + \bigtriangledown_x f(x)$, the expression of $\mathbb{E}_p[\mathcal{A}_q f(x)]$ can be compute by the the difference of the score $s_q(x) - s_p(x)$ with $f(x)$ (the expression of $f(x)$ is known). With socre function $s_p = \bigtriangledown_x \log p(x) = \frac{\bigtriangledown_x p(x)}{p(x)}$, suppose $p(x) = f(x)/Z$ where $Z = \int_{x\in\mathcal{X}} f(x)dx$, then we have

$$s_p(x) = \bigtriangledown_x \log p(x) = \bigtriangledown_x \log \frac{f(x)}{\int_{x\in\mathcal{X}} f(x)dx} = \bigtriangledown_x \log f(x) \tag{2.18}$$

From equation (2.16) we know that when the model $p(x)$ is only known up to a normalizing constant term, the value of score function $s_p(x)$ can be obtained by $\bigtriangledown_x \log f(x)$ where the normalizing constant term is disappears. That's the reason why Stein's operator helps us to bypass the computation of intractable likelihoods.

2.4 Kernel Stein Discrepancy

In this section, we answer the second question that we mentioned in section 2.3: How to build a *statistical discrepancy* given a Stein's operator. First of all, we need to know what is a statistical discrepancy. Here we give the definition of a discrepancy as below:

Definition 2.4.1 (Statistical discrepancy) Suppose a discrepancy between two distributions $\mathbb{P}$ and $\mathbb{Q}$ is $m(\mathbb{P}, \mathbb{Q})$. Then proper discrepancy measure function m should be follow two conditions: (1) $m(\mathbb{P}, \mathbb{Q}) = 0 \Leftrightarrow \mathbb{P} \equiv \mathbb{Q}$. (2) $m(\mathbb{P}, \mathbb{Q}) \geqslant 0$ increases as the difference between $\mathbb{P}$ and $\mathbb{Q}$ increases. Note that definition of statistical discrepancy different from the definition of *Metric* (from mathematics) which requires $m(\mathbb{P}, \mathbb{Q}) = m(\mathbb{Q}, \mathbb{P})$. For some kinds of statistical discrepancy (e.g. K-L divergence), we have $m(\mathbb{P}, \mathbb{Q}) \neq m(\mathbb{Q}, \mathbb{P})$

Kernelized Stein Discrepancy (KSD) is a stein discrepancy with the kernel. Assume $\mathcal{X}$ is the data space and $\mathcal{P}(\mathcal{X})$ is the set of all Borel distributions on $\mathcal{X}$. For simplicity, we will focus on $\mathcal{X} = \mathbb{R}^d$. The KSD is a function $\mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X}) \to \mathbb{R}^+$ which measure the similarity between two distribution $\mathbb{P}, \mathbb{Q} \in \mathcal{P}(\mathcal{X})$. The KSD is closely connects to *Integral Pseudo-probability Metrics (IPMs)* [Müller (1997)].

$$\mathcal{D}_{\mathcal{F}}(\mathbb{P}, \mathbb{Q}) = \sup_{f \in \mathcal{F}} |\mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)]| \tag{2.19}$$

It's well known that different choices of f result in different statistical discrepancy. Another popular discrepancy is the Maximum Mean discrepancy(MMD) whose f is choosen from a unit-ball of some RKHS associated with a kernel. For more information on IPM, see Anastasiou et al. (2021).

In this section, we first give the introduction of Kernelized Stein discrepancy(KSD). Following the definition of stein identity and IPM, the definition of *Stein discrepancy* is given below:

Definition 2.4.5 (Stein's discrepancy) $\mathbb{P}$ is a target distribution support on a non-empty set $\mathcal{X}$, suppose $\mathcal{F}$ is the stein class of it and $\mathcal{A}_p$ is the stein operator acting on the stein class of $\mathbb{P}$. Then the stein discrepancy between $\mathbb{P}$ and another distribution $\mathbb{Q}$ can is given below, with appropriate norm $||\cdot||^*$:

$$S(\mathbb{P}, \mathbb{Q}) = \sup_{f \in \mathcal{F}} ||\mathbb{E}_{X \sim Q}[\mathcal{A}_p f(X)] - \mathbb{E}_{X \sim Q}[\mathcal{A}_q f(X)]||^* = \sup_{f \in \mathcal{F}} ||\mathbb{E}_q[\mathcal{A}_p f(X)]||^* \tag{2.20}$$

With properly choice of norm and $\mathcal{F}$ (sufficiently large), we have $S(\mathbb{P}, \mathbb{Q})$ when $\mathbb{P} \neq \mathbb{Q}$. And $S(\mathbb{P}, \mathbb{Q}) = 0$ when $\mathbb{P} = \mathbb{Q}$. Then $S(\mathbb{P}, \mathbb{Q})$ is a proper discrepancy to measure the dissimilarity between $\mathbb{P}$ and $\mathbb{Q}$.

The **Kernelized Stein discrepancy(KSD)** $\mathbb{S}(\mathbb{P}, \mathbb{Q})$ between two probability distribution (or so-called "models") $\mathbb{P}$ and $\mathbb{Q}$ can be defined as follow:

Definition 2.5.1(Langevin Kernel Stein Discrepancy (KSD)) Suppose the smooth probability density function of $\mathbb{P}$ and $\mathbb{Q}$ is denoted by p and q where p and q are support on $\mathcal{X}$. Then $\mathcal{A}_p f(x)$ is the Stein's operator of p where $f \in \mathcal{F}$ and $\mathcal{F}$ is the stein class of p. Then the Kernel Stein Discrepancy (KSD) can be defined by the square of Stein discrepancy:

$$\begin{aligned} \mathbb{S}(\mathbb{P}, \mathbb{Q}) &= \sup_{f \in \mathcal{F}} (\mathbb{E}_{X \sim Q}[\mathcal{A}_p f(X)] - \mathbb{E}_{X \sim Q}[\mathcal{A}_p f(X)])^2 \\ &= \sup_{f \in \mathcal{F}} (\mathbb{E}_{X \sim Q}[\mathcal{A}_p f(X)])^2 \end{aligned} \tag{2.21}$$

To compute the above equation, we need to introduce the kernel form of KSD. Recall the definition 2.4.4, given the probability density p and its Stein class f, the Stein operator $\mathcal{A}_p f(x) = s_p(x)f(x)^T + \bigtriangledown f(x) = \frac{d}{dx}f(x) + f(x)\frac{d}{dx}\log p(x)$. Then we can consider rewriting the stein operator in form of a dot-product:

Lemma 2.5.1 Suppose p and q are two smooth densities support on $\mathcal{Z}$ and g is the stein class of p. Then the expetation of the stein operator of g under q: $\mathbb{E}_q \mathcal{A}_p f(z) = \langle f, \mathbb{E}_q \xi_z \rangle_{\mathcal{F}}$

The proof of Lemma 2.5.1 can be found in Appendix C.

Then the closed-form expression for KSD is as follow:

$$\begin{aligned}
\mathbb{S}(\mathbb{P}, \mathbb{Q}) &= \sup_{||f||_{\mathcal{F}} \leqslant 1} \mathbb{E}_{z\sim q} \mathcal{A}_p f(z) \\
&= \sup_{||f||_{\mathcal{F}} \leqslant 1} \mathbb{E}_{z\sim q} \langle f, \xi_z \rangle_{\mathcal{F}} \\
&= \sup_{||f||_{\mathcal{F}} \leqslant 1} \langle f, \mathbb{E}_{z\sim q} \xi_z \rangle_{\mathcal{F}} \\
&= ||\mathbb{E}_{z\sim q} \xi_z||_{\mathcal{F}}
\end{aligned} \tag{2.22}$$

According to Chwialkowski et al. (2016), there is a closed-form expression for $||\mathbb{E}_{z\sim q}\xi_z||_{\mathcal{F}}$:

$$||\mathbb{E}_{z\sim q}\xi_z||^2_{\mathcal{F}} = \mathbb{E}_{z,z'\sim q}[h_p(z, z')] \tag{2.23}$$

where

$$\begin{aligned}
h_p(x, y) = &\bigtriangledown \log p(x)^T \bigtriangledown \log p(y) k(x, y) \\
&+ \bigtriangledown \log p(y)^T \bigtriangledown_x k(x, y) \\
&+ \bigtriangledown \log p(x)^T \bigtriangledown_y k(x, y) \\
&+ \langle \bigtriangledown_x k(x, \cdot), \bigtriangledown_y k(\cdot, y) \rangle_{\mathcal{F}^d}
\end{aligned} \tag{2.24}$$

To estimate the $\mathbb{E}_{z,z'\sim q}[h_p(z, z')]$ in practice, we use the *U-statistic* [Serfling (2009)]:

Definition 2.5.2 (U-statistic) Let $X_n = \{x_1, x_2, ..., x_n\}$ be i.i.d random variables. Let h be a symmetric kernel function of degree 2. With the assumption: $E[h(x_i, x_j)] < \infty$, we define the U-statistic as:

$$U(h) = C_n^2 \sum_{i<j} h(x_i, x_j) = \frac{1}{n(n-1)} \sum_{i\neq j} h(x_i, x_j) = \frac{2}{n(n-1)} \sum_{i<j} h(x_i, x_j) \tag{2.25}$$

Given the i.i.d sample $\{x_i\}$ drawn from an unknown p and the score function $s_q(x)$, we can estimate $\mathbb{S}(p, q)$ by

$$\hat{\mathbb{S}}_u(p, q) = \frac{1}{n(n-1)} \sum_{1\leqslant i\neq j\leqslant n} h_q(x_i, x_j) \tag{2.26}$$

The above equation is a U-statistic where "U" stands for unbiasedness. Note that there're two assumptions for the U-statistic: 1. h_q should be a symmetric function. 2. $E[h(x_i, x_j)] < \infty, \quad \forall x_i \sim \{X_i\}_{i=1}^n$.

Some papers [Chwialkowski et al. (2016) ; Key et al. (2021)] use *V-statistic* where $\hat{\mathbb{S}}_v(p, q) = \frac{1}{n^2}\sum_{1\leqslant i,j\leqslant n} h_q(x_i, x_j)$. This estimator able to provide a nonnegative result but tends to be biased. More details can be found in Serfling (2009). To avoid biasedness, we will focus on U-statistic in this report. Also, for more information related to Stein's method, see Anastasiou et al. (2021).

2.5 Goodness-of-fit Testing

The Kernel method and Stein method have been widely applied to Hypothesis Testing in Statistics and Machine learning. Goodness-of-fit testing (GOF) is one of the most important hypothesis testing as well as the foundation of our composite relative goodness-of-fit testing (CR-GOF). In this section, we will give a brief introduction to GOF.

Whether in statistical theory or in dealing with practical problems, we often need to answer whether the population distribution of the data belongs to a specific distribution (or family of distributions) that is required by the corresponding statistical model. In other words, can we use a known distribution (or family of distributions) to fit the given data? This is the question that goodness-of-fit testing is designed to investigate.

Since the paper on the Chi-square test (Pearson (1900)), goodness-of-fit testing is often considered by scholars to be the beginning of modern mathematical statistics. Since we usually assume the model is a distribution, generally the goodness-of-fit testing can be regarded as a *distribution comparison problem* as below:

Definition 2.6.1 (Distribution comparison) Suppose $\mathbb{P}$ and $\mathbb{Q}$ are two distributions with respect to two density functions p and q (sometimes only known up to a normalizing constant term). Then the null hypothesis of the distribution comparison problem is $H_0 : \mathbb{P} = \mathbb{Q}$. And the alternative hypothesis is $H_1 : \mathbb{P} \neq \mathbb{Q}$.

In practice, $\mathbb{P} = \mathbb{Q}$ is satisfied only when we can determine their equivalence by the expression of their density function. Otherwise, we cannot guarantee that the two distribution is equivalent to each other. In general, after we take the noise of the sample into consideration, we need a statistical discrepancy to measure the similarity between them. If the discrepancy between them is smaller than a critical value, then we agree that there's no significant difference between these two distributions. However, what is the criteria for a good or bad fit? How to use the discrepancy to build a hypothesis test? How can we test whether the model from a given model set can fit the given data? To answer these questions, a detailed definition of composite goodness-of-fit testing is needed.

Definition 2.6.2 (Composite goodness-of-fit testing) Suppose $m(\mathbb{P}, \mathbb{Q})$ is a statistical discrepancy (follow the definition of 2.4.1) between two distributions $\mathbb{P}$ and $\mathbb{Q}$. Suppose $X_n = \{x_1, x_2, ..., x_n\}$ is a set of data which sample from the distribution $\mathbb{X}$. Then the goodness-of-fit testing is to test the null hypothesis $H_0 : \mathbb{X} \in \{\mathbb{P}\}_{\theta\in\Theta}$, where $\{\mathbb{P}\}_{\theta\in\Theta}$ is a set of parametric distributions (Θ is the parameter space). We choose a specific distribution $P^* \in \{\mathbb{P}\}_{\theta\in\Theta}$ so that the discrepency $m(P^*, \mathbb{X}) = \min_{P\in\mathbb{P}} m(P, \mathbb{X})$. Given a testing level α, if $P(m(P^*, \mathbb{X}) \geqslant c_\alpha) \leqslant \alpha$,where c_α is the $(1-\alpha)$-quantile of the distribution of $m(P^*, \mathbb{X})$, then we can reject the H_0. In parctise, we estimate the optimal model $\hat{P}$ that $m(\hat{P}, X_n) = \min_{P\in\mathbb{P}} m(P, X_n)$.

As different choices of statistic discrepancy m result in different kinds of goodness-of-fit tests, the choice of statistical discrepancy m is crucial. A powerful goodness-of-fit test should have a properly m in order to determine whether the sample X_n has significantly difference from the target distribution $\mathbb{P}$ or not. In generally, there're two types of statistic discrepancy: *Integral Probability Metrics(IPM)* [Müller (1997)] and *ϕ-discrepancy*.

Except for the choice of discrepancy m, there're two important concepts for hypothesis testing: *Type-I-error* and *Power*. The details are given in the Appendix.

With well defined Kernelized Stein discrepancy(KSD) in Gorham and Mackey (2015), Liu et al. (2016) and Chwialkowski et al. (2016) use $\mathbb{S}(\mathbb{P}, X_n)$ as a statistic discrepancy m to construct a

kernel-based quadratic-time goodness-of-fit testing. It allow people to do goodness-of-fit testing when model $\mathbb{P}$ is only known up to a normalizing constant term. Later in Key et al. (2021) extend the non-composite KSD-based goodness-of-fit test into the composite case by using minimum-KSD estimator [Barp et al. (2019)] for parameter estimation. For other scenarios using KSD as goodness-of-fit testing, see Xu (2022). Here we give a brief introduction to this kind of KSD-based goodness-of-fit testing (In short: *K-GOF*).

Since $\mathbb{S}(\mathbb{P}, \mathbb{X})$ is used as a statistical discrepancy, recall section 2 we mentioned that the U-statistic can be used as an efficient estimation of $\mathbb{S}$ in practice. Suppose i.i.d sample $\{x_i\}_{i=1}^n$ is given, we can estimate $\mathbb{S}$ by using a U-statistic:

$$\hat{\mathbb{S}}_u(\mathbb{P}, X_n) = \frac{1}{n(n-1)} \sum_{1 \leqslant i \neq j \leqslant n} h_p(x_i, x_j) \tag{2.27}$$

where h_p is a positive definied matrix follow the definition as equation (2.20).

As it being proved in Liu et al. (2016), when $\mathbb{P} \neq \mathbb{X}$, $\sqrt{n}(\hat{\mathbb{S}}_u(\mathbb{P}, \mathbb{X}_n) - \mathbb{S}(\mathbb{P}, \mathbb{X})) \sim \mathcal{N}(0, \sigma_u^2)$ where $\sigma_u^2 = \mathrm{var}_{x \sim X_n}(\mathbb{E}_{x' \sim X_n})[h_p(x, x')]$. When $\mathbb{P} = \mathbb{X}$, $n\hat{\mathbb{S}}(\mathbb{P}, \mathbb{X}) \sim \sum_{j=1}^{\infty} c_j(Z_j^2 - 1)$ where $\{c_j\}$ are the eigenvalue of h_p and $\{Z_j\}$ are i.i.d standrad gaussian random variable. In order to control the type-I error, we need to obtain the threshold c_α under the null hypothesis. To do that, we need to know the CDF of the null distribution $\mathbb{S}(\mathbb{P}, X_n)$, denote as $F_{\hat{\mathbb{S}}_u}$.

However, the asymptotic distribution of $F_{\hat{\mathbb{S}}_u}$ is difficult to find and may not have an analytic form in practice. To solve this problem, wild bootstrap [Shao (2010); Wu (1986)] and parametric bootstrap [Stute et al. (1993)] are applied. We will elaborate on this idea in section 3. Here we only show the procedure of KSD-based goodness-of-fit (KGOF) with wild bootstrap (Wild - KGOF in Algorithm 1) and with parametric bootstrap (Parametric - KGOF in Algorithm 2) as below:

Algorithm 1: Wild - KGOF

Input: $X_n, \mathbb{P}_\theta, \alpha, b$

θ is given;
for $k \in \{1, ..., b\}$ **do**
  $w^{(k)} = (w_1, ..., w_n)$;
  $\Delta^{(k)} = \frac{1}{n} \sum_{i,j=1}^n w_i^{(k)} w_j^{(k)} h_{p_{\hat{\theta}}}(x_i, x_j)$;
$c_\alpha = \mathrm{quantile}(\{\Delta^{(1)}, ..., \Delta^{(b)}\}, 1 - \alpha)$;
if $\Delta = \hat{\mathbb{S}}_u(\mathbb{P}_{\hat{\theta}}, X_n) \geqslant c_\alpha$ **then**
  reject the null;
else
  do not reject the null;

Algorithm 2: Parametric - KGOF

Input: $X_n, \mathbb{P}_\theta, \alpha, b$

$\hat{\theta^*} = \arg\min_{\theta \in \Theta} \mathbb{S}(\mathbb{P}_\theta, X_n)$;
for $k \in \{1, ..., b\}$ **do**
  $X_n^{(k)} = \frac{1}{n} \sum_{i=1}^n \delta_{y_i^k}, \{y_i^{(k)}\}_{i=1}^n \sim X_n$;
  $\hat{\theta}^{(k)} = \arg\min_{\theta \in \Theta} \mathbb{S}(\mathbb{P}_\theta, X_n^{(k)})$;
  $\Delta^{(k)} = \hat{\mathbb{S}}_u(\mathbb{P}_{\hat{\theta}^{(k)}}, X_n^{(k)})$;
$c_\alpha = \mathrm{quantile}(\{\Delta^{(1)}, ..., \Delta^{(b)}\}, 1 - \alpha)$;
if $\Delta = \hat{\mathbb{S}}_u(\mathbb{P}_{\hat{\theta}}, X_n) \geqslant c_\alpha$ **then**
  reject the null;
else
  do not reject the null;

It's well known that the above goodness-of-fit testing with KSD is a kind of IPM-based goodness-of-fit test since it is motivated by the general approach of using IPMs within a hypothesis testings framework [Anastasiou et al. (2021)]. In Appendix B we review some of the other classical goodness-of-fit testing such as the Chisquare test (based on ϕ discrepancy), the K-S Test. In this section, we only focus on the KSD-based goodness-of-fit test.

2.6 Relative goodenss-of-fit testing

The Relative goodness-of-fit test (Or in short: relative test) is one kind of goodness-of-fit test which aims to evaluate the relative performance of two models. That's it, given two models $\mathbb{P}$ and $\mathbb{Q}$ we wishes to test which one fits the sample $X_n = \{x_i\}_{i=1}^n \in \mathbb{X}$ better.

For this problem, the likelihood ratio test [Lehmann et al. (2005)] is a powerful choice but it does not work for the case when $\mathbb{P}, \mathbb{Q}$ is only known up to a normalizing constant term. Later, the two-sample test [Gretton et al. (2012)] using MMD motivate people to design a sample-based relative goodness-of-fit testing [Bounliphone et al. (2015)]. Rather than benefit from model expression by taking the explicit structure of the models into consideration, it simply draws sample and calculates the MMDs between different models and sample. In Jitkrittum et al. (2018), the Stein features are used to evaluate the performance of each model. In Kanagawa et al. (2019), the difference between the two KSDs is used to build a test statistic for the relative testing of two given latent models. However, these tests do not work for the composite relative goodness-of-fit testing. That's it, $\mathbb{P}, \mathbb{Q}$ is two sets of model: $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$ and $\{\mathbb{Q}_\phi\}_{\phi\in\Phi}$ rather than two specific models. Based on the composite goodness-of-fit test [Key et al. (2021)], we propose a novel composite relative test in section 3. It successfully solves the above problem and its performance is fully discussed in section 4.

2.7 Minimum-KSD estimator

The minimum-KSD estimator (Barp et al. (2019)) is one of the most essential parts involved in the procedure of our composite relative goodness-of-fit testing. In this section, we give a brief introduction to it. Also, we will elaborate the details of how to use minimum-KSD estimator for the exponential family model in Section 4.

Maximum likelihood estimation (MLE) and Maximum a posteriori estimation (MAP) are two kinds of the most commonly used methods for statistical inference. However, they failed to apply to the models that only known up to a normalizing constant term. When they are infeasible, people usually use core matching, contrastive divergence, or minimum probability flow to obtain tractable parameter estimates. In Barp et al. (2019), a unifying technique known as *minimum Stein discrepancy estimators* (SD) is proposed:

Definition 2.7.1 (minimum Stein discrepancy estimators) Given identical and independent (i.i.d) realisations from $\mathbb{X}$ and a sequence of measure (e.g. a distribution family) $\mathcal{P}_\Theta = \{\mathbb{P}_\theta : \theta \in \Theta\}$ where Θ is the parameter space as well as a statistical discrepancy D. Denote the optimal parameter as θ^*. The definition of it is given below:

$$\theta^* = \arg\min_{\theta\in\Theta} D(\mathbb{P}_\theta||\mathbb{X}) \tag{2.28}$$

In practice, instead of $\mathbb{X}$, a i.i.d dataset $\{X_i\}_{i=1}^n = \{x_1, x_2, ..., x_n\} \in \mathbb{X}$ is given. Then we can estimate the optimal parameter θ^* by using the dataset $\{X_i\}_{i=1}^n$, the result is denoted as $\hat{\theta}$.

$$\hat{\theta} = \arg\min_{\theta\in\Theta} D(\mathbb{P}_\theta||\{X_i\}_{i=1}^n) \to \theta^* \text{ when } n \to \infty \tag{2.29}$$

Suppose we use the kernelized Stein discrepancy (KSD) as D, then we obtain the minimum-KSD estimator:

$$\hat{\theta} = \arg\min_{\theta\in\Theta} \mathbb{S}(\mathbb{P}_\theta||\{X_i\}_{i=1}^n) \to \theta^* \text{ when } n \to \infty \tag{2.30}$$

Then the problem is: how to solve the above equation to get the solution of $\hat{\theta}$. In Barp et al. (2019), the Stochastic Riemannian Gradient Descent(SRGD) algorithm is being applied. Here we firstly give the result and then explain it in detail:

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \gamma_t \hat{g}(\{X_i^t\}_{i=1}^n)^{-1} d_{\theta_t} \hat{SD}(\{X_i^t\}_{i=1}^n||\mathbb{P}_\theta) \tag{2.31}$$

Now we explain the meaning of equation (2.31). Recall that the most common Stochastic Riemannian Gradient Descent(SGD) algorithm:

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \gamma_t d_{\theta_t} \mathcal{L}(\{X_i^t\}_{i=1}^n) \tag{2.32}$$

where γ_t is the learning rate in time t and $\mathcal{L}$ is a pre-defined loss function.

For minimum Stein discrepancy estimators, we use Stein discrepancy (such as Langevin Stein discrepancy) as a loss function. Then the crucial question is how to find the gradient of the Stein discrepancy $-d_{\theta_t}\hat{SD}(\{X_i^t\}_{i=1}^n||\mathbb{P}_\theta)$. According to Barp et al. (2019):

$$d_{\theta_t}\hat{SD}(\{X_i^t\}_{i=1}^n||\mathbb{P}_\theta) = g(\theta)^{-1} d_{\theta_t}\hat{SD}(\{X_i^t\}_{i=1}^n||\mathbb{P}_\theta) \tag{2.33}$$

where $g(\theta)$ is the information tensor associated to diffusion kernelized Stein discrepancy (DKSD) and $\hat{g}_{\theta_t}(\{X_i^t\}_{i=1}^n)$ is a unbiased estimator of $g(\theta)$. For more details, see Proposition 3 in Barp et al. (2019). Minimum KSD estimators hold additional appeal for exponential family models since the DKSD of them are convex quadratics with closed form solutions. Then the solution of the minimum-KSD estimator can be easily computed. More discussion on this will be in Section 4.

3 COMPOSITE RELATIVE TEST AND A RESULTING TEST STATISTIC

3.1 Problem setup

The goal of the composite relative goodness-of-fit test is that given two sets of parametric distribution families (usually only known up to their normalizing constant term) $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$ and $\{\mathbb{Q}_\phi\}_{\phi\in\Phi}$ where Θ and Φ are the parameter space of θ and ϕ respectively, we wish to determine which distribution family fits the distribution $\mathbb{X}$ better. The one which fits the distribution better should have a smaller Kernelized Stein discrepancy ($\mathbb{S}$) than the other one. For the non-composite relative goodness-of-fit test (without parameter estimation), the hypothesis test can be formulated as:

$$\begin{aligned} H_0 &: \mathbb{S}(\mathbb{P}_\theta||\mathbb{X}) \leqslant \mathbb{S}(\mathbb{Q}_\phi||\mathbb{X}) \\ H_1 &: \mathbb{S}(\mathbb{P}_\theta||\mathbb{X}) > \mathbb{S}(\mathbb{Q}_\phi||\mathbb{X}) \end{aligned} \tag{3.1}$$

If the distribution family $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$ fits the distribution $\mathbb{X}$ better than $\{\mathbb{Q}_\phi\}_{\phi\in\Phi}$, then there exist at least one model $\mathbb{P}_{\theta^*} \in \{\mathbb{P}_\theta\}_{\theta\in\Theta}$ that $\mathbb{P}_{\theta^*}$ fits $\mathbb{X}$ better than all the models $\{\mathbb{Q}_\phi\}_{\phi\in\Phi}$. Formally,

$$\exists \mathbb{P}_{\theta^*} \in \{\mathbb{P}_\theta\}_{\theta\in\Theta}, \forall \mathbb{Q}_\phi \in \{\mathbb{Q}_\phi\}_{\phi\in\Phi}: \quad \mathbb{S}(\mathbb{P}_{\theta^*}||\mathbb{X}) \leqslant \mathbb{S}(\mathbb{Q}_\phi||\mathbb{X}) \tag{3.2}$$

In practice, we can only obtain a sample $X_n = \{x_1, x_2, ..., x_n\}$ which generated from $\mathbb{X}$. Then the parameter θ and ϕ are need to estimated from the sample X_n, denote as $\hat{\theta}$ and $\hat{\phi}$.

Also, it's possible that there exist more than one models in $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$ that fit the sample X_n better than the best model in $\{\mathbb{Q}_\theta\}_{\phi\in\Phi}$. We can simplify the problem into compare the *best* model in $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$ and $\{\mathbb{Q}_\phi\}_{\phi\in\Phi}$. To choose two optimal models $\mathbb{P}_{\hat{\theta^*}} \in \{\mathbb{P}_\theta\}_{\theta\in\Theta}$ and $\mathbb{Q}_{\hat{\phi^*}} \in \{\mathbb{Q}_\phi\}_{\phi\in\Phi}$, the minimized KSD estimator [Barp et al. (2019)] is applied. Hence, the *best* model $\mathbb{P}_{\hat{\theta^*}}$ can be defined as:

$$\mathbb{P}_{\hat{\theta}} \in \{\mathbb{P}_\theta\}_{\theta\in\Theta}, \quad \text{where } \hat{\theta} = \arg\min_{\theta\in\Theta} \hat{\mathbb{S}}_u(\mathbb{P}_\theta||X_n) \tag{3.3}$$

Under regularity conditions, the estimator $\hat{\theta} \to \theta^* = \arg\min_{\theta\in\Theta} \hat{\mathbb{S}}_u(\mathbb{P}_\theta||\mathbb{X})$ as $n \to \infty$. More details can be found in Barp et al. (2019). In section 4 the details of how to minimize the KSD for the exponential family model (kernelized and non-kernelized) will be introduced. In the next section, test statistics will be built according to our hypothesis.

3.2 Test statistic

As before, $\mathbb{P}_\theta$ is a parametric model with parameter θ and $\mathbb{Q}_\phi$ is a parametric model with parameter ϕ. $\mathbb{X}$ is the population distribution (or data generating process) of a given sample $X_n = \{x_1, x_2, ..., x_n\}$. Then the null hypothesis in equation 3.1 can be equally rewritten in form of the difference of squared KSDs:

$$\mathbb{S}(\mathbb{P}_\theta, \mathbb{X}) - \mathbb{S}(\mathbb{Q}_\phi, \mathbb{X}) \leqslant 0 \tag{3.4}$$

Similar to the idea in Kanagawa et al. (2019), the equation 3.4 motivates us to design a test statistic to estimate the above difference of squared KSDs.

Recall that $\mathbb{S}(\mathbb{P}_\theta, \mathbb{X}) = E_{x_i,x_j}[h_{p_\theta}(x_i, x_j)]$ where $x_i, x_j \in \mathbb{X}$. Given $X_n = \{x_1, x_2, ..., x_n\}$, it can be estimated by a simple closed-form U-statistic:

$$\hat{\mathbb{S}}_u(\mathbb{P}_\theta, X_n) = \frac{1}{n(n-1)} \sum_{1\leqslant i\neq j\leqslant n} h_{p_\theta}(x_i, x_j) \tag{3.5}$$

where

$$\begin{aligned} h_{p_\theta}(x, y) = &\bigtriangledown \log \mathbb{P}_\theta(x)^T \bigtriangledown \log \mathbb{P}_\theta(y) k(x, y) \\ &+ \bigtriangledown \log \mathbb{P}_\theta(y)^T \bigtriangledown_x k(x, y) \\ &+ \bigtriangledown \log \mathbb{P}_\theta(x)^T \bigtriangledown_y k(x, y) \\ &+ \langle \bigtriangledown_x k(x, \cdot), \bigtriangledown_y k(\cdot, y) \rangle_{\mathcal{H}} \end{aligned} \tag{3.6}$$

Here the $k(x, y)$ is a positive definite kernel function with two input $x, y \in X_n$. Further, we simplify the notation $\mathbb{S}(\mathbb{P}_\theta, \mathbb{X})$ into $\mathbb{S}(\mathbb{P}_\theta)$ for later derivation. From now on, $\mathbb{S}(\mathbb{P}_\theta)$ stands for the KSD between the parametric model $\mathbb{P}_\theta$ and distribution $\mathbb{X}$ while $\hat{\mathbb{S}}_u(\mathbb{P}_\theta)$ stands for its U-statistic

$\hat{\mathbb{S}}_u(\mathbb{P}_\theta, X_n)$. For the same idea, we also have $\hat{\mathbb{S}}_u(\mathbb{Q}_\phi) = \frac{1}{n(n-1)} \sum_{1 \leqslant i \neq j \leqslant n} h_{Q_\phi}(x_i, x_j)$. Note that the result in equation (3.6) is a $d \times d$ matrix when $x, y \in \mathbb{R}^d$. When x, y are two scalars, the result remains as a scalar.

With the above notation, let's define the difference of the two KSDs as below:

Definition 3.2.1 (Difference of KSDs)

$$\mathbb{S}(\mathbb{P}_\theta, \mathbb{Q}_\phi) = \mathbb{S}(\mathbb{P}_\theta) - \mathbb{S}(\mathbb{Q}_\phi) = E_{x_i, x_j}[h_{p_\theta, q_\phi}(x_i, x_j)] \tag{3.7}$$

where $h_{p_\theta, q_\phi}(x_i, x_j) = h_{p_\theta}(x_i, x_j) - h_{q_\phi}(x_i, x_j)$ for $x_i, x_j \in \mathbb{X}$.

Since $\hat{\mathbb{S}}_u(\mathbb{P}_\theta)$ and $\hat{\mathbb{S}}_u(\mathbb{Q}_\phi)$ are two U-statistics with respect to two difference kernel h_{p_θ} and h_{q_ϕ}. Recall the definition of U-statistic in definition 2.5.2, when two assumptions is satisfied (h is a symmetric matrix and $E[h(x_i, x_j)] < \infty$, then we can consturct a U-statistic. Since the difference of two symmetric matrix $h_{p_\theta} - h_{q_\phi}$ also a symmetric matrix, and $E[(h_{p_\theta} - h_{q_\phi})(x_i, x_j)] = E[h_{p_\theta}(x_i, x_j)] - E[h_{q_\phi}(x_i, x_j)] < \infty$, so that the difference of two U-statistic is also a U-statistic. Similar to the previous definition, let's define their difference as below:

Definition 3.2.2 (Test statistic)

$$\hat{\mathbb{S}}_u(\mathbb{P}_\theta, \mathbb{Q}_\phi) = \hat{\mathbb{S}}(\mathbb{P}_\theta, \mathbb{X}) - \hat{\mathbb{S}}_u(\mathbb{Q}_\phi, \mathbb{X}) = \frac{1}{n(n-1)} \sum_{1 \leqslant i \neq j \leqslant n} h_{p_\theta, q_\phi}(x_i, x_j) \tag{3.8}$$

where $h_{p_\theta, q_\phi}(x_i, x_j) = h_{p_\theta}(x_i, x_j) - h_{q_\phi}(x_i, x_j)$ for $x_i, x_j \in X_n$.

The form in equation (3.8) allows efficient estimation of $\mathbb{S}(\mathbb{P}_\theta, \mathbb{Q}_\phi)$.

In the next section, we will talk about how to use the above test statistic to build a composite relative goodness-of-fit test.

3.3 Test procedure

Given two sets of candidate models $\{\mathbb{P}_\theta\}_{\theta \in \Theta}, \{\mathbb{Q}_\phi\}_{\phi \in \Phi}$ (usually only known up to normalizing constant term) and a sample $X_n = \{x_1, x_2, ..., x_n\} \in \mathbb{X}$. The test procedure of our novel composite relative goodness-of-fit test is two-stages testing as below:

Stage 1 (Estimation): $\hat{\theta} = \arg\min_{\theta \in \Theta} KSD^2(\mathbb{P}_\theta || X_n), \quad \hat{\phi} = \arg\min_{\phi \in \Phi} KSD^2(\mathbb{Q}_\phi || X_n)$

Stage 2 (Testing): reject H_0 if $\hat{\mathbb{S}}_u(\mathbb{P}_{\hat{\theta}}, \mathbb{Q}_{\hat{\phi}}) \geqslant c_\alpha$

Here c_α is the threshold (critical value) of the test. Formally, denote $F_{\hat{\mathbb{S}}_u}$ as the CDF of $\hat{\mathbb{S}}_u(\mathbb{P}_\theta, \mathbb{Q}_\phi)$ under the null $\mathbb{S}(\mathbb{P}_\theta) - \mathbb{S}(\mathbb{Q}_\phi) \leqslant 0$. Then set c_α as the $1 - \alpha$ quantile of $F_{\hat{\mathbb{S}}_u}$.

In practice, the asymptotic distribution of $F_{\hat{\mathbb{S}}_u}$ is difficult to find and may not have an analytic form. However, as Liu et al. (2016) mentioned, these types of problems always show up in statistical hypothesis testing problems such as the Anderson-Darling test, Cramer-von Mises test, Kolmogorov-Smirnov test [Kolmogorov (1933a)] and two-sample tests [Gretton et al. (2012)]. This problem result in different approximation methods such as *Bootstrap*[Arcones and Gine (1992); Huskova and Janssen (1993)] and eigenvalue approximation.

For stage 2, the bootstrap method will be applied to estimate the threshold c_α. Here we only give a brief introduction of *Wild Bootstrap* [Shao (2010)] and *Parametric Bootstrap* in the composite relative goodness-of-fit testing. The main idea of Boostrap is resampling from a

given sample for multiple times in order to mimic the data generating process (the population distribution). It allows us to simulate from the null distribution to compute test thresholds.

Recall the problem of non-composite goodness-of-fit testing: given a parametric model $\mathbb{P}_\theta$ and a sample $X_n \in \mathbb{X}$, we wish to test whether the model $\mathbb{P}_\theta$ can fit the sample X_n well. For most of the research on this area [Liu et al. (2016); Chwialkowski et al. (2016); Fernandez and Gretton (2019); Key et al. (2021); Xu (2022) etc], the wild-bootstrap is widely used for estimating the test threshold c_α. Different from the problem of composite goodness-of-fit testing, the non-composite goodness-of-fit testing doesn't require people to choose an optimal model P_{θ^*} from a model set $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$ by using the parameter estimation techniques such as Maximum Likelihood Estimator or Minimum-KSD Estimator [Barp et al. (2019)] since the target model $\mathbb{P}_\theta$ is given. Hence the wild bootstrap which doesn't involve a parameter estimation procedure as well as able to attain the correct significance level asymptotically (see Huskova and Janssen (1993)) is regarded as the so-called "gold-standard" method.

To extend the goodness-of-fit testing to a composite case (given a parametric model set $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$ rather than a single model P_θ), the parameter estimation procedure is involved. For this problem, the wild bootstrap as well as the parametric bootstrap are being applied and discussed, see Key et al. (2021). Similar to this idea, they can also be applied in our composite relative goodness-of-fit problem. The testing procedure using wild-bootstrap is given in Algorithm 1. Note that the different ways of resampling (different settings of the weights w_i, w_j) result in different wild bootstrap methods. Two commonly used wild bootstrap methods will be introduced in section 4.

Algorithm 3: Wild bootstrap test

Input: $X_n, \mathbb{P}_\theta, \mathbb{Q}_\phi, \alpha, b$

$\hat{\theta} = \arg\min_{\theta\in\Theta} \mathbb{S}(\mathbb{P}_\theta, X_n)$;

$\hat{\phi} = \arg\min_{\phi\in\Phi} \mathbb{S}(\mathbb{Q}_\phi, X_n)$;

for $k \in \{1, ..., b\}$ **do**

  $w^{(k)} = (w_1, ..., w_n)$;

  $\Delta^{(k)} = \frac{1}{n}\sum_{i,j=1}^n w_i^{(k)} w_j^{(k)} h_{p_{\hat{\theta}}, q_{\hat{\phi}}}(x_i, x_j)$;

$c_\alpha = \text{quantile}(\{\Delta^{(1)}, ..., \Delta^{(b)}\}, 1-\alpha)$;

if $\Delta = \hat{\mathbb{S}}_u(\mathbb{P}_{\hat{\theta}}, \mathbb{Q}_{\hat{\phi}}) \geqslant c_\alpha$ **then**

  reject the null;

else

  Do not reject;

Algorithm 4: Parametric bootstrap test

Input: $X_n, \mathbb{P}_\theta, \mathbb{Q}_\phi, \alpha, b$

$\hat{\theta^*} = \arg\min_{\theta\in\Theta} \mathbb{S}(\mathbb{P}_\theta, X_n)$;

$\hat{\phi^*} = \arg\min_{\phi\in\Phi} \mathbb{S}(\mathbb{Q}_\phi, X_n)$;

for $k \in \{1, ..., b\}$ **do**

  $X_n^{(k)} = \frac{1}{n}\sum_{i=1}^n \delta_{y_i^k}, \{y_i^{(k)}\}_{i=1}^n \sim X_n$;

  $\hat{\theta^*}^{(k)} = \arg\min_{\theta\in\Theta} \mathbb{S}(\mathbb{P}_\theta, X_n^{(k)})$;

  $\hat{\phi^*}^{(k)} = \arg\min_{\phi\in\Phi} \mathbb{S}(\mathbb{Q}_\phi, X_n^{(k)})$;

  $\Delta^{(k)} = \hat{\mathbb{S}}_u(\mathbb{P}_{\hat{\theta^*}^{(k)}}, \mathbb{Q}_{\hat{\phi^*}^{(k)}})$;

$c_\alpha = \text{quantile}(\{\Delta^{(1)}, ..., \Delta^{(b)}\}, 1-\alpha)$;

if $\Delta = \hat{\mathbb{S}}_u(\mathbb{P}_{\hat{\theta^*}}, \mathbb{Q}_{\hat{\phi^*}}) \geqslant c_\alpha$ **then**

  reject the null;

else

  Do not reject;

The wild bootstrap can be successfully applied for the non-composite goodness-of-fit testing [Liu et al. (2016); Chwialkowski et al. (2016); Fernandez and Gretton (2019); Xu (2022) ect] since $F_{\hat{\mathbb{S}}_u}$ (the CDF of $\{\Delta^{(1)}, \Delta^{(2)}, ..., \Delta^{(b)}\}$) will converge to $F_{\mathbb{S}}$ (CDF of the $\mathbb{S}$) when $b \to \infty$. Also, $\Delta^{(k)}$ is converge to Δ when $n \to \infty$. However, it may raise an error for the composite case when the sample size is small, resulting in obtaining the wrongly approximation of asymptotic distribution $F_{\hat{\mathbb{S}}_u}$ and the test threshold c_α.

The error is raised parameter estimation: in composite relative goodness-of-fit problem, we need to select two optimal models $\mathbb{P}_{\theta^*}, \mathbb{Q}_{\phi^*}$ which fit $\mathbb{X}$ best. However, we can only obtain a sample set $X_n = \{x_1, x_2, ..., x_n\}$ rather than the population $\mathbb{X}$ in practice. Hence we can only obtain

the estimation $\mathbb{P}_{\hat{\theta}}, \mathbb{Q}_{\hat{\phi}}$ of $\mathbb{P}_{\theta^*}, \mathbb{Q}_{\phi^*}$ respectively by using the minimum KSD estimator [Barp et al. (2019)]. If we directly assume that $\hat{\theta} = \theta^*, \hat{\phi} = \phi^*$, then it will lead a wrong approximation of $F_{\hat{\mathbb{S}}_u}$ since each times in the bootstrap procedure, we use a new sample $X_n^{(k)}$ $(k \in 1, ..., b)$ to calculate the test statistic $\Delta^{(k)}$ rather than use the original data set X_n. Although $X_n^{(k)}$ is generate by X_n, the asymptotic distribution of them may be difference in practise and result in wrong approximation of $F_{\hat{\mathbb{S}}_u}$ and c_α. In section 4 you can see that by using different methods to draw samples $X_n^{(k)}$ from X_n results in different kinds of error of c_α as well as testing power.

To solve this question, we need to take the estimation error into account by re-estimate the parameter with a new sample $X_n^{(k)}$ during the bootstrap procedure. That is, given a new sample $X_n^{(k)}$, we need to estimate the parameters $\hat{\theta}^{(k)}, \hat{\phi}^{(k)}$ and use them to obtain the value of $\Delta^{(k)}$ rather than directly use $\hat{\theta}, \hat{\phi}$ like the wild bootstrap. This kind of new bootstrap which take parameter estimation error into consideration is known as *parametric bootstrap*. [Stute et al. (1993); Key et al. (2021)]. The whole procedure of it is given in Algorithm 2.

4 EXPERIMENT

Recall the null hypothesis of KSD-based composite relative goodness-of-fit testing (KCR-GOF):

$$\mathbb{S}(\mathbb{P}_{\theta^*}||\mathbb{X}) - \mathbb{S}(\mathbb{Q}_{\phi^*}||\mathbb{X}) \leqslant 0 \tag{4.1}$$

where $\theta^* = \arg\min_{\theta\in\Theta} \mathbb{S}(\mathbb{P}_\theta||\mathbb{X})$ and $\phi^* = \arg\min_{\phi\in\Phi} \mathbb{S}(\mathbb{Q}_\phi||\mathbb{X})$

In practice, when the sample $X_n = \{x_i\}_{i=1}^n$ is given, we can check:

$$\hat{\mathbb{S}}_u(\mathbb{P}_{\hat{\theta}}||X_n) - \hat{\mathbb{S}}_u(\mathbb{Q}_{\hat{\phi}}||X_n) \leqslant 0 \tag{4.2}$$

where $\hat{\theta} = \arg\min_{\theta\in\Theta} \hat{\mathbb{S}}_u(\mathbb{P}_\theta||X_n)$ and $\hat{\phi} = \arg\min_{\phi\in\Phi} \hat{\mathbb{S}}_u(\mathbb{Q}_\phi||X_n)$

When model $\mathbb{P}_\theta$ fits the sample X_n better, then $\hat{\mathbb{S}}_u(\mathbb{P}_{\hat{\theta}}||X_n)$ is samll than the $\hat{\mathbb{S}}_u(\mathbb{Q}_{\hat{\phi}}||X_n)$.

In section 4.1, we start with a toy case of one-dimensional Gaussian and Laplace model which we can control the data generating process so that the type I error and testing power could be obtained. This section include three experiments. In **experiment 1**, We set the first set of candidate models $\{\mathbb{P}\}_{\theta\in\Theta}$ to be the Gaussian model, and the second set of candiate models $\{\mathbb{Q}\}_{\phi\in\Phi}$ to be the Laplace model. In **experiment 2**, we reverse the null hypothesis by setting $\{\mathbb{P}\}_{\theta\in\Theta}$ to be the Laplace model and $\{\mathbb{Q}\}_{\phi\in\Phi}$ to be the Gaussian model. For these two experiment, two kinds of bootstarp is applied. In **experiment 3**, we focus on the parametirc bootstrap.

The KCR-GOF can also handle the model that only known up to a normalization constant term. In section 4.2, the kernelized exponential family model and another model (gaussian mixture?) will used to modeling the *Galaxy dataset* [Postman et al. (1986)].

4.1 Gaussian models v.s. Laplace models

In this section, we consider two sets of one-dimensional candidate models: Gaussian model and Laplace model with equal shift parameter (mean value) and unknown scale parameter.

In our experiments, we fix the choice of kernel and two kinds of wild-bootstrap methods are applied. Recall the procedure of Wild-bootstrap in Algorithm 3, the test statistic from a bootstrap sample is calculated by $\Delta^{(k)} = \frac{1}{n}\sum_{i,j=1}^{n} w_i^{(k)} w_j^{(k)} h_{p_{\hat{\theta}},q_{\hat{\phi}}}(x_i, x_j)$. There're several ways to set the weight w_i, w_j. Here we choose two ways which most commonly used: the first way is from Liu et al. (2016), by setting the $w_i, w_j \sim \mathit{Multinomial}(n, 1/n, ..., 1/n) - 1$ (denoted as Wild Bootstrap 1). The second way is from Chwialkowski et al. (2016), Key et al. (2021), by setting the $w_i, w_j \sim \mathit{Rademacher}$ (denoted as Wild Bootstrap 2).

Experiemernt 1: We set the first set of candidate models $\{\mathbb{P}\}_{\theta\in\Theta}$ to be the Gaussian model, and the second set of candidate models $\{\mathbb{Q}\}_{\phi\in\Phi}$ to be the Laplace model. Under H_0, we generate sample $X_n = \{x_i\}_{i=1}^n \sim \mathcal{N}(3, 1)$ with different sample size n. Under H_1, we generate sample $X_n = \{x_i\}_{i=1}^n \sim \text{Laplace}(3, 1)$.

Under H_0 (that's means the sample is generated from a normal distribution), then the Gaussian model $\mathbb{P}_\theta$ (with estimated scale parameter $\theta = \sigma$) fits the sample better than the Laplace model $\mathbb{Q}_\phi$ (with estimated scale parameter $\phi = b$). In opposite, under H_1, then the Laplace model fits better than the gaussian model.

Before the type I error and testing power is given, an intuitive example is shown in Figure 3. In figure 3, we set $n = 150$. Under the different hypotheses, we plot the curve with the truth value of the scale parameter in blue line while the estimated curves of the two models are in different colors. Note that we use minimum-KSD estimator [Barp et al. (2019)] for parameter estimation. A closed-form expression for the KSD estimator is provided in Appendix C.

From the result in Figure 3 we can conclude two facts: 1. The performance of parameter estimation is good since the curve of truth distribution and estimated distribution are close to each other. 2. Our method is able to make testing the decisions with strong evidence: when under or not under the null hypothesis, the p-values are 0.998 and 0.00 respectively.

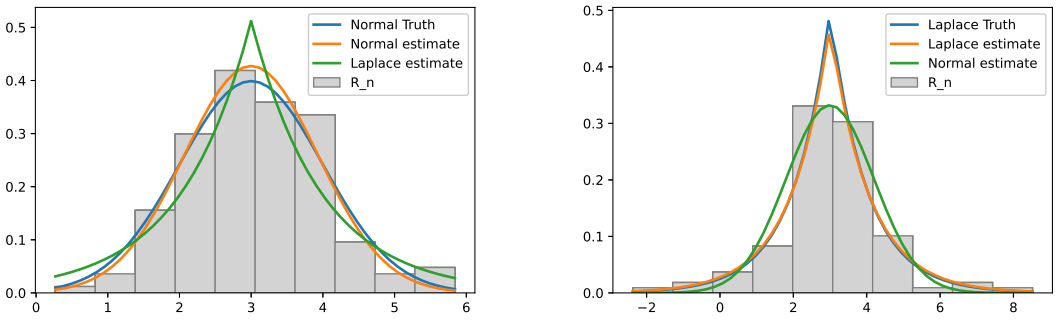

Figure 2: Visualization of composite relative goodness-of-fit test under H_0 and H_1. **Left**: Under H_0: Sample is generated from $\mathcal{N}(3, 1)$ hence that Gaussian model should fits the data better than Laplace model. Two candidate models obtained by parameter estimation: $\mathcal{N}(3, 1.0455)$ and Laplace$(3, 1.0946)$; Test Statistics = -0.00815; P value= 0.998; **Test result**: Do not reject the H_0; **Right**: Under H_1: Sample is generated from Laplace$(3, 1)$ hence that Laplace model should fits the data better than Gaussian model. Two candidate models obtained by parameter estimation: $\mathcal{N}(3, 1.2009)$ and Laplace$(3, 1.05375)$; Test Statistics = 0.01794; P value= 0.00; **Test result**: reject the H_0.

As we know the result of a single test may come from coincidence. To fully investigate the performance of KCR-GOF, we repeat the experiment for $m = 300$ times for different settings of sample size (from 20 to 200). Then we obtain the results of Type I error and testing Power which are shown in Figure 4. The raw data of them are shown in Table 1.

model	$n = 20$	$n = 40$	$n = 60$	$n = 80$	$n = 100$	$n = 150$	$n = 200$
Wild-1 Type I error	0.28	0.18666	0.17333	0.12	0.05333	0.06	0.02333
Wild-2 Type I error	0.05333	0.03333	0.03333	0.04333	0.03666	0.03666	0.03666
Wild-1 Power	0.46333	0.57	0.65333	0.71334	0.79666	0.86333	0.94
Wild-2 Power	0.153332	0.32333	0.406667	0.51	0.6	0.77333	0.85

Table 1: Experiment 1: Type I error for two models for $\alpha = 0.05$. "Wild-1" stands for the first method of the wild bootstrap and "Wild-2" stands for the second method.

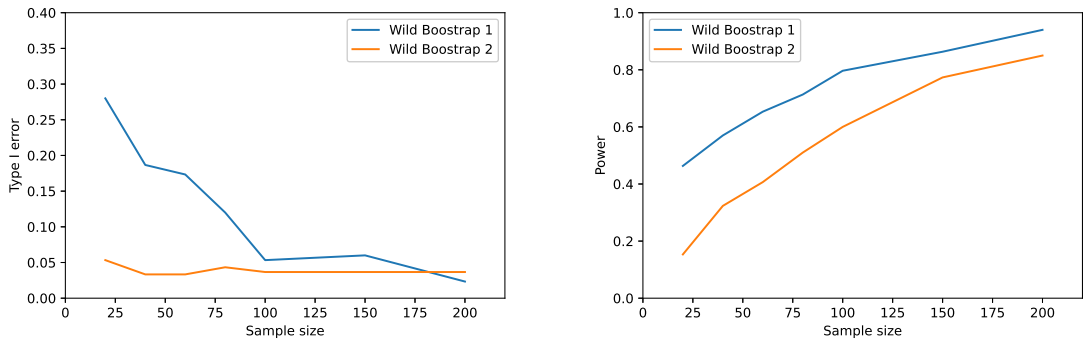

Figure 3: Type I error and testing power for **experiment 1** that $H_0 : \mathbb{S}(\mathbb{P}_\theta, \mathbb{X}) - \mathbb{S}(\mathbb{Q}_\phi, \mathbb{X}) \leqslant 0$ where $\mathbb{P}_\theta$ is choosed from Gaussian distribution, $\mathbb{Q}_\phi$ is choosed from Laplace distribution. **Left:** Type I error. **Right:**Testing Power.

Discussion of experiment 1:

In experiment 1, we investigate the type I error and testing power. Under the null hypothesis, the Gaussian model should fit the sample better than the Laplace model. For both methods (with different ways of Wild bootstrap), the testing power are increase as the sample size. Eventually, they will converge to 1. That means our test is consistent with the sample size. They are powerful even when the sample size is small. When sample size $n \geqslant 100$, the Type I error can be well controlled at around 0.05.

However, when the sample size is not large enough (smaller than 100), the type I error of the first wild bootstrap method is larger than the second method, resulting in unfairness in the comparison of their testing power. To compare the testing power in a fair way, we need to control their type I error at around 0.05.

Experiemernt 2: Now we reverse the null hypothesis. Under the H_0, the Laplace models should fit the sample better than the Gaussian models since the sample is generated from the Laplace distribution. In formal: $H_0 : \mathbb{S}(\mathbb{P}_\theta, \mathbb{X}) - \mathbb{S}(\mathbb{Q}_\phi, \mathbb{X}) \leqslant 0$ where $\mathbb{P}_\theta$ is choose from Laplace distribution, $\mathbb{Q}_\phi$ is choose from Gaussian distribution. The results of experiment 2 are shown in Table 2 and Figure 5.

model	$n = 20$	$n = 40$	$n = 60$	$n = 80$	$n = 100$	$n = 150$	$n = 200$
Wild-1 Type I error	0.31	0.19333	0.14	0.11666	0.08	0.025	0.01333
Wild-2 Type I error	0.11	0.04666	0.05333	0.033333	0.04	0.013333	0.003
Wild-1 Power	0.53	0.64	0.61	0.64	0.703333	0.813333	0.833333
Wild-2 Power	0.5333	0.60333	0.61	0.683333	0.713333	0.803333	0.84666

Table 2: Experiment 2: Type I error for two models for $\alpha = 0.05$. "Wild-1" stands for the first method of the wild bootstrap and "Wild-2" stands for the second method.

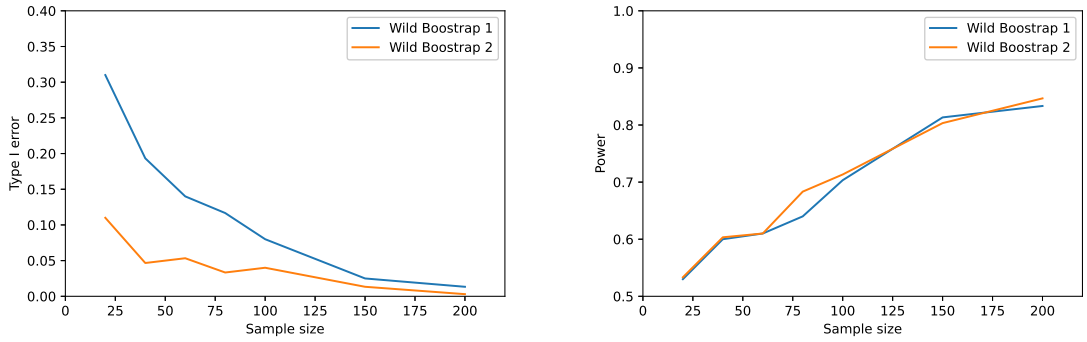

Figure 4: Type I error and testing power for two for Gaussian v.s. Laplace case. Left: Type I error). Right:Testing Power

Discussion of experiment 2:

Under the H_0 of experiment 2, the Laplace should fit the sample (generate from Laplace distribution) better than the Gaussian model. The Type I error of the above two methods is given in Table 1 and Figure 4.

Similar to experiment 1, although the testing power is increase as the sample size, the type I error cannot be controlled very well when the sample size is small (less than 100).

Conclusion of experiment 1 and 2:

To sum up the results of CR-KGOF with two kinds of wild bootstrap in experiments 1 and 2, the testing power is increase as the sample size while different types of bootstrap methods result in different performances. As the H_0 for these two experiments are opposite (for experiment 1, the Gaussian model should fit the data better; for experiment 2, the Laplace model should fit the data better), the CR-KOF is able to determine the better candidate model under different H_0.

However, these bootstrap methods own their drawback of wrongly approximating the critical value c_α resulting in uncontrolled type I error. As we know the comparison of testing power is fair only when the type I error is well controlled at around 0.05. In experiments 1 and 2, the type I error tends to be large than 0.05 resulting in over-estimating the testing power when the sample size is small (less than 100). When the sample size is large, the type I error tends to be smaller than 0.05 resulting in underestimating the testing power.

As mentioned in Key et al. (2021), there're two types of error in composite goodness-of-fit testing. The first error comes from the limitation of sample X_n. That is, we can only approximate the KSD (between the optimal models $\mathbb{P}_{\theta^*}, \mathbb{Q}_{\phi^*}$ and distribution $\mathbb{X}$) $\mathbb{S}(\mathbb{P}_{\theta^*}||\mathbb{X}), \mathbb{S}(\mathbb{Q}_{\phi^*}||\mathbb{X})$ by using the U-statistic $\hat{\mathbb{S}}_u(\mathbb{P}_{\hat{\theta}}||X_n), \hat{\mathbb{S}}_u(\mathbb{Q}_{\hat{\phi}}||X_n)$ since we cannot obtain the population of X_n. The second error comes from the estimation of parameter estimation since we cannot obtain

the optimal parameters θ^*, ϕ^* by using the given sample set X_n. Although we can get the estimation of them $(\hat{\theta}, \hat{\phi})$ by using the minimum-KSD estimator, it may introduce the second type of error.

Experiment 3: In this experiment, the parametric bootstrap method is applied in order to control the type I error. The formal procedure of this algorithm is given in section 3.

For the bootstrap-based goodness-of-fit test, we need to repeat the experiment for b times to get the approximation of critical value c_α. During the procedure of bootstrap, everytime we obtain a new sample $X_n^{(k)}$ we need to calculate the $\hat{\mathbb{S}}_u(\mathbb{P}_{\hat{\theta}^{(k)}}||X_n^{(k)})$ where $\hat{\theta}^{(k)}$ is obtain by minimum-KSD estimator with $X_n^{(k)}$. However, the wild bootstrap use $\hat{\mathbb{S}}_u(\mathbb{P}_{\hat{\theta}}||X_n^{(k)})$ instead of $\hat{\mathbb{S}}_u(\mathbb{P}_{\hat{\theta}^{(k)}}||X_n^{(k)})$ where $\hat{\theta}$ is obtain by minimum-KSD estimator with X_n. This may introduce the approximation error of c_α resulting in type I error.

The basic idea of parametric bootstrap follows the correct procedure that we mention above. That is, every time we obtain a new sample $X_n^{(k)}$, we will re-estimate the $\hat{\theta^{(k)}}$ and re-calculate the $\hat{\mathbb{S}}_u(\mathbb{P}_{\hat{\theta}^{(k)}}||X_n^{(k)})$. This method is able to avoid the approximation error of c_α when b is large.

However, this method doesn't work when b is a small value (For wild bootstrap we choose $b = 5000$) since the approximate error may also introduce by the number of repeat times b. For wild-bootstrap, we can set the b to be a large number since we only need to estimate the parameters once. However, for parametric bootstrap, we need to estimate the parameters for b times. Then this algorithm is low-efficiency when b is large. Since the value of b has to be large, the parametric bootstrap always lack of efficiency.

For this reason, we do not provide the type I error and testing power as we did in experiments 1 and 2. Here we only provide the relative computational time as below:

Bootstrap method	Wild Bootstrap 1	Wild Bootstrap 2	Parametric bootstrap
n = 20	1	1.333	1260
n = 40	1	1.7	761
n = 60	1	2.1	415

Table 3: Experiment 3: Relative time cost of three kinds of bootstrap methods. We set the relative time of Wild Bootstrap 1 as 1. When $n = 20$, if Wild Bootstrap take 1 second to calculate the power, then the parametric bootstrap will take 1260 seconds. In this experiment, $b = 1000$, experiment times $m = 5$. The kernel function is IMQ kernel with huristic method.

5 Conclusion

The generative model becomes more and more popular these days. For this kind of model, we wish to train a model to learn the distribution of given samples in order to generate some new samples. Except for optimizing the parameters of the given model, selecting a proper type of model is also important. Given two generative models, how to determine which one fit the data better?

When two models are non-parametric models, we can only access their performance by evaluating the samples they generate. However, if the two models are the parametric generative model (graphical model such as the Boltzmann machine), then we can make use of the information inside their density function (such as the structure and dependency between different variables).

The main challenge is that the classical goodness-of-fit method fails to evaluate the model which is only known up to a normalization constant term. Liu et al. (2016) proposed a KSD-based goodness-of-fit that can bypass the calculation of normalization constant term. Later Key et al. (2021) extend it to the composite case.

Inspired by these two papers, we propose a novel method for the composite relative goodness-of-fit problem base on Kernelized Stein discrepancy, known as KCR-GOF. Different from the composite goodness-of-fit test that only tests whether a set of given parametric models can fit the data, the goal of our method is to compare two sets of parametric models and determined which one can fit the given data better.

The KCR-GOF is two-stage testing: firstly, we select two models from two sets of models respectively by using the minimum-KSD estimator. Then we calculate the KSD between these models and the given sample result in two KSDs. The difference between these two KSDs can be used to build a goodness-of-fit test.

In the experiment part, we consider the Gaussian model and the Laplace model as two candidate model sets. With two types of wild bootstrap methods, we found that the KCR-GOF is able to determine which model fit the given data better under the different null hypothesis. Also, we do an investigation of parametric bootstrap and found that this algorithm always lacks efficiency.

Except for the one-dimensional model, the KCR-GOF also works for the model which is only known up to a normalization constant term. Due to the limitation of time, we do not provide more experiment details of this kind of model. There're two directions for future work: 1. Use the Kernelized exponential model and Gaussian model as two candidate models to model the galaxy dataset [Postman et al. (1986)]. 2. Do more research on the kernel choice to improve the performance of the KCR-GOF.

References

Anastasiou, A., A. Barp, F.-X. Briol, B. Ebner, R. E. Gaunt, F. Ghaderinezhad, J. Gorham, A. Gretton, C. Ley, Q. Liu, et al. (2021). Stein's method meets statistics: A review of some recent developments. *arXiv preprint arXiv:2105.03481*.

Arcones, M. A. and E. Gine (1992). On the bootstrap of u and v statistics. *The Annals of Statistics*, 655–674.

Balakrishnan, N., V. Voinov, and M. S. Nikulin (2013). *Chi-squared goodness of fit tests with applications*. Academic Press.

Barp, A., F.-X. Briol, A. Duncan, M. Girolami, and L. Mackey (2019). Minimum stein discrepancy estimators. *Advances in Neural Information Processing Systems 32*.

Betsch, S. and B. Ebner (2020). Testing normality via a distributional fixed point property in the stein characterization. *TEST 29*(1), 105–138.

Bounliphone, W., E. Belilovsky, M. B. Blaschko, I. Antonoglou, and A. Gretton (2015). A test of relative similarity for model selection in generative models. *arXiv preprint arXiv:1511.04581*.

Chen, W. Y., L. Mackey, J. Gorham, F.-X. Briol, and C. Oates (2018). Stein points. In *International Conference on Machine Learning*, pp. 844–853. PMLR.

Chwialkowski, K., H. Strathmann, and A. Gretton (2016). A kernel test of goodness of fit. In *International conference on machine learning*, pp. 2606–2615. PMLR.

Cochran, W. G. (1952). The $\chi 2$ test of goodness of fit. *The Annals of mathematical statistics*, 315–345.

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine learning 20*(3), 273–297.

Cox, D. R. (1957). Note on grouping. *Journal of the American Statistical Association 52*(280), 543–547.

Cramer, H. (1946). Mathematical methods of statistics, princeton univ. *Press, Princeton, NJ*.

Fang, K.-T. and S.-D. He (1988). The problem of selecting a given number of representative points in a normal population and a generalized mills' ratio. Technical report, STANFORD UNIV CA DEPT OF STATISTICS.

Fernandez, T. and A. Gretton (2019). A maximum-mean-discrepancy goodness-of-fit test for censored data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2966–2975. PMLR.

Flury, B. A. (1990). Principal points. *Biometrika 77*(1), 33–41.

Gareth, J., W. Daniela, H. Trevor, and T. Robert (2013). *An introduction to statistical learning: with applications in R*. Spinger.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative adversarial nets. *Advances in neural information processing systems 27*.

Gorham, J. and L. Mackey (2015). Measuring sample quality with stein's method. *Advances in Neural Information Processing Systems 28*.

Greenwood, P. E. and M. S. Nikulin (1996). *A guide to chi-squared testing*, Volume 280. John Wiley & Sons.

Gretton, A., K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola (a2012). A kernel two-sample test. *The Journal of Machine Learning Research 13*(1), 723–773.

Gretton, A., D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. K. Sriperumbudur (b2012). Optimal kernel choice for large-scale two-sample tests. *Advances in neural information processing systems 25*.

Hamdan, M. (1963). The number and width of classes in the chi-square test. *Journal of the American Statistical Association 58*(303), 678–689.

Henze, N., S. G. Meintanis, and B. Ebner (2012). Goodness-of-fit tests for the gamma distribution based on the empirical laplace transform. *Communications in Statistics-Theory and Methods 41*(9), 1543–1556.

Henze, N. and J. Visagie (2020). Testing for normality in any dimension based on a partial differential equation involving the moment generating function. *Annals of the Institute of Statistical Mathematics 72*(5), 1109–1136.

Huskova, M. and P. Janssen (1993). Consistency of the generalized bootstrap for degenerate u-statistics. *The Annals of Statistics*, 1811–1823.

Jitkrittum, W., H. Kanagawa, P. Sangkloy, J. Hays, B. Schölkopf, and A. Gretton (2018). Informative features for model comparison. *Advances in Neural Information Processing Systems 31*.

Kanagawa, H., W. Jitkrittum, L. Mackey, K. Fukumizu, and A. Gretton (2019). A kernel stein test for comparing latent variable models. *arXiv preprint arXiv:1907.00586*.

Key, O., T. Fernandez, A. Gretton, and F.-X. Briol (2021). Composite goodness-of-fit tests with kernels. *arXiv preprint arXiv:2111.10275*.

Kim, B., R. Khanna, and O. O. Koyejo (2016). Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems 29*.

Kingma, D. P. and M. Welling (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Koehler, K. J. and K. Larntz (1980). An empirical investigation of goodness-of-fit statistics for sparse multinomials. *Journal of the American Statistical Association 75*(370), 336–344.

Kolmogorov, A. (1933a). *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Springer-Verlag.

Kolmogorov, A. (1933b). Sulla determinazione empirica di una lgge di distribuzione. *Inst. Ital. Attuari, Giorn. 4*, 83–91.

Lehmann, E. L., J. P. Romano, and G. Casella (2005). *Testing statistical hypotheses*, Volume 3. Springer.

Ley, C. and Y. Swan (2013). Stein's density approach and information inequalities. *Electronic Communications in Probability 18*, 1–14.

Li, J., X. Peng, and J. Liang (2022). An rp-aided pearson-fisher chi-square approach to testing goodness-of-fit for location-scale distributions. *Mathematics*.

Lim, J. N., M. Yamada, B. Schölkopf, and W. Jitkrittum (2019). Kernel stein tests for multiple model comparison. *Advances in Neural Information Processing Systems 32*.

Liu, Q., J. Lee, and M. Jordan (2016). A kernelized stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, pp. 276–284. PMLR.

Maydeu-Olivares, A. and L. Cai (2006). A cautionary note on using g2 (dif) to assess relative model fit in categorical data analysis. *Multivariate Behavioral Research 41*(1), 55–64.

Muandet, K., K. Fukumizu, B. Sriperumbudur, B. Schölkopf, et al. (2020). Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning 10*(1-2), 1–141.

Müller, A. (1997). Integral probability metrics and their generating classes of functions. *Advances in Applied Probability 29*(2), 429–443.

Neyman, J. (1949). Contribution to the theory of the chi-square test. *Press, University of California*.

Oates, C. J., M. Girolami, and N. Chopin (2014). Control functionals for monte carlo integration. *arXiv e-prints*, arXiv–1410.

Pearson, K. (1900). X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 50*(302), 157–175.

Postman, M., J. P. Huchra, and M. J. Geller (1986). Probes of large-scale structure in the corona borealis region. *The Astronomical Journal 92*, 1238–1247.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review 65*(6), 386.

Sadhanala, V., Y.-X. Wang, A. Ramdas, and R. J. Tibshirani (2019). A higher-order kolmogorov-smirnov test. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2621–2630. PMLR.

Schölkopf, B., A. J. Smola, F. Bach, et al. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Schrab, A., B. Guedj, and A. Gretton (2022). Ksd aggregated goodness-of-fit test.

Serfling, R. J. (2009). *Approximation theorems of mathematical statistics*. John Wiley & Sons.

Shao, X. (2010). The dependent wild bootstrap. *Journal of the American Statistical Association 105*(489), 218–235.

Stein, C. (1945). A two-sample test for a linear hypothesis whose power is independent of the variance. *The Annals of Mathematical Statistics 16*(3), 243–258.

Stein, C. (1972). A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the sixth Berkeley symposium on mathematical statistics and probability, volume 2: Probability theory*, Volume 6, pp. 583–603. University of California Press.

Stute, W., W. G. Manteiga, and M. P. Quindimil (1993). Bootstrap based goodness-of-fit-tests. *Metrika 40*(1), 243–256.

Sutskever, I., G. E. Hinton, and G. W. Taylor (2008). The recurrent temporal restricted boltzmann machine. *Advances in neural information processing systems 21*.

Tate, M. W. and L. A. Hyer (1973). Inaccuracy of the x2 test of goodness of fit when expected frequencies are small. *Journal of the American Statistical Association 68*(344), 836–841.

Wu, C.-F. J. (1986). Jackknife, bootstrap and other resampling methods in regression analysis. *the Annals of Statistics 14*(4), 1261–1295.

Xu, W. (2021). *Advances in Non-parametric Hypothesis Testing with Kernels*. Ph. D. thesis, UCL (University College London).

Xu, W. (2022). Standardisation-function kernel stein discrepancy: A unifying view on kernel stein discrepancy tests for goodness-of-fit. In *International Conference on Artificial Intelligence and Statistics*, pp. 1575–1597. PMLR.

Yang, Z., W. Cheng, and J. Zhang (2011). *Goodness-of-fit Test*. China Science Publishing & Media Ltd.

6 Appendix

6.1 A. Proofs

6.1.1 Proof of Lemma 2.2.1

Lemma 2.2.1 Let $\mathcal{H}$ be any Hilbert space, $\mathcal{X}$ a non-empty set and $\phi : \mathcal{X} \to \mathcal{H}$. Then a kernel function $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$ is a positive definite function. And the reverse direction also holds.

Proof. First we proof the positive direction:

$$\begin{aligned} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(x_i, x_j) &= \sum_{j=1}^{n} \langle a_i \phi(x_i), a_j \phi(x_j) \rangle_{\mathcal{H}} \\ &= || \sum_{i=1}^{n} a_i \phi(x_i) ||_{\mathcal{H}}^2 \geqslant 0 \end{aligned} \tag{6.1}$$

The other side is proofed by the the Moore-Aronszajn theorem.

6.1.2 Proof of Lemma 2.5.1

Lemma 2.5.1 Suppose p and q are two smooth densities support on $\mathcal{Z}$ and g is the stein class of p. Then the expetation of the stein operator of g under q: $\mathbb{E}_q \mathcal{A}_p g(z) = \langle f, \mathbb{E}_q \xi_z \rangle_{\mathcal{F}}$

Proof. From the Jenseon theory and Cauchy-Schwarz we have:

$$|\mathbb{E}_q \langle f, \xi_z \rangle_{\mathcal{F}}| \leqslant \mathbb{E}_q |\langle f, \xi_z \rangle_{\mathcal{F}}| \leqslant ||f||_{\mathcal{F}} \mathbb{E}_q ||\xi_z||_{\mathcal{F}} \tag{6.2}$$

Then we check whether the $||\xi_z||_{\mathcal{F}}$ is bounded. So,

$$\begin{aligned} ||\xi_z||_{\mathcal{F}}^2 &= \langle \xi_z, \xi_z \rangle_{\mathcal{F}} \\ &= \langle a, a \rangle_{\mathcal{F}} \quad \text{where } a = (\frac{d}{dz} \log p(z)) k(z, \cdot) + \frac{d}{dz} k(z, \cdot) \\ &= \left\langle \frac{d}{dz} \log p(z)) k(z, \cdot), \frac{d}{dz} \log p(z)) k(z, \cdot) \right\rangle_{\mathcal{F}} \\ &+ \left\langle \frac{d}{dz} k(z, \cdot), \frac{d}{dz} k(z, \cdot) \right\rangle_{\mathcal{F}} \\ &+ 2 \left\langle \frac{d}{dz} \log p(z)) k(z, \cdot), \frac{d}{dz} k(z, \cdot) \right\rangle_{\mathcal{F}} \end{aligned} \tag{6.3}$$

Then the result of the above three term can be calculated(Arthur Gretton's Slide 2022) as below:

$$\left\langle \frac{d}{dz} \log p(z)) k(z, \cdot), \frac{d}{dz} \log p(z)) k(z, \cdot) \right\rangle_{\mathcal{F}} = [(\frac{d}{dz} \log p(z))^2 k(z, z)] \tag{6.4}$$

where $k(z, z) = c$, c is a constant.

$$\left\langle \frac{d}{dz} k(z, \cdot), \frac{d}{dz} k(z, \cdot) \right\rangle_{\mathcal{F}} = C > 0 \tag{6.5}$$

where C is a constant.

$$\left\langle \frac{d}{dz}\log p(z))k(z,\cdot), \frac{d}{dz}k(z,\cdot)\right\rangle_{\mathcal{F}} = 0 \tag{6.6}$$

Combine the above three term we found that $||\xi_x|| = C + (\frac{d}{dz}\log p(x))^2 c$ where C, c are two constant. So $\mathbb{E}_q||\xi_z||_{\mathcal{F}} = \mathbb{E}_q\sqrt{C + (\frac{d}{dz}\log p(x))^2 c} \leqslant \sqrt{\mathbb{E}_q[C + (\frac{d}{dz}\log p(z))^2 c]}$. So that when $\mathbb{E}_q(\frac{d}{dz}\log p(z))^2 < 0$, the $||\xi_z||_{\mathcal{F}}$ is bounded. Then the Riesz theorem hold so that $\mathbb{E}_q\mathcal{A}_p g(z)$ can be written as a kernel-form.

Then the closed -form expression for KSD as follow:

$$\begin{aligned}
\mathbb{S}(p,q) &= \sup_{||g||_{\mathcal{F}}\leqslant 1} \mathbb{E}_{z\sim q}\mathcal{A}_p g(z) \\
&= \sup_{||g||_{\mathcal{F}}\leqslant 1} \mathbb{E}_{z\sim q}\left\langle g, \xi_z\right\rangle_{\mathcal{F}} \\
&= \sup_{||g||_{\mathcal{F}}\leqslant 1} \left\langle g, \mathbb{E}_{z\sim q}\xi_z\right\rangle_{\mathcal{F}} \\
&= ||\mathbb{E}_{z\sim q}\xi_z||_{\mathcal{F}}
\end{aligned} \tag{6.7}$$

6.2 Other goodness-of-fit testing

Parametric test goodness-of-fit tests assume pre-define parametric distribution or distribution family to be tested. In the literatures, two types of parametric tests is being well studied by scholars.

The first one is can be regard as *chi-squared type* goodness-of-fit test which using chi-squared type statistic for testing. This kind of tests are based on Pearson (1900) and the asymptotic distribution of their statistics usually follow a chi-squared distribution. There are some examples of 'chi-squared type' goodness-of-fit test: Cramer (1946) proposed the famous likelihood ratio test, Neyman (1949) revised the chi-squared statistics. For more 'chi-squared type' goodness-of-fit tests, see Yang et al. (2011). In general, the chi-square type statistics is a kind of ϕ-divergency where $D_f(P||Q) = \int q(x) f(\frac{p(x)}{q(x)})$ and $f = \frac{(1-u)^2}{u}, u = \frac{p(x)}{q(x)}$. (Need to rewrite here.)

The second type of parametric tests is the IPM-based goodness-of-fit test where *IPM* stand for *Integral Probability Metrics* (Müller (1997))defined as below:

Definition 2.7.1 (Integral Probability Metrics(IPM)) Suppose P and Q are two distributions with with continous densities on $X \in \mathcal{X}^d$, f is any function in the function class $\mathcal{F}$, then the IPM is define as below:

$$IPM[P,Q] = \sup_{f\in F} |E_P[f(X)] - E_Q[f(X)]| \tag{6.8}$$

The IPM is important because...

Stein operator have been applied to design the IPM for goodness-of-fit test.

We begin with a simple definition of Pearson's original Chisquare-testing.

Definition 2.7.2 (Pearson's chi-square statistic)

$$\chi_P^2 = \sum_{i=1}^{k} \frac{(n_i - np_i)^2}{np_i} \to \chi^2(k-1), \text{ (under } H_0) \tag{6.9}$$

where n_i denotes the number of observations located in the interval (a_{i-1}, a_i) $(i = 1, \ldots, k)$ in the sample $\{x_1, \ldots, x_n\} \in R^1$, $\chi^2(k-1)$ stands for the chi-square distribution with $k-1$ degrees of freedom.

Since the idea of using Chi-square test for goodness-of-fit testing was proposed (Pearson (1900)), there have been a rich source of literature and a huge amount of discussion on how to construct the better Pearson chi-square test and how to effectively apply it to practical problems. Because there exists two uncertainties in constructing Pearson-Fisher chi-square test, statisticians never stop exploring the various possibilities for improving its performance.

The first uncertainty is lead by the choice of splitting points or so-called 'grouping points'. Two common choices for grouping points is equal-number grouping points and equal-probability grouping points. However, these two types of grouping point didn't take the property of the target distribution into consideration lead to the loss of testing power(See Li et al. (2022)). As the concept of 'representative points' which used to approximate a distritbuion with empirical measure supported on a set of points is proposed, the idea of using representative points to improve Chi-square testing is came out. One kind of representative points is the MSE-representative points (See Fang and He (1988), Flury (1990)) which generated by minimizing the Mean square error between a continous distribution and its discreteization. Latter the MSE-representative points being applied to the Chi-square test and gain better testing power. (Li et al. (2022)). However, the MSE-representative points is not work for the problem with target distribution which only known up to a normalization term. Fortunately, the Stein poitns (Chen et al. (2018)) which generated by minimizing the KSD may fixs this problem. The second uncertainty of Chi-square test lead by the number of grouping intervals(or so-called 'cell'). Since Cochran (1952), it's well known that when the number of cells is small, the chi-square approximation will yields incorrect p-values. Also, regardless of sample size, the chi-square approximation cannot be used to test the overall fit of the model when the number of cells is too large(Maydeu-Olivares and Cai (2006)). This problem resulted a lot of discussions and explorations in the literature, see, for example, CoxCox (1957), HamdanHamdan (1963), Tate and HyerTate and Hyer (1973), Koehler and LarntzKoehler and Larntz (1980), Greenwood and Nikulin Greenwood and Nikulin (1996), Voinov, Nikulin, and BalakrishnanBalakrishnan et al. (2013).

For the IPM-based goodness-of-fit test, three kind of tests will be introduced: KS-Test, and the test based on Maximum Mean dicrepancy(MMD) and Kernelized Stein discrepancy(KSD).

K-S Test (Brief Intro) The Kolmogorov-Smirnov (K-S) (Kolmogorov (1933a)) test is one of the most famous IPM-type goodness-of-fit test. This kind of goodness-of-fit is also known as *empirical distribution function testing* which use the functional discrepancy between the empirical distribution F_n and population distribution F_0 for testing. Denote $F_p(z) := \Pr\{x \leqslant z \text{ for } x \sim p\}$ be the cumulative distribution and $F_X \equiv F_X(z) := \frac{1}{|X|}\sum_{i=1}^{m} 1_{z \leqslant x_i}$ is its empirical counterpart.The statistics of K-S test is $L_\infty : ||F_X - F_Y||_\infty$, for two sets of observations X and Y. The K-S test is a fast and general-purpose parametric test. Usually, the K-S test doesn't provide enough testing power for some kind of distribution, even uniform normal distribution. (See Li et al. (2022)). Another drawback of K-S test is that, it's systematically less sensetive to some type of difference such as tail difference. In Sadhanala et al. (2019), the K-S test was re-defined in a higher-order total variation ball and applied in high dimensional testing.

6.2.1 Type I error and Testing Power

Except for the choice of discrepancy m, there're two important concepts for hypothesis testing: *Type-I-error* and *Power*.

Definition 2.6.2 (Type I error and Power) The null hypothesis for a statistical hypothesis test is denoted as H_0, while the alternative hypothesis is H_1. Then the type I error is defined as $P(\text{reject } H_0|H_0)$ while the testing power: $P(\text{do not reject } H_1|H_1)$.

The Type I error owe its name by the definition of "Type II error": $P(\text{reject}H_1|H_1)$. Since people tend to be more care about the Type I erorr than Type II error, it need to be well controlled (below a given number α, usually $\alpha = 0.05$). In practice, suppose we repeat the test for m times, then type I error can be compute by $\sum_{i=1}^{m} I(\text{reject the } H_0)/m$ under null hypothesis where I is the indicate function. For testing power, we calculate by this way under alternative hypothesis: $\sum_{i=1}^{m} I(\text{do not reject the } H_1)/m$.

6.3 C. Experiment detialed

6.3.1 Kernel Choice

The choice of kernel is the crucial problem in kernel based hypothesis testing problems including relative goodness-of-fit test and two sample test. That is, a particular kernel may work well for some set of observation sample while not being so useful for the others. In kernel-based goodness-of-fit test, the testing power is being strongly affected by the choice of kernel.

A paper [Xu (2021)] pointed out that, the KSD-based goodness-of-fit testing usually achieves low testing power when a translation invariant kernel is applied. Also, it's not robust when the target distribution is multi-modal. While using this kind of simple kernels, they usually map the different distributions to nearby mean embeddings so that it's hard to distinguish the difference between them.

In this area, the first paper[Gretton et al. (2012)] suggested using a heuristic method which choosing the median Euclidean distance between data as the bandwidth for Radial basis function kernel (such as Gaussian kernel). Later in [Gretton et al. (2012)], the method of using linear combinations of kernels is proposed. Reccently, Schrab et al. (2022) proposed a new method that combine multiple kernel functions together to achieve the SOTA performance of goodness-of-fit testing.

For some specific problem that related to the big data (such as images generation), the deep kernel [Xu (2021)] could be applied. The basic idea of deep kernel is using tranning data to learn the parameters of a pre-defined parametric kernel function. This idea is quite similar to the trainning process of the neural network.

Since our paper is not focus on the choice of kernel, we use the heuristic method in Gretton et al. (2012). To improve the performance of KCR-GOF, the more complicated kernel method could be applied.

6.3.2 Closed-form expression for KSD estimator

In section 4, all the models we use are belongs to exponential family. Here we explain how to ues the minimum-KSD estimator [Barp et al. (2019)] for this kind of model.

For kernelized and non-kernelized exponential models, they share a common form of probabililty density as below:

$$p_\theta = \exp(\eta(\theta) \cdot t(x) - \alpha(\theta) + b(x)) \tag{6.10}$$

where η is the invertible map (map the natural paramter to the parameter that we want, such as the shift-parameter in laplace distribution). t is the sufficient statistic. a is the transformation function with parameter θ. b is the transformation function with intput data x.

According to Key et al. (2021), the KSD estimator is given by

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \hat{\mathbb{S}}_u(\mathbb{P}_\theta || X_n) = -\frac{1}{2}(\Lambda_n^{-1} \upsilon_n) \tag{6.11}$$

where $\Lambda_n = \frac{1}{n^2} \sum_{i=1}^n \Lambda(x_i, x_j)$, and $\upsilon_n = \frac{1}{n^2} \sum_{i=1}^n \upsilon(x_i, x_j)$ with functions

$$\begin{aligned} &\Lambda(x,y) = k(x,y) \nabla t(x) \nabla t(y)^T \\ &\upsilon(x,y) = k(x,y) \nabla b(x) \nabla t(y)^T + \nabla t(x)^T \nabla_y k(x,y) + k(x,y) \nabla b(y) \nabla t(x)^T + \nabla t(y)^T \nabla_x k(x,y) \end{aligned} \tag{6.12}$$

6.4 Experiment setting in section 4.1

6.5 Experiment setting in section 4.2