A Kernel Stein Test of Goodness of Fit for Sequential Models

Jerome Baum [*1] **Heishiro Kanagawa** [*2 3] **Arthur Gretton** [2]

Abstract

We propose a goodness-of-fit measure for probability densities modeling observations with varying dimensionality, such as text documents of differing lengths or variable-length sequences. The proposed measure is an instance of the kernel Stein discrepancy (KSD), which has been used to construct goodness-of-fit tests for unnormalized densities. The KSD is defined by its Stein operator: current operators used in testing apply to fixed-dimensional spaces. As our main contribution, we extend the KSD to the variable-dimension setting by identifying appropriate Stein operators, and propose a novel KSD goodness-of-fit test. As with the previous variants, the proposed KSD does not require the density to be normalized, allowing the evaluation of a large class of models. Our test is shown to perform well in practice on discrete sequential data benchmarks.

1. Introduction

This paper addresses the problem of evaluating a probabilistic model for observations with variable dimensionality. This problem commonly arises in sequential data analysis, where sequences of different lengths are observed, and generative models (e.g., Markov models) are used to draw inferences. Our problem setting also concerns a scenario where a data point is a collection of observations that may not be sequentially ordered, as in the topic modeling of text documents. Our task is formalized as follows: given a sample $\{X_i\}_{i=1}^n$ from a distribution Q on a sample space $\mathcal{X}$ (e.g., the set of all possible sequences), we aim to quantify the discrepancy of distribution P modeling Q.

One approach to this problem is generating samples from the model and computing a sample-based discrepancy measure, such as the maximum mean discrepancy (MMD) (Gretton et al., 2012; Lloyd & Ghahramani, 2015). A disadvantage of this approach is that it potentially requires many samples to see the departure of the model from the data. For example, if a sequence model misspecifies a particular transition from a given history, we would need to generate sequences sharing this history to observe the disparity. Unfortunately, the MMD approach becomes less efficient as the state space enlarges or the length of the history grows, necessitating repeated sampling. This observation motivates using a measure that exploits information provided by the model, such as dependence relations between different time points.

A model-dependent measure may be derived based on Stein's method (Stein, 1972), a technique from probability theory developed originally to obtain explicit rates of convergence to normality. The key construct from Stein's method is a distribution-specific operator called a Stein operator that modifies a function to have zero expectation under the distribution. A model-specific Stein operator $\mathcal{A}_P$ may be defined to construct a zero-expectation function $\mathcal{A}_P f$; its expectation $\mathbb{E}_{X\sim Q}[\mathcal{A}_P f(X)]$ under the sample distribution serves as a discrepancy measure, since a non-zero expectation indicates the deviation from the model. One may generalize this idea to a family of functions $\mathcal{F}$, and the resulting worst-case summary $\sup_{f\in\mathcal{F}}|\mathbb{E}_{X\sim Q}[\mathcal{A}_P f(X)]|$ is called a Stein discrepancy, proposed by Gorham & Mackey (2015). An appropriate choice of the operator and the function class yields a computable Stein discrepancy. This paper focuses on an instance called the kernel Stein discrepancy (KSD) (Oates et al., 2016; Chwialkowski et al., 2016; Liu et al., 2016; Gorham & Mackey, 2017), where functions from a reproducing kernel Hilbert space (RKHS) are used.

The KSD is a versatile framework for designing goodness-of-fit measures. Given a Stein operator and a reproducing kernel, the KSD is expressed by an expectation of a Stein-modified kernel, leading to tractable estimators only involving sample evaluations of the modified kernel. This feature allows us to use a wealth of kernels from the literature: by designing a Stein operator, one can adapt the kernel to the model and define a bespoke discrepancy measure. Based on the zero-mean kernel theory by Oates et al. (2016), Chwialkowski et al. (2016) and Liu et al. (2016) originated this line of work, where they combined the Langevin Stein

[*]Equal contribution [1]Department of Computer Science, University College London, UK [2]Gatsby Computational Neuroscience Unit, UCL [3]School of Mathematics, Statistics and Physics, Newcastle University, UK. Correspondence to: Jerome Baum <jerome@jeromebaum.com>, Heishiro Kanagawa <heishiro.kanagawa@gmail.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

operator proposed by Gorham & Mackey (2015) with an RKHS; remarkably, the resulting (Langevin) KSD is computable for densities with unknown normalizing constants. There have been numerous extensions to models in other data domains: categorical data (Yang et al., 2018), point-pattern data (Yang et al., 2019), censored data (Fernandez et al., 2020), directional data (Xu & Matsuda, 2020), and functional data (Wynne et al., 2022). We refer the reader to Anastasiou et al. (2021) for a review on the KSD's applications outside goodness-of-fit testing.

A limitation of the preceding Stein works is that they require the model distribution to be defined on a fixed-dimension space. For example, Kanagawa et al. (2023) evaluate latent Dirichlet allocation models (Blei et al., 2003) for text documents assuming a fixed document length, an assumption unlikely to hold in practice. Relevant work has been accomplished by Wynne et al. (2022), where they propose a KSD for distributions on an (infinite-dimensional) Hilbert space, and treat continuous-time models, such as Brownian motions and stochastic differential equation models. While their test applies to sequential models, our target setting is different, as we focus on discrete-time models such as Markov chains, and do not require a density with respect to a Gaussian measure.

In the present work, we extend the KSD framework to the variable-dimension setting. Specifically, we identify a Stein operator for distributions on the set of all univariate sequences of finite alphabets. This is to our knowledge the first such operator, and it may of independent interest. Our Stein operator builds on the class of Zanella-Stein operators recently proposed by Hodgkinson et al. (2020), which are derived from a Markov jump process having the target distribution as invariant measure; we review the Zanella-Stein operator in Section 2. In Section 3, we derive a new Stein operator using a Markov process admitting transitions between sets of different sizes. Based on our Stein operator and the associated KSD, we propose a novel goodness-of-fit test for sequential models. As in previous variants, the proposed KSD does not require the model to be normalized, allowing the evaluation of *intractable models*, including Markov random fields (Section 4.5) and conditional generative models (Section 4.6). As the proposed operator involves a number of tunable parameters, as our second contribution, we offer guidance on how to select these based on empirical power analysis.

2. Background

All Stein discrepancies are build around a particular Stein operator. In this section, we recall a class of operators, the Zanella-Stein operators, and describe the challenges in constructing operators of this class in the sequential setting. We also review the associated kernel Stein discrepancy (KSD), a goodness-of-fit measure that leverages a given Stein operator to construct a test statistic.

Zanella-Stein Operator The Zanella-Stein (ZS) operators are a class of Stein operators for distributions on a countable set $\mathcal{X}$, proposed by Hodgkinson et al. (2020) following the generator method of Barbour (1988). In the following, we assume that the model P has probability mass function p positive everywhere in $\mathcal{X}$. For a real-valued function f on $\mathcal{X}$, a ZS operator $\mathcal{A}_P$ is defined by

$$(\mathcal{A}_P f)(x) \coloneqq \sum_{y \in \partial x} g\left(\frac{p(y)}{p(x)}\right)(f(y) - f(x)), \quad (1)$$

where $\partial x \subset \mathcal{X}$ is a set of *neighborhood* points, and $g : (0, \infty) \to (0, \infty)$ is a *balancing function*, which must satisfy $g(t) = tg(1/t)$ for $t > 0$. Several choices of g are known in the literature: the minimum probability flow operator discussed in (Barp et al., 2019) uses $g(t) = \sqrt{t}$; the choice $g(t) = t/(1+t)$ is known as the Barker Stein operator (Barker, 1965; Shi et al., 2022).

A Zanella-Stein operator is the infinitesimal generator of a Markov jump process called a Zanella process (Zanella, 2019; Power & Goldman, 2019) designed to have the target distribution P as an invariant distribution. The process is based on the idea of *locally informed proposals*: these jump from a given point x to any of its neighbors y, at a rate $g(p(y)/p(x))$ so that detailed balance is satisfied for P, ensuring that P is an invariant distribution. For the operator to characterize P, we must specify our notion of neighborhood. Let $(\mathcal{V}, \mathcal{E})$ be the directed graph with vertices $\mathcal{V} = \mathcal{X}$ and edges $\mathcal{E} = \{(x, y) | y \in \partial x\}$ induced by the chosen neighborhood structure. The graph $(\mathcal{V}, \mathcal{E})$ must have the following properties:

1. Symmetry: $(x, y) \in \mathcal{E} \Leftrightarrow (y, x) \in \mathcal{E}$.
2. Strong connectivity: the transitive closure of $\mathcal{E}$ is $\mathcal{X} \times \mathcal{X}$, so that there is a path from every point to every other point.

In fact, symmetry alone implies that P is in the set of invariant distributions; strong connectivity is required to ensure that P is the unique invariant distribution, as otherwise the corresponding Stein discrepancy cannot distinguish P from other distributions even for a rich test function class. In Hodgkinson et al. (2020), the aperiodicity condition is additionally required. For a pure jump-type process, we only need the irreducibility of the process (Kallenberg, 2021, Theorem 13.12), and hence we may dispense with this condition. Note that we can conveniently choose a sparse neighborhood to accelerate the computation, although this may result in reduced sensitivity of the discrepancy. In Section 4, we discuss in more detail the tradeoff between computational cost and test power.

Challenges in the Variable-Dimension Setting The generator method (as in the ZS operator above) allows us to obtain an operator for any state space with an appropriate Markov process. Indeed, the Zanella process is not the only allowable choice: other Markov chains or processes have also been considered for discrete spaces, including birth-death processes (Shi et al., 2022), the Glauber dynamics Markov chain (Reinert & Ross, 2019; Bresler & Nagaraj, 2019) (we refer the reader to Shi et al., 2022, for a review). Defining a Markov process is often a challenge, however; prior work has thus only considered processes in fixed-dimensional spaces, and has not dealt with sequential models such as those considered here. The ZS operators are particularly attractive in our setting, since they may be defined on any discrete set. The main challenge in deriving an operator in this class is the construction of an appropriate neighborhood for models of variable-length sequences, which is highly nontrivial. Our contribution is to develop an effective neighborhood definition for the sequential setting, which satisfies the three requirement outlined above, and yields an associated test that has good statistical power against alternatives.

Kernel Stein Discrepancy We next review the construction of a test statistic from the Stein operator. A computable discrepancy measure may be defined using a reproducing kernel Hilbert space (RKHS) (Aronszajn, 1950; Steinwart & Christmann, 2008). Following (Oates et al., 2016; Chwialkowski et al., 2016; Liu et al., 2016; Gorham & Mackey, 2017), we define the kernel Stein discrepancy (KSD) associated with our (yet to be specified) Zanella-Stein operator as follows:

$$\mathrm{KSD}\big[Q\|P\big] \coloneqq \sup_{\|f\|_{\mathcal{H}}\leq 1} \big|\mathbb{E}_{X\sim Q}[\mathcal{A}_P(f)(X)]\big|, \qquad (2)$$

where $\mathcal{H}$ is the RKHS of real-valued functions on $\mathcal{X}$ corresponding to a positive definite kernel $k : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ with the natural norm $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f\rangle_{\mathcal{H}}}$ defined by the inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$. By the vanishing property $\mathbb{E}_{Y\sim P}[\mathcal{A}_P f(Y)] = 0$, the KSD is zero if $Q = P$. The other direction requires further assumptions on the operator and the RKHS. According to Hodgkinson et al. (2020, Proposition 4), when the corresponding Zanella process is exponentially ergodic and the RKHS is C_0-universal, we have $\mathrm{KSD}\big[Q\|P\big] = 0$ only if $P = Q$. For a finite state space $\mathcal{X}$, exponential ergodicity is satisfied by any irreducible Markov jump process. The C_0-universality is equivalent to the integrally strict positive definiteness (Sriperumbudur et al., 2011); in the particular case $|\mathcal{X}| < \infty$, this condition states that the Gram matrix defined over all the configurations on $\mathcal{X}$ is strictly positive definite.

The use of an RKHS yields a closed-form expression of the KSD (Hodgkinson et al., 2020, see Proposition 1 and the paragraph following the proof):

$$\mathrm{KSD}^2\big[Q\|P\big] = \mathbb{E}_{X,X'\sim Q\otimes Q}[h_p(X,X')], \qquad (3)$$

where X, X' are i.i.d. random variables with law Q, provided $\mathbb{E}_{X\sim Q}[h_p(X,X)^{1/2}] < \infty$. The proof of this result follows as in (Gorham & Mackey, 2017, Proposition 2), noting that the inner product between f and $\mathcal{A}_P k(x,\cdot)$ *reproduces* $\mathcal{A}_P f(x)$ for any $f \in \mathcal{H}$. The function h_p is called a *Stein kernel*, defined by

$$\begin{aligned} &h_p(x,y) \\ &= \sum_{\nu\in\mathcal{N}_x}\sum_{\tilde{\nu}\in\mathcal{N}_y} g_\nu(x) g_{\tilde{\nu}}(y)\,\{k(\nu(x),\tilde{\nu}(y)) + k(x,y) \\ &\qquad\qquad -k(x,\tilde{\nu}(y)) - k(\nu(x),y)\}, \end{aligned}$$

where we identify each point in the neighborhood ∂x as a mapping from x to itself, and $g_\nu(x) = g\{p(\nu(x))/p(x)\}$.

Given a sample $\{X_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} Q$, the squared KSD expression (3) admits a simple unbiased estimator

$$U_{n,P}\left(\{X_i\}_{i=1}^n\right) \coloneqq \frac{1}{n(n-1)}\sum_{i\neq j}^{n} h_p(X_i,X_j), \qquad (4)$$

which is a U-statistic (Hoeffding, 1948). By the zero-mean property of the Stein kernel, the U-statistic is degenerate if $Q = P$ (Chwialkowski et al., 2016; Liu et al., 2016). In this case, the scaled statistic $nU_{n,P}\left(\{X_i\}_{i=1}^n\right)$ asymptotically follows the law of $\sum_{j=1}^\infty \lambda_j(Z_j^2 - 1)$, where Z_j are i.i.d. standard normal variables, and λ_j are eigenvalues given by the eigenvalue problem $\sum_{x'\in\mathcal{X}} h_p(x,x')a(x')p(x') = \lambda a(x)$ with $\sum_{x\in\mathcal{X}} a(x)^2 p(x) < \infty$ (see, e.g., Serfling, 2009, Section 5.5). One of our proposed tests in Section 3 simulates this distribution using a wild bootstrap procedure.

Kernels for Sequences The performance of the test developed in the next section depends on the choice of the reproducing kernel. As mentioned above, the C_0-universality is a condition that guides kernel selection since it renders the test consistent (along with the exponential ergodicity). A trivial example of C_0-universal kernels is the Dirac kernel that outputs 1 if two inputs are identical, and 0 otherwise; but this kernel might not be useful in practice, since all data points in the corresponding feature space are orthogonal, and provide no information about each other. As this example shows, kernels need not be universal to provide powerful tests, as long as they encode relevant features to the setting where they are used. The literature on sequence kernels is well-established (see, e.g., Király & Oberhauser, 2019, for a recent account), and one could use existing kernels, such as global alignment kernels (Cuturi et al., 2007; Cuturi, 2011) and string kernels (Haussler, 1999; Lodhi et al., 2002; Leslie & Kuang, 2004). Alternatively, one may also define a kernel by the inner product of explicit features (e.g.,

neural network embedding of sequences) instead of known kernels.

3. Testing Sequential Models

We now address the design of a Stein operator for the variable-dimension setting. We begin by formally defining the sample space $\mathcal{X}$. Let $\mathcal{S}$ be a nonempty finite set of symbols. For an integer $\ell \geq 1$, we denote by $\mathcal{S}^{(\ell)}$ the set of all length-ℓ sequences formed by symbols in $\mathcal{S}$; i.e., $\mathcal{S}^{(\ell)} = \prod_{j=1}^{\ell} \mathcal{S}$. Then, the sample space $\mathcal{X}$ is given by the set of all sequences $\bigsqcup_{\ell=1}^{\ell_{\max}} \mathcal{S}^{(\ell)}$, where $\bigsqcup$ denotes the disjoint union, and $\ell_{\max}$ is ∞ or a finite positive integer. In this setting, a random sample in $\mathcal{X}$ is a sequence whose length is randomly determined.

As noted in Section 2, to obtain a KSD in this setting, we need to design a neighborhood structure which specifies our ZS operator. We first introduce a simple structure for sequences, which edits only the end of the string, to illustrate the required connectivity principles. We then describe a more advanced neighborhood structure, allowing for greater modifications, which will be the foundation of our tests.

Neighborhood Choice For the Stein operator to uniquely characterize the model, we require the strong connectivity of the induced graph. We achieve this by introducing *inter- and intra-length state transitions*. Specifically, for a length-ℓ sequence $x_{(1:\ell)} = (x_1, \ldots, x_\ell) \in \mathcal{S}^{(l)}$, we propose the following neighborhood:

$$\partial x_{(1:\ell)} = \mathcal{I}_x \cup \{x_{(1:\ell-1)}\} \cup \mathcal{R}_x \tag{5}$$

with

$$\mathcal{I}_x = \{(x_1, \ldots, x_\ell, s) : s \in \mathcal{S}\},$$
$$\mathcal{R}_x = \{(x_1, \ldots, x_{l-1}, s) : s \in \mathcal{N}_{\mathrm{rep}}(x_\ell)\},$$

where $\mathcal{N}_{\mathrm{rep}}(s)$ denotes a neighborhood chosen for a symbol $s \in \mathcal{S}$; this is chosen such that the induced graph is strongly connected. The first two sets in (5) represent the *inter-length transitions*: Each point of $\mathcal{I}_x$ corresponds to adding a symbol to the end of the given sequence $x_{(1:\ell)}$, whereas $\{x_{(1:l-1)}\}$ stands for the deletion of the last symbol and shortening the sequence. The third set $\mathcal{R}_x$ represents *intra-length transitions*, where the last symbol of the sequence is replaced. A trivial choice for satisfying the strong connectivity is using the whole alphabet set $\mathcal{S}$ for $\mathcal{N}_{\mathrm{rep}}(s)$ for each $s \in \mathcal{S}$. This approach may be permitted when the alphabet size is small. One might otherwise want to consider using smaller sets to reduce the computational cost. For example, we can use a smaller $\mathcal{N}_{\mathrm{rep}}(s)$, provided that the graph (with vertices $\mathcal{S}$) induced by the choice satisfies the three conditions required in Section 2; e.g., as in (Yang et al., 2018), we may introduce a cyclic order $\{0, \ldots, |\mathcal{S}| - 1\}$ to $\mathcal{S}$, and take two adjacent points $\mathcal{N}_{\mathrm{rep}}(s) = \{s+1, s-1\}$, where $s \pm 1$ denotes $(s \pm 1) \bmod |\mathcal{S}| - 1$.

The structure proposed above is a minimal neighborhood choice that guarantees strong connectivity. Indeed, we can edit the length of a sequence, and change the character in each position to a desired one because of the strong connectivity in the intra-length transition. As we will see in Section 4, however, this minimal choice may not produce a powerful test, as it only uses information of neighboring length sequences. For example, the tail insertion operation by $\mathcal{I}_x$ only encodes the structure of the target distribution in terms of the end of the sequence. For sequential models such as Markov chain models, the longer a sequence is, the more evidence we should have in favor of, or against, a given model. Editing only the final element of a sequence would therefore not make use of such accumulated evidence. Indeed, this approach performs relatively weakly in our benchmark experiments (Figure 2b, case of $J = 1$).

Proposed Neighborhood Based on the above observation, we generalize the above neighborhood set. Specifically, we consider expanding the neighborhood set by modifying sequence elements at different locations: we propose the J-location modification neighborhood

$$\partial^J_{x_{(1:\ell)}} = \cup_{j=1}^{J} \big(\mathcal{I}_{x_{(1:\ell)},j} \cup \mathcal{D}_{x_{(1:\ell)},j} \cup \mathcal{R}_{x_{(1:\ell)},j}\big), \tag{6}$$

where

$$\mathcal{I}_{x_{(1:\ell)},j} \subset \mathcal{S}^{(\ell+1)} = \{\mathrm{ins}(x_{(1:\ell)}, j, s) : s \in \mathcal{N}_{\mathrm{ins}}\},$$
$$\mathcal{D}_{x_{(1:\ell)},j} \subset \mathcal{S}^{(\ell-1)} = \{\mathrm{del}(x_{(1:\ell)}, j) \text{ if } x_{\ell-j+1} \in \mathcal{N}_{\mathrm{ins}}\},$$
$$\mathcal{R}_{x_{(1:\ell)},j} \subset \mathcal{S}^{(\ell)} = \{\mathrm{rep}(x_{(1:\ell)}, j, s) : s \in \mathcal{N}_{\mathrm{rep}}(x_\ell)\}.$$

Here, $\mathrm{ins}(x_{(1:\ell)}, j, s)$ denotes the sequence extending $x_{(1:\ell)}$ by inserting s at the $(j-1)$-th location counting back from the end of the string, using symbols from a fixed set $\mathcal{N}_{\mathrm{ins}} \subset \mathcal{S}$; $\mathrm{del}(x_{(1:\ell)}, j)$ deletes the $(j-1)$-th element counting back from the end of $x_{(1:\ell)}$; and $\mathrm{rep}(x_{(1:\ell)}, j, s)$ replaces by s the $(j-1)$-th element from the end of $x_{(1:\ell)}$. Note that the deletion and insertion operations must be paired to ensure the symmetry of the graph. As with the substitution set $\mathcal{N}_{\mathrm{rep}}$, we may choose $\mathcal{N}_{\mathrm{ins}}$ such that it depends on the point x; the resulting neighborhood requires additional care to maintain symmetry and (optionally) strong connectivity. See Section 4.6 for an example.

Test Procedure Using one of the proposed neighborhoods for the Zanella-Stein operator, we can define a KSD goodness-of-fit test for sequential models. Given an i.i.d. data $\{X_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} Q$, we consider testing the null $H_0 : \mathrm{KSD}\big[Q\|P\big] = 0$ against the alternative $H_1 : \mathrm{KSD}\big[Q\|P\big] \neq 0$. Note that when the KSD distinguishes

Algorithm 1 Parametric bootstrap test

Require: P; a target distribution on $\mathcal{X}$.
Require: $D_n = \{X_1, \ldots, X_n\}$; data.
Require: α; the desired level of the test.
for $b = 1, \ldots, B$ **do**
  Sample $D_n^{(b)} = \{X_1^{(b)}, \ldots, X_n^{(b)}\} \overset{\text{i.i.d.}}{\sim} P$
  $\Delta_b \leftarrow U_{n,P}\left(D_n^{(b)}\right)$
end for
$\hat{t}_\alpha \leftarrow (1-\alpha)$-quantile of $\{\Delta_1, \ldots, \Delta_B\}$
if $U_{n,P}(D_n) > \hat{t}_\alpha$ **then**
  Reject H_0
end if

Algorithm 2 Wild bootstrap test

Require: P; a target distribution on $\mathcal{X}$.
Require: $D_n = \{X_1, \ldots, X_n\}$; data.
Require: α; the desired level of the test.
for $b = 1, \ldots, B$ **do**
  Draw $W_b \sim \mathrm{Multi}(n; 1/n, \ldots, 1/n)$
  $\Delta_b \leftarrow U_{n,P}^{W_b}(D_n)$
end for
$\hat{t}_\alpha \leftarrow (1-\alpha)$-quantile of $\{\Delta_1, \ldots, \Delta_B\}$
if $U_{n,P}(D_n) > \hat{t}_\alpha$ **then**
  Reject H_0
end if

any distributions, the null and alternative become $H_0 : P = Q$ and $H_1 : P \neq Q$, respectively. Below, we present two tests, which differ in how they compute test thresholds.

The first test is the parametric bootstrap test in Algorithm 1. The parametric bootstrap repeatedly draws samples from the model P and simulates the distribution of the statistic (4). Because of this feature, this procedure does not apply if sampling from the model is infeasible.

The second test is the wild bootstrap test described in Algorithm 2, which is analogous to the existing KSD tests (Chwialkowski et al., 2016; Liu et al., 2016; Yang et al., 2018). As we have seen in Section 2, the asymptotic distribution of the (scaled) KSD estimate (4) is known. The wild bootstrap test simulates this distribution by repeatedly computing the following quantity:

$$U_{n,P}^{W}(D_n) := \frac{\sum_{i \neq j}^{n} (W_i - 1)(W_j - 1) h_p(X_i, X_j)}{n(n-1)} \quad (7)$$

where $W = (W_1, \ldots, W_n)$ is a sample from a multinomial distribution $\mathrm{Multi}(n; 1/n, \ldots, 1/n)$ with n trials and n events and independent of the data $D_n = \{X_1, \ldots, X_n\}$. The test has an only asymptotically correct size (Dehling & Mikosch, 1994, Theorem 3.1), and may not be suitable for a small sample size n. In contrast to the parametric bootstrap test, this test does not require sampling from the model, and only needs single evaluation of the Stein kernel h_p over the data. Treating the evaluation of the Stein kernel as constant, the test's computational complexity is $O(n^2)$, since the computation of the KSD statistic is $O(n^2)$; this computation can be paralellized.

4. Experiments

We investigate the performance of the proposed test through synthetic experiments with defined ground truths.[1] In particular, we aim to answer the following questions: (a) how leveraging model structure using a Stein kernel improves test performance over the maximum mean discrepancy (MMD) (Gretton et al., 2012, a purely sample-based kernel discrepancy where no model information is used); (b) the effect of neighborhood choice on the KSD test power. In the following, we use the Barker balancing function $g(t) = t/(1+t)$ to define the KSD.

Throughout our experiments, we will make use of two simple kernels. The first is the exponentiated Hamming kernel, $k_{\mathrm{H}}(x_{(1:\ell)}, y_{(1:\ell)}) := \exp\{-\ell^{-1} \sum_{i=1}^{\ell} \delta_{x_i}(y_i)\}$, where $\delta_x(y) = 1$ if $x = y$ and 0 otherwise. Additionally, we define $k_{\mathrm{H}}(x_{(1:\ell)}, y_{(1:\ell')}) = 0$ whenever $\ell \neq \ell'$. The second is a string kernel which we refer to as the *contiguous subsequence kernel* (CSK). This kernel counts the number of contiguous subsequences of a particular length, present in the two input sequences,

$$k_{\mathrm{CSK}}(x, y) := \frac{k_{\mathrm{u}}(x, y)}{\sqrt{k_{\mathrm{u}}(x, x) \cdot k_{\mathrm{u}}(y, y)}},$$

where

$$k_{\mathrm{u}}(x_{(1:\ell)}, y_{(1:\ell')}) := \sum_{i=1}^{\ell-t+1} \sum_{j=1}^{\ell'-t+1} \delta_{x_{(i:i+t-1)}}\left(y_{(j:j+t-1)}\right),$$

with parameter t for the subsequence length. We use the normalized version so that the kernel does not overly depend on the sequence lengths. The Hamming kernel does not take into account the sequence structure and can be considered as a naive choice, whereas the CSK kernel does.

4.1. Demonstration of Test Power

We first show that the KSD indeed benefits from information supplied by the model. For this purpose, we construct a problem where the alternative is given by adding a slight perturbation to the model. We choose $\mathcal{S} = \{0, \ldots, m-1\}$ for our alphabet, with $m = 8$. The model distribution is a random

[1] The code is available at https://github.com/test-for-sequential-models/code

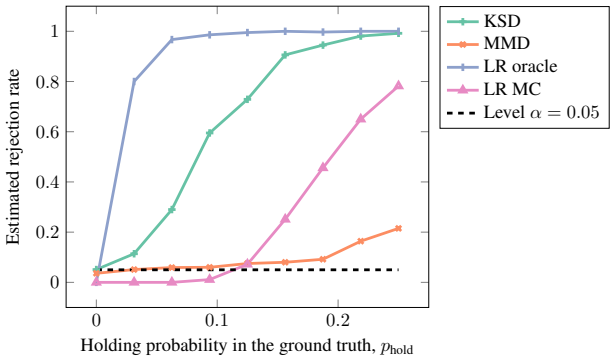


Figure 1: Estimated rejection rate (test power) of the proposed test when the model distribution and ground truth differ by a perturbation that is unlikely to be observed in any given sample.

walk through $\mathcal{S}$, starting from a random uniform element of $\mathcal{S}$, with increments drawn uniformly from $\{-1, +1\}$. We use a cyclic structure, where we identify 0 with m such that the random walk wraps around. The chain terminates with probability $1/20$ after each step. We construct an alternative distribution as follows: At every step, if and only if the current element is $0 \in \mathcal{S}$, we randomly replace the increment with 0 with a probability $p_{\text{hold}} \in [0, 1/4]$. That is, we introduce a random holding step that occurs on average in $1/(4m) = 1/32$ of steps, when the perturbation parameter is at its maximum. The perturbation is unlikely to be observed in any given sample, making the problem challenging.

For the KSD test we use the J-location modification neighborhood (6) with $J = \infty$; i.e., we allow edits anywhere in the sequence. As baselines we use an MMD test that draws 100 samples from the model, a likelihood-ratio test where the alternative consists of all first-order Markov chains on $\mathcal{S}$, and an oracle likelihood-ratio test where the alternative is the (generally unknown) ground truth distribution. Both kernel tests are built using the CSK kernel. We use parametric bootstrap for all four tests. Tests are evaluated on $n = 30$ i.i.d. sample points from the perturbed distribution.

In Figure 1 we can see that the KSD test outperforms the two non-oracle baselines. At low degrees of perturbation, the problem is challenging because perturbations are rarely observed. The KSD test is able to overcome this difficulty by exploiting information provided by the model.

4.2. Choice of Neighborhood Graph

Our test in the preceding experiment performed well, in part due to a good choice of neighborhood. We now further investigate the choice of neighborhood for the Zanella-Stein operator. In order to compare the performance of different choices of Stein kernel, we constructed 12 synthetic testing scenarios with various properties, ensuring our conclusions are broadly applicable, and not specific to properties of a particular model. The experiments in this section will evaluate the average performance of each test across these scenarios. We use discrete-time Markov chains for the model P and ground truth Q. The testing scenarios are fully specified in Appendix A.3 (see Appendix A.4 for individual results).

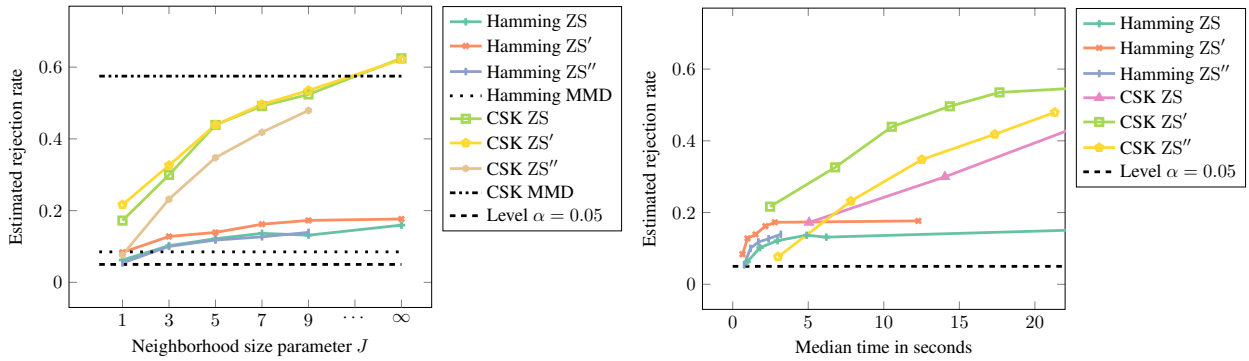
More densely connected neighborhood graphs should yield more powerful tests, but such tests are slower to evaluate. To understand the tradeoff between compute budget and test power, we construct a family of KSD tests parameterized by the size of the neighborhood of each point. We consider three classes of neighborhood: (a) the J-location modification neighborhood $\partial^J_{x_{(1:\ell)}}$, (b) the substitution neighborhood $\cup_{j \leq J} \mathcal{R}_{x_{(1:\ell)},j}$, and (c) the neighborhood $\cup_{j \leq J} (\mathcal{I}_{x_{(1:\ell)},j} \cup \mathcal{D}_{x_{(1:\ell)},j})$ of insertions and deletions. In each case, the symbol neighborhoods $\mathcal{N}_{\text{ins}}$ and $\mathcal{N}_{\text{rep}}$ are set to the entire alphabet $\mathcal{S}$. We denote the resulting Stein operators by ZS, ZS$'$, and ZS$''$, respectively. We construct KSDs by applying these operators on the two kernels specified earlier. To estimate the critical value, we use parametric bootstrap. We also include a pair of MMD tests, one for each kernel, as a baseline.

Results are in Figure 2a. Larger neighborhoods lead to more powerful tests, as expected. However, Figure 2b shows that we face diminishing returns when trading off compute budget against test power. The CSK kernel always does better than the Hamming kernel. Interestingly, for the CSK kernel, the simpler ZS$'$ test performs slightly better than the ZS test, since in this instance, the larger neighbourhood of the latter does not yield significant additional information, while entailing a small increase in variance of the test statistic. Both of these Stein tests outperform the CSK MMD for $J = \infty$.

Another approach to controlling the size and connectivity of the neighborhood is to control the symbol neighborhoods $\mathcal{N}_{\text{ins}}$ and $\mathcal{N}_{\text{rep}}$. We consider the operator ZS$'$ (a neighborhood consisting of substitutions) with a cyclic structure over the alphabet $\mathcal{S}$ and the symbol neighborhood $\mathcal{N}_{\text{rep}}(s)$ limited by the distance in the cyclic alphabet. In Figure 3 we can see no notable impact of symbol neighborhood on test power. This implies that we may construct tests using a sparse neighborhood. The experiment also shows that changes to the size of the neighborhood are not sufficient to fully explain our previous results. Test power is evidently also impacted by the location of edits rather than just the number.

4.3. Use of Evidence in Long Sequences

Edits across the entire sequence lead to higher test power. One explanation for this effect is that editing the entire sequence, instead of just editing near the end, allows the test to make better use of the evidence available from longer



(a) Rejection rate against the neighborhood size J

(b) Rejection rate against median computation time

Figure 2: Estimated rejection rate of three different families of Zanella-Stein test with varying neighborhood size and underlying kernel. We evaluate the tests on 12 different synthetic testing scenarios, evaluating each test multiple times on independently sampled synthetic datasets. To estimate the overall rejection rate, we report a simple average across all test evaluations. Test power increases as we increase the neighborhood size. We see diminishing returns particularly when the Hamming kernel is used to construct the test.

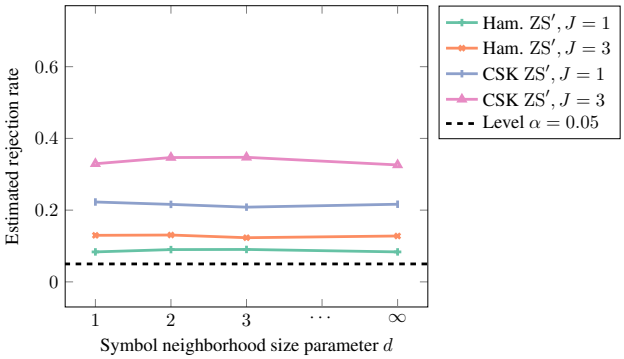


Figure 3: Estimated rejection rate of several Zanella-Stein tests based on the substitution neighborhood. We control the size of the symbol neighborhood $\mathcal{N}_{\text{rep}}$ and estimate rejection rate as in Figure 2.

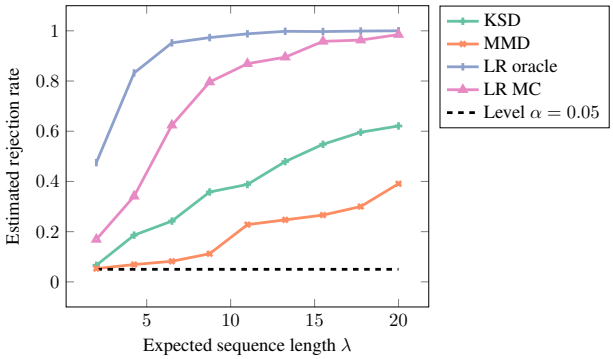


Figure 4: Estimated rejection rate of the proposed test, where we control the expected length of the samples, while holding the sample count fixed at 8. As the expected length increases, test power grows.

sequences. Where the data are distributed according to a Markov chain or some other model that mixes quickly, longer sequences provide more information about the distribution. We now show that the proposed test can indeed make use of such information.

In order to illustrate this feature, we construct the following problem: The model distribution and ground truth are second-order Markov chains over a finite alphabet with $|\mathcal{S}| = 8$. We select two transition kernels randomly from a Dirichlet distribution with concentration parameter $\alpha = 1$. The model distribution uses one of these two kernels, whereas the ground truth uses an even mixture of the two kernels, so that at each step each of the kernels is used with equal probability. The two chains both stop with a fixed probability $1/\lambda$ after each step. We fix the sample size n at 8 and vary the stopping probability in order to control the length of the samples.

The KSD test for this experiment is constructed using a large, strongly connected neighborhood: specifically, the J-location modification neighborhood with $J = \infty$, which allows insertion, deletion, and substitution anywhere in the sequence. As a baseline, we compare against an MMD test using 100 samples from the model distribution, as well as two likelihood-ratio tests: an oracle test, and a test where the alternative consists of all second-order Markov chains over $\mathcal{S}$. The two kernel tests are constructed using the CSK kernel with subsequence length $t = 3$, and we use parametric bootstrap for all tests.

The results are shown in Figure 4. We can see that the test is able to reject more often on longer sequences, without the need for more samples, and outperforms the MMD test.

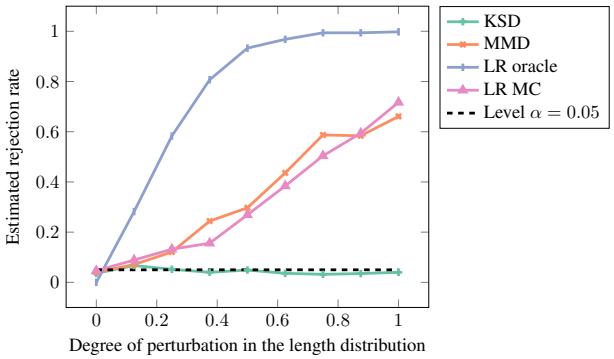


Figure 5: Estimated rejection rate of the proposed KSD test. We use a randomly generated Markov chain as the model distribution and perturb the distribution of lengths for the data. The proposed KSD test fails to reject in this setting.

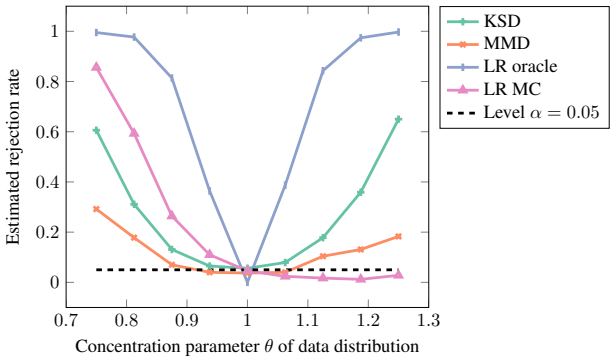


Figure 6: Performance comparison using a simple MRF model with a concentration parameter θ varied; the parameter θ is fixed at $\theta = 1$ for the model, and is chosen from 0.75 to 1.25 for the sample distribution. Our proposed test outperforms both non-oracle baselines in one region and outperforms MMD in the other region.

4.4. Sensitivity to Changes in Length

There is a caveat with the family of neighborhoods we have described: these neighborhoods are not particularly sensitive to changes in the distribution over sequence lengths. We demonstrate this point by choosing a randomly generated Markov chain over a finite alphabet $|\mathcal{S}| = 10$ as the model distribution, with the transition kernel again chosen from a Dirichlet distribution as above. Here, the ground truth and model use the same transition kernel, but we vary the stopping probability. The stopping probability is selected such that the expected sequence length is 8 under the model distribution, whereas we vary the sequence length between 8 and 20 under the ground truth.

We use the same tests as in the previous experiment, except that we use a subsequence length t of 2 for the CSK kernel and the alternative for the LR baseline now consists of first- rather than second-order Markov chains.

In Figure 5, we can see that the KSD test fails to reject entirely, outperformed by both types of baseline. This experimental result highlights the need for bespoke tests, when we desire that a test is sensitive to a specific aspect of the distribution. A test user who is particularly interested in the length distribution would likely benefit from directly testing the distribution over lengths using a classical test.

4.5. Testing Intractable Models

A major benefit of the proposed test is that it does not require us to sample from the model distribution, nor to compute a normalized density. Here, we showcase our test using a Markov Random Field (MRF) model, a class of intractable models. An MRF model P is defined by normalizing an exponentiated *potential function* h, i.e., we have $p(x) \propto \exp(h(x))$. The normalization constant is often intractable, and hence so is the probability mass function p.

We design an MRF model by specifying a single potential h across the entire sequence space $\mathcal{X}$:

$$h\left(x_{(1:\ell)}\right) := \begin{cases} C \cdot \ell + \theta \sum_{i=1}^{\ell-1} \delta_{x_i}\left(x_{i+1}\right), & \text{if } \ell \leq M \\ -\infty, & \text{else.} \end{cases}$$

In our experiment below, we hold the length parameters C and M fixed, while perturbing the distribution by varying θ. The parameter θ is a concentration parameter controlling the correlation between successive elements of a sequence; symbol repetitions become more common as θ grows.

Our problem is given as follows. We specify $\theta = 1$ for the model distribution and vary $\theta \in [0.75, 1.25]$ for the data distribution; the setup tests the sensitivity to this perturbation. We specify $M = 20$ in all cases, and vary C with θ so that the mean length is fixed at 10 and the length distribution does not depend on θ. The alphabet size is 3. We compare the proposed test against the MMD and LR baselines. Note that although it is generally challenging to sample from MRF models or to compute their normalized densities, the above special model allows us to perform both operations, and hence MMD and LR tests. For the LR baseline we use an oracle test (with access to the data distribution), as well as a test with H_1 consisting of first-order Markov chains; the latter baseline does not contain the above MRF family in the alternative. The MMD and KSD tests use the CSK kernel with subsequence length $t = 2$.

The proposed test performs competitively with the non-oracle baselines. In Figure 6 we can see that the proposed test outperforms the MMD baseline across the full range of perturbations. In particular, it outperforms the non-oracle LR baseline in one region; the LR baseline fails as θ grows, due to lack of consistency. We emphasize that our proposed test applies more generally than the LR test, since we do

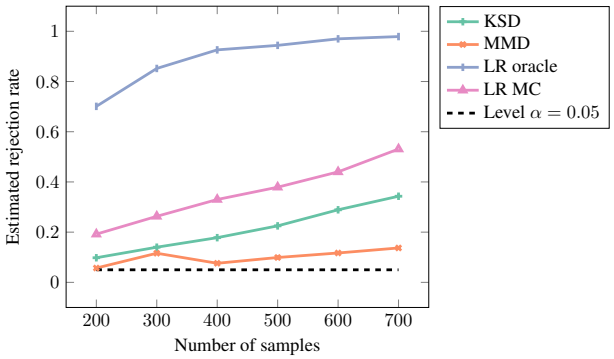


Figure 7: Performance of the proposed KSD test on a simple MRF model. We perform the same experiment as shown in Figure 6, in this case holding fixed $\theta = 0.9$ and varying the number of samples. In this region, our proposed test outperforms the MMD baseline but not the LR baseline.

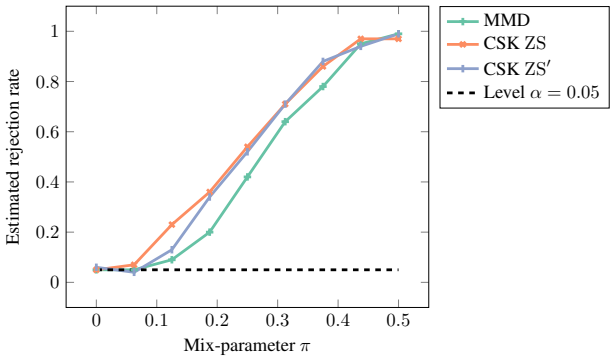


Figure 8: Performance comparison in the language model experiment. The two KSD tests outperform the MMD test. In particular, the ZS variant beats the ZS′ one when the mixture proportion π is small, showing a benefit of the richer neighborhood structure.

not require normalized densities. Moreover, using the wild bootstrap procedure, we do not need to sample from the model, unlike the MMD test.

We also show how test power improves with the data size in Figure 7. We choose the region where our test outperforms the MMD baseline, but not the LR baseline. The parameter θ is fixed to 0.9, while we vary the sample size n. We can see that the MMD baseline has a low sample efficiency for this problem, confirming again the MMD's inability to use the model structure. Both the LR baseline and our proposed KSD test make use of model information. Only the KSD test can work in the absence of a normalized density, however.

4.6. Inspecting Output-Constrained Generative Models

In practical applications, we are often interested in the ability of a model to output a particular set of patterns or values. For example, in language modeling, one might be interested in the ability to generate a subset of a language or sentences of a particular form (e.g., questions). Mathematically, we can state this objective as follows: for a model P, we aim to evaluate the goodness-of-fit of its conditional distribution P_A, derived by restricting the model to a set $A \subset \mathcal{X}$ (see below for an example of A). The probability mass function (pmf) p_A of the conditional P_A is given by

$$p_A(x) = \frac{p(x)}{\sum_{\tilde{x} \in A} p(\tilde{x})},$$

where p is the pmf of P. The normalization constant appearing in p_A is typically unknown (even when p is normalized) and thus as in the MRF experiment, this task is challenging both for MMD and likelihood-based diagnostics.

Following the above setup, we conduct an additional experiment. Specifically, we consider evaluating the ability of a language model to generate *question sentences* (i.e., A is the set of sequences ending with symbol '?'). Our task is detecting the departure of one model configuration from its mixture with another (with the mixture parameter π varied). We use the pretrained model `gpt2` (Radford et al., 2019) to obtain two configurations: these are attained by inputting two different prompts: `How are` and `Where is`. To sample from the language model, we generate a sequence of up to 5 additional tokens following the prompt, stopping when we encounter the token '?'. The sample is rejected if this token is not encountered within 5 steps following the prompt. For each prompt, this gives a different distribution over question sentences. The model distribution is specified as the output with the prompt `How are`, while the ground truth is a mixture of outputs by randomly selecting between the two prompts. We draw $n = 50$ samples.

We compare the KSD against the MMD test. For the MMD, we generate samples using rejection sampling. This is tractable in this experiment, because the prompts are designed to yield a low rejection probability (questions are likely to be generated). For the KSD, we use the ZS and ZS′ configurations as in Section 4.2 with $J = 1$. We choose sparse, point-dependent insertion and substitution neighborhoods; we use 16 high-probability tokens for a given context, and thus do not use the sparsification approach from Section 4.2. Both MMD and KSD tests use wild bootstrap procedures. Here, we do not include likelihood-based tests as they are intractable in this setting. For further experimental details, see Appendix A.2. Figure 8 shows the result, showcasing the KSD's utility in a challenging task.

Acknowledgements

We thank the three anonymous referees for their helpful feedback. HK and AG acknowledge the financial support by the Gatsby Charitable Foundation.

References

Anastasiou, A., Barp, A., Briol, F.-X., Ebner, B., Gaunt, R. E., Ghaderinezhad, F., Gorham, J., Gretton, A., Ley, C., Liu, Q., Mackey, L., Oates, C. J., Reinert, G., and Swan, Y. Stein's method meets computational statistics: A review of some recent developments. *Statistical Science*, 38(1), 2021. doi: 10.1214/22-sts863.

Aronszajn, N. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

Barbour, A. D. Stein's method and Poisson process convergence. *Journal of Applied Probability*, 25:175–184, 1988. Publisher: Cambridge University Press.

Barker, A. Monte Carlo calculations of the radial distribution functions for a proton-electron plasma. *Australian Journal of Physics*, 18(2):119, 1965. doi: 10.1071/ph650119.

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

Blei, D., Ng, A., and Jordan, M. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Bresler, G. and Nagaraj, D. Stein's method for stationary distributions of Markov chains and application to Ising models. *Annals of Applied Probability*, 29, 2019.

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

Cuturi, M. Fast global alignment kernels. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 929–936, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Cuturi, M., Vert, J.-P., Birkenes, O., and Matsui, T. A kernel for time series based on global alignments. 2007. doi: 10.1109/icassp.2007.366260.

Dehling, H. and Mikosch, T. Random quadratic forms and the bootstrap for U-statistics. *Journal of Multivariate Analysis*, 51(2):392–413, 1994. Publisher: Elsevier.

Fernandez, T., Rivera, N., Xu, W., and Gretton, A. Kernelized Stein discrepancy tests of goodness-of-fit for time-to-event data. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 3112–3122, 2020.

Gorham, J. and Mackey, L. Measuring sample quality with Stein's method. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Gorham, J. and Mackey, L. Measuring sample quality with kernels. In *Proceedings of The 34th International Conference on Machine Learning*, pp. 1292–1301, 2017.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. Publisher: JMLR. org.

Haussler, D. Convolution kernels on discrete structures, 1999.

Hodgkinson, L., Salomone, R., and Roosta, F. The reproducing stein kernel approach for post-hoc corrected sampling. 2020.

Hoeffding, W. A class of statistics with asymptotically normal distribution. *Ann. Math. Statist.*, 19(3):293–325, September 1948.

Kallenberg, O. *Foundations of Modern Probability*. Springer International Publishing, 2021. doi: 10.1007/978-3-030-61871-1.

Kanagawa, H., Jitkrittum, W., Mackey, L., Fukumizu, K., and Gretton, A. A kernel Stein test for comparing latent variable models. To appear in Journal of the Royal Statistical Society. Series B, (Statistical Methodology), 2023.

Király, F. J. and Oberhauser, H. Kernels for sequentially ordered data. 20, 2019. Publisher: Journal of Machine Learning Research.

Leslie, C. and Kuang, R. Fast string kernels using inexact matching for protein sequences. *Journal of Machine Learning Research*, 5:1435–1455, dec 2004. ISSN 1532-4435.

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

Lloyd, J. and Ghahramani, Z. Statistical model criticism using kernel two sample tests. In *Advances in Neural Information Processing Systems*, pp. 829–837, 2015.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. Text classification using string kernels. 2:419–444, 2002.

Oates, C. J., Girolami, M., and Chopin, N. Control functionals for Monte Carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79 (3):695–718, 2016. doi: 10.1111/rssb.12185.

Power, S. and Goldman, J. V. Accelerated sampling on discrete spaces with non-reversible markov processes. 2019.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Reinert, G. and Ross, N. Approximating stationary distributions of fast mixing Glauber dynamics, with applications to exponential random graphs. *Annals of Applied Probability*, 29, 2019.

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

Shi, J., Zhou, Y., Hwang, J., Titsias, M. K., and Mackey, L. W. Gradient estimation with discrete Stein operators. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Sriperumbudur, B. K., Fukumizu, K., and Lanckriet, G. R. G. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12: 2389–2410, 2011.

Stein, C. M. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. 1972.

Steinwart, I. and Christmann, A. *Support Vector Machines*. Information Science and Statistics. Springer, 2008 edition, 2008.

Wynne, G., Kasprzak, M., and Duncan, A. B. A spectral representation of kernel Stein discrepancy with application to goodness-of-fit tests for measures on infinite dimensional hilbert spaces. 2022. doi: https://doi.org/10.48550/arXiv.2206.04552.

Xu, W. and Matsuda, T. A Stein goodness-of-fit test for directional distributions. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pp. 320–330, 2020.

Yang, J., Liu, Q., Rao, V. A., and Neville, J. Goodness-of-fit testing for discrete distributions via Stein discrepancy. In *ICML*, 2018.

Yang, J., Rao, V., and Neville, J. A Stein–Papangelou goodness-of-fit test for point processes. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 226–235, 2019.

Zanella, G. Informed proposals for local MCMC in discrete spaces. 115:852 – 865, 2019.

A. Experimental details

We provide further details for the experiments that we conducted.

A.1. Likelihood Ratio Test

The LR baselines use the likelihood ratio statistic:

$$T_m^{\mathrm{LR}}\left(\left\{Y^{(l)}\right\}_{l=1}^m\right) := 2\left(\sup_{p\in H_0\cup H_1}\log p\left(\left\{Y^{(l)}\right\}_{l=1}^m\right) - \sup_{p\in H_0}\log p\left(\left\{Y^{(l)}\right\}_{l=1}^m\right)\right), \tag{8}$$

where H_0 and H_1 are the null and alternative hypotheses, respectively. In our experiments, we use a simple null hypothesis for the likelihood ratio statistic. The alternative used in the oracle baseline is a simple hypothesis containing only the (normally unknown) ground truth. The alternative used in the MC baseline is a composite hypothesis consisting of all Markov chains of some order, with the order depending on the experiment. In experiments where we use an alternative consisting of first-order Markov chains, the alternative consists of all initial distributions over the alphabet together with all transition kernels through the alphabet, with an additional stopping point:

$$H_1 := \mathrm{Sim}_{\mathcal{S}} \times \left(\mathrm{Sim}_{\mathcal{S}\cup\{\text{stop}\}}\right)^{\mathcal{S}}, \tag{9}$$

$$\mathrm{Sim}_{\mathcal{S}} := \left\{(\theta_i)_{i\in\mathcal{S}} : \sum_{i\in\mathcal{S}}\theta_i = 1 \text{ and } \theta_i \geq 0 \text{ for any } i\in\mathcal{S}\right\}. \tag{10}$$

In experiments where we use an alternative consisting of second-order Markov chains, the alternative consists of the analogous collection of second-order kernels:

$$H_1 := \mathrm{Sim}_{\mathcal{S}} \times \left(\mathrm{Sim}_{\mathcal{S}\cup\{\text{stop}\}}\right)^{\mathcal{S}} \times \left(\mathrm{Sim}_{\mathcal{S}\cup\{\text{stop}\}}\right)^{\mathcal{S}\times\mathcal{S}}. \tag{11}$$

A.2. Further Details Regarding the Language Model Experiment in Section 4.6

In Section 4.6, we consider the problem of evaluating a generative model with an output constraint. Although the support of the distribution is restricted to the set of question sentences, our setting still covers this scenario because we may construct a question sentence from any sequence by adding the suffix '?' (i.e., the set of all sequences is bijective to that of all question sequences). In reality, however, the assumption that the model has positive probability everywhere is unlikely to hold; e.g., some tokens never appear in questions (our experiment limits possible sequences with prompts). This assumption is placed to ensure that the formalism is well-defined; in practice, to follow the formalism, we may assume that the model has extremely small probabilities that do not affect the model's output effectively. That said, it is more useful to consider a neighborhood structure that spans the model's support rather than the entire sequence space. As described below, our neighborhood design excludes any points which are zero-probability under the model.

The model is configured using *top-k filtering* with $k = 16$. The sampling procedure is as follows: When the model samples the next token, x_{n+1}, for an already-sampled prefix $s = (x_1, \ldots, x_n)$, only the most likely k tokens from the alphabet are considered. That is, we sort the alphabet $\mathcal{S}$ using the conditional model density $\Pr(x_{n+1} = \bullet | x_1, \ldots, x_n)$, and keep only the k elements which have the largest probability using this ordering. Then, we renormalize the probabilities of these k points and sample the next token from the resulting distribution.

As mentioned in the main body, we use point-dependent insertion and substitution sets. Specifically, for a given sequence $s = (x_1, \ldots, x_n, '?')$, the substitution neighborhood $\mathcal{R}_{x_{(1:n)},n}$ consists of all sequences of the form $s' = (x_1, \ldots, x'_n, '?')$, where x'_n is chosen from any of the top-k next tokens after the prefix $s_{(1:n-1)}$. Similarly, the insertion neighborhood $\mathcal{I}_{x_{(1:n)},n}$ consists of all top-k sequences $s' = (x_1, \ldots, x_n, x'_{n+1}, '?')$, where x'_{n+1} is any of the top-k next tokens after the prefix $s_{(1:n)}$. The deletion neighborhood consists of the single sequence $s' = (x_1, \ldots, x_{n-1}, '?')$. Since an observed point must have non-zero probability (otherwise we can immediately reject the null hypothesis), x_n must itself be one of the top-k next tokens after the prefix $s_{(1:n-1)}$ Hence, substitutions are symmetric. Insertions are similarly symmetric.

This prefix-dependent neighborhood generalizes the approach described in Section 4.2, where we used a subset of the alphabet as a symbol neighborhood $\mathcal{N}_{\mathrm{rep}}(s) = \mathcal{N}_{\mathrm{rep}}(x_n) \subsetneq \mathcal{S}$. Our generalization is that the substitution neighborhood depends not on the symbol being replaced, but on the point as a whole via the prefix. Similarly, the insertion neighborhood also depends on the point as a whole.

The neighborhood graph used in the ZS′ variant is not strongly connected, rendering the resulting test inconsistent. The neighborhood graph used in the ZS variant is not guaranteed to be strongly connected. This is because the neighborhood graph that we use satisfies two conditions: the edges consist only of transitions with a bounded edit distance (all transitions have edit distance 1), and transitions to zero-probability sequences are excluded from the graph. Depending on the model structure, it may then be the case that the neighborhood graph is disconnected. For example, the sequences `Where is my pigeon?` and `Where is the cat?` may be in the support of the model. A path in the neighborhood graph used in the ZS variant would have to pass through the sequences `Where is my?` and `Where is the?`. We would not expect the these rather unnatural sentences to be in the support set of the model when top-k filtering is used, and in that case the necessary transitions would be excluded from the neighborhood graph. This would cause the graph to be disconnected.

We use the pretrained model from Hugging Face (https://huggingface.co/gpt2) to conduct the experiment.

A.3. Details of Testing Scenarios

In several experiments, we referred to a collection of 12 synthetic testing scenarios to evaluate the test power. We provide the details of these scenarios. The scenarios are intended to be dissimilar to each other in form and difficulty, although all model distributions and ground truths are Markov chains of some order.

Some of the distributions described here do not have support on the entire sequence space. In our earlier discussion, we required support everywhere as a technical condition relating to the connectivity of the Zanella process. To ensure that the probability of all sequences is positive, we introduce low-probability "restart" events into all of our Markov chains that resample uniformly from the alphabet with low probability, rather than sampling from the specified transition kernel. The probability of these events is small, $\varepsilon \approx 10^{-3}$. We do not mention this further, in order to simplify our discussion.

Binary i.i.d. sequences We draw sequences of i.i.d. Bernoulli variables. The model and ground truth differ in the parameter of the Bernoulli distribution.

Alphabet: $\{0, 1\}$.

Model distribution: Length follows a Poisson distribution with mean 20. Contents of the sequence are i.i.d. Bernoulli with $p = 0.6$.

Ground truth: Length follows a Poisson distribution with mean 20. Contents of the sequence are i.i.d. Bernoulli with $p = 0.4$.

Sample size: 10.

Binary sequences: misspecified Markov order We simulate the case where we are mistaken about the Markov order of the data. The model consists of i.i.d. Bernoulli variables, while the ground truth follows a simple first-order Markov chain.

Alphabet: $\{0, 1\}$.

Model distribution: Length follows a geometric distribution with mean 20. Contents of the sequence are i.i.d. Bernoulli with $p = 0.6$.

Ground truth: First-order Markov chain. Initial distribution uniform on $\{0, 1\}$. The transition kernel is

$$\Pr(\text{stop}|X_t = x) = 1/\lambda, \tag{12}$$

$$\Pr(X_{t+1} = y|X_t = x) = (1 - 1/\lambda)\,\delta_x(y), \tag{13}$$

with $\lambda = 20$. This gives sequences alternating between $\{0, 1\}$.

Sample size: 30.

Random walk I We take a random walk through a cyclic alphabet and perturb it by letting the ground truth hold, or "skip a step", with low probability.

Alphabet: $\mathcal{S} = \{1, \ldots, 8\}$ with a cyclic structure.

Model distribution: Random walk through $\mathcal{S}$ according to the cyclic structure. Start uniformly on $\mathcal{S}$. After each step, stop with probability $1/\lambda$, with $\lambda = 8$.

Ground truth: Random walk with holding. Start uniformly on $\mathcal{S}$. At each step, hold or "skip a step", with probability $p = 0.2$, letting $X_{t+1} = X_t$. Otherwise, sample a step from $\{-1, +1\}$ uniformly. After each step, stop with probability $1/\lambda$, with $\lambda = 8$.

Sample size: 30.

Random walk II The same scenario, but with few long sequences instead of many short sequences. We only let the ground truth hold in a subset of states, which reduces the degree of perturbation and makes the problem more challenging.

Alphabet: $\mathcal{S} = \{1, \ldots, 30\}$ with a cyclic structure.

Model distribution: Same as in `Random walk I` with $\lambda = 30$.

Ground truth: Same as in `Random walk I` but with $\lambda = 30$ and sampling the step from $\{-1, 0, +1\}$ according to:

$$\Pr\left(X_{t+1} - X_t = \Delta | X_t = x\right) = \begin{cases} p, & \text{if } \Delta = 0, x \leq i_{\text{max}}, \\ \frac{1-p}{2}, & \text{if } \Delta \in \{-1, +1\}, x \leq i_{\text{max}}, \\ \frac{1}{2}, & \text{if } \Delta \in \{-1, +1\}, x > i_{\text{max}}, \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

with $i_{\text{max}} = 8$ and $p = 0.2$.

Sample size: 8.

Random walk III We introduce memory into the random walk by correlating or anti-correlating successive steps, which gives us a simple second-order MC.

Alphabet: $\mathcal{S} = \{1, \ldots, 10\}$ with a cyclic structure.

Model distribution: Same as in `Random walk I`, but we modify the transition kernel to correlate successive steps. Specifically, letting $\delta = 0.95$, we sample the step from $\{-1, 0, +1\}$ according to:

$$\Pr\left(X_{t+2} - X_{t+1} = d | X_{t+1}, X_t\right) = \begin{cases} \delta, & \text{if } d = X_{t+1} - X_t, |d| = 1, \\ 1 - \delta, & \text{if } d = X_t - X_{t+1}, |d| = 1, \\ \frac{1}{2}, & \text{if } |d| = 1 < |X_t - X_{t+1}|, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

$$\tag{16}$$

We have to treat the case where $|X_t - X_{t+1}| > 1$ due to the low probability restart event mentioned earlier.

Ground truth: Same as model distribution but with $\delta = 0.05$.

Sample size: 30.

Random walk IV Same as in `Random walk III` but with $\lambda = 30$ and sample size 8. That is, few long sequences instead of many short ones.

Random second-order MC I We define a distribution over second-order MC and sample two random ones. The stopping probability is constant so that we can control the length independently of the perturbation of the contents. We sample many short sequences.

Alphabet: $\mathcal{S} = \{1, \ldots, 10\}$.

Model distribution: We sample a second-order Markov chain by sampling the initial distribution and transition kernels from Dirichlet distributions with concentration parameter $\alpha = 1$. At each step, the chain stops with probability $1/\lambda$ where $\lambda = 8$.

Ground truth: Same as model distribution, but the Markov chain is sampled from a different random number generator (RNG) seed in order to give a different MC.

Sample size: 30.

Random second-order MC II Same as in `Random second-order MC I` but with $\lambda = 20$ and sample size 8. That is, we sample a small number of long sequences.

Random second-order MC III Same as in `Random second-order MC I` but with $\lambda = 8$ and sample size 8. That is, we sample a small number of short sequences.

Random MC with varied initial distribution I We sample a single random first-order MC for the model and ground truth, and set a fixed initial distribution which we perturb. This poses a hard problem, as most of the observed data relates to transitions that come from an unperturbed transition kernel.

Alphabet: $\mathcal{S} = \{1, \ldots, 10\}$.

Model distribution: We sample a first-order Markov chain by sampling the transition kernel from a Dirichlet distribution with concentration parameter $\alpha = 1$. The initial distribution is uniform over $\mathcal{S}$. At each step, the chain stops with probability $1/\lambda$ where $\lambda = 8$.

Ground truth: Same as model distribution, with the same RNG seed. in order to give the same transition kernel. The initial distribution is an equal mixture of $\mathrm{Unif}\,(\mathcal{S})$ and $\mathrm{Unif}\,(\{1, 2\})$.

Sample size: 30.

Random MC with varied initial distribution II Same as in `Random MC with varied initial distribution I` but with $\lambda = 20$ and sample size 8. This problem is significantly harder, as we observe 8 rather than 30 instances of the initial distribution, and the longer sequences do not provide any additional evidence.

Random MC with varied length distribution We vary the distribution over lengths while keeping the content distribution the same.

Alphabet: $\mathcal{S} = \{1, \ldots, 10\}$.

Model distribution: We sample a first-order Markov chain by sampling the transition kernel from a Dirichlet distribution with concentration parameter $\alpha = 1$. The initial distribution is uniform over $\mathcal{S}$. At each step, the chain stops with probability $1/\lambda$ where $\lambda = 8$.

Ground truth: We use the same initial distribution and transition kernel as in the model. At each step, the chain stops with probability $1/\lambda$ where $\lambda = 20$.

Sample size: 30.

A.4. Results of Individual Testing Scenarios

We report the individual per-scenario rejection rates for an earlier experiment, for which we reported the average rate across scenarios in Figure 2. The per-scenario rates are shown in Table 1.

Two of the scenarios are designed to highlight the failure of the inconsistent test variants. These are *Binary sequences: True distribution is not i.i.d. sequence*, which shows the failure of the inconsistent CSK kernel; and *Random MC w/ varied length dist*, which shows the failure of the inconsistent variant ZS'.

Test	CSK ZS						CSK ZS$'$						CSK ZS$''$					CSK MMD
J	1	3	5	7	9	∞	1	3	5	7	9	∞	1	3	5	7	9	
Scenario																		
Binary sequences: Few long i.i.d. sequences	0.237	0.415	0.715	0.870	0.932	1.000	0.242	0.507	0.792	0.902	0.960	1.000	0.225	0.375	0.623	0.690	0.810	0.995
Binary sequences: True distribution is not i.i.d. sequence	0.075	0.022	0.050	0.030	0.030	0.040	0.033	0.007	0.013	0.018	0.013	0.005	0.065	0.048	0.050	0.080	0.090	0.013
Random 2nd-order MC: Few long sequences	0.040	0.158	0.282	0.438	0.500	0.907	0.075	0.172	0.335	0.420	0.517	0.917	0.045	0.180	0.253	0.352	0.472	0.670
Random 2nd-order MC: Few short sequences	0.065	0.117	0.225	0.280	0.357	0.477	0.068	0.155	0.207	0.315	0.360	0.443	0.033	0.135	0.255	0.297	0.375	0.125
Random 2nd-order MC: Many short sequences	0.177	0.420	0.720	0.835	0.907	0.973	0.175	0.490	0.760	0.887	0.910	0.968	0.080	0.422	0.710	0.877	0.935	0.970
Random MC w/ varied initial dist: Few long sequences	0.052	0.065	0.060	0.068	0.048	0.043	0.043	0.043	0.055	0.028	0.048	0.040	0.070	0.065	0.060	0.055	0.062	0.100
Random MC w/ varied initial dist: Many short sequences	0.075	0.052	0.102	0.048	0.060	0.050	0.055	0.080	0.090	0.068	0.150	0.105	0.068	0.062	0.065	0.098	0.075	0.125
Random MC w/ varied length dist	0.033	0.018	0.013	0.018	0.025	0.035	0.025	0.007	0.028	0.045	0.030	0.025	0.013	0.000	0.010	0.018	0.010	0.690
Random walk with memory: Few long sequences	0.220	0.323	0.487	0.570	0.580	0.978	0.395	0.465	0.517	0.580	0.655	0.958	0.085	0.060	0.133	0.223	0.263	1.000
Random walk with memory: Many short sequences	0.525	0.710	0.892	0.943	0.968	0.995	0.693	0.762	0.907	0.943	0.980	0.998	0.135	0.207	0.393	0.500	0.733	1.000
Random walk: Few long sequences	0.130	0.383	0.740	0.810	0.880	1.000	0.147	0.405	0.603	0.755	0.800	1.000	0.030	0.417	0.652	0.838	0.925	0.595
Random walk: Many short sequences	0.438	0.912	0.978	0.993	1.000	1.000	0.647	0.818	0.963	0.995	0.998	0.998	0.072	0.810	0.970	0.993	1.000	0.618
Test	Hamming ZS						Hamming ZS$'$						Hamming ZS$''$					Hamming MMD
J	1	3	5	7	9	∞	1	3	5	7	9	∞	1	3	5	7	9	
Scenario																		
Binary sequences: Few long i.i.d. sequences	0.030	0.065	0.070	0.043	0.040	0.100	0.100	0.092	0.133	0.220	0.215	0.195	0.033	0.037	0.040	0.052	0.037	0.060
Binary sequences: True distribution is not i.i.d. sequence	0.055	0.117	0.170	0.215	0.180	0.152	0.133	0.347	0.492	0.603	0.745	0.815	0.060	0.120	0.138	0.177	0.220	0.087
Random 2nd-order MC: Few long sequences	0.058	0.225	0.225	0.263	0.207	0.237	0.062	0.085	0.117	0.120	0.128	0.113	0.035	0.140	0.220	0.190	0.237	0.050
Random 2nd-order MC: Few short sequences	0.077	0.177	0.190	0.280	0.205	0.210	0.077	0.142	0.090	0.150	0.135	0.142	0.075	0.160	0.228	0.237	0.223	0.052
Random 2nd-order MC: Many short sequences	0.115	0.188	0.233	0.233	0.315	0.287	0.205	0.245	0.273	0.285	0.280	0.247	0.052	0.217	0.205	0.265	0.273	0.080
Random MC w/ varied initial dist: Few long sequences	0.065	0.022	0.060	0.070	0.062	0.035	0.037	0.058	0.045	0.077	0.062	0.090	0.048	0.062	0.070	0.052	0.040	0.043
Random MC w/ varied initial dist: Many short sequences	0.140	0.085	0.030	0.070	0.062	0.065	0.163	0.210	0.172	0.163	0.217	0.230	0.040	0.060	0.068	0.068	0.060	0.070
Random MC w/ varied length dist	0.005	0.015	0.025	0.037	0.033	0.225	0.015	0.013	0.005	0.010	0.003	0.000	0.013	0.007	0.010	0.030	0.030	0.275
Random walk with memory: Few long sequences	0.080	0.060	0.065	0.035	0.068	0.085	0.068	0.113	0.077	0.090	0.045	0.060	0.052	0.037	0.070	0.052	0.075	0.077
Random walk with memory: Many short sequences	0.030	0.048	0.125	0.075	0.080	0.077	0.033	0.065	0.062	0.083	0.050	0.075	0.083	0.087	0.102	0.080	0.110	0.098
Random walk: Few long sequences	0.043	0.072	0.113	0.138	0.175	0.225	0.068	0.055	0.087	0.043	0.065	0.075	0.080	0.102	0.105	0.147	0.122	0.068
Random walk: Many short sequences	0.050	0.152	0.155	0.182	0.150	0.215	0.043	0.110	0.115	0.102	0.128	0.077	0.080	0.170	0.160	0.170	0.245	0.060

Table 1: Estimated rejection rate of three different families of Zanella-Stein test with varying neighborhood size and underlying kernel. We evaluate the tests on 12 different synthetic testing scenarios, evaluating each test multiple times on independently sampled synthetic datasets. To estimate the per-scenario rejection rate, we report an average across the test evaluations for each scenario. The data shown here is averaged across scenarios in Figure 2 above.

B. Additional Experiments

B.1. Balancing Function

We conduct experiments to show the relative performance of the Barker balancing function $t \mapsto t/(1+t)$ against the minimum probability flow (MPF) $t \mapsto \sqrt{t}$. Table 2 shows performance of a KSD on the 12 testing scenarios detailed in Appendix A.3, comparing the two balancing functions. Figure 9 shows performance of a KSD on the MRF experiment, comparing the two balancing functions against various baselines. We observe that the MPF operator tends to yield higher power than the Barker operator. A drawback of the MPF operator is that the ratio in Equation (1) can be numerically unstable when $p(x)$ is extremely small. The Barker operator circumvents this issue as the ratio becomes $p(y)/\{p(x)+p(y)\}$ (Shi et al., 2022, also mentions this point).

	Barker	MPF
Binary i.i.d. sequences	0.88	0.80
Binary sequences: misspecified Markov order	0.01	0.00
Random walk I	0.98	1.00
Random walk II	0.69	0.82
Random walk III	0.93	0.96
Random walk IV	0.54	0.66
Random second-order MC I	0.84	0.90
Random second-order MC II	0.38	0.53
Random second-order MC III	0.28	0.45
Random MC with varied initial distribution I	0.07	0.05
Random MC with varied initial distribution II	0.05	0.09
Random MC with varied length distribution	0.05	0.02

Table 2: Estimated test power of the Zanella-Stein kernel goodness-of-fit test, constructed from the CSK kernel using two different balancing functions. We choose either the Barker or the minimum probability flow (MPF) function.

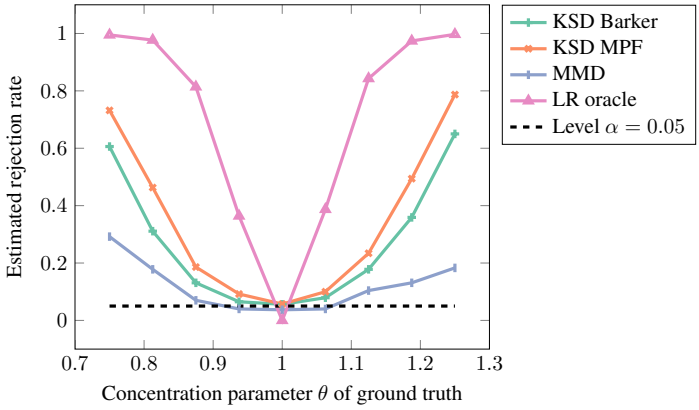


Figure 9: Performance comparison of two choices of balancing function in the construction of the KSD: Barker against minimum probability flow.

B.2. Larger Neighborhoods

In Section 3, we considered a neighborhood structure where we modified one element of the sequence. Here, we conduct an experiment to see the effect of using a larger edit distance. We use the MRF experiment in Section 4.5 as a benchmark and compare the original KSD used in Section 4.5 against a KSD that allows for larger edit distance. The neighborhood for the latter KSD is extended by adding double-substitution, that is, substitution in two locations. Figure 10 shows the result, where the oracle and MMD baselines are included as a reference. We can see that test sensitivity is at best improved by a negligible amount in one region, while similarly it worsens in the other region. For this problem, expanding the neighborhood as above does not improve test sensitivity despite a higher computational cost.

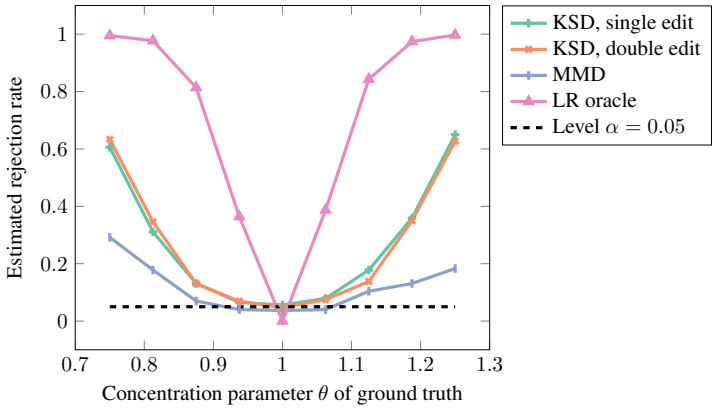


Figure 10: Performance comparison of the single-edit neighbourhood test versus a variant which allows for double-edits.