Transport Elliptical Slice Sampling

Alberto Cabezas
Lancaster University

Christopher Nemeth
Lancaster University

Abstract

We propose a new framework for efficiently sampling from complex probability distributions using a combination of normalizing flows and elliptical slice sampling (Murray et al., 2010). The central idea is to learn a diffeomorphism, through normalizing flows, that maps the non-Gaussian structure of the target distribution to an approximately Gaussian distribution. We then use the elliptical slice sampler, an efficient and tuning-free Markov chain Monte Carlo (MCMC) algorithm, to sample from the transformed distribution. The samples are then *pulled back* using the inverse normalizing flow, yielding samples that approximate the stationary target distribution of interest. Our transport elliptical slice sampler (TESS) is optimized for modern computer architectures, where its adaptation mechanism utilizes parallel cores to rapidly run multiple Markov chains for a few iterations. Numerical demonstrations show that TESS produces Monte Carlo samples from the target distribution with lower autocorrelation compared to non-transformed samplers, and demonstrates significant improvements in efficiency when compared to gradient-based proposals designed for parallel computer architectures, given a flexible enough diffeomorphism.

1 INTRODUCTION

Markov Chain Monte Carlo (MCMC) algorithms enable scientists to draw samples from complex distributions, which are typically produced by models that aim to represent the intricate details found in real-world datasets. The exploration of these complex and high-dimensional distributions is challenging, and to be efficient, practitioners use the local pointwise information of the target distribution to create a Markov chain of dependent samples. The ideal outcome would be to have independent samples, but the Markov chain approach generates samples that are correlated sequentially. Therefore, a major focus in MCMC research is to develop algorithms that reduce these correlations and generate samples that approximate independence.

Designing efficient MCMC algorithms usually relies on using local gradient information from the target distribution; by discretizing, for example, Hamiltonian (Duane et al., 1987; Neal et al., 2011) or Langevin (Rossky et al., 1978; Grenander and Miller, 1994) dynamics of a process stationary on our target distribution. Calculating gradients has been automated (Linnainmaa, 1976) but optimizing these algorithms to efficiently minimize both computations and correlations between sequential samples requires algorithmic parameters to be manually tuned. Much work has been dedicated to developing efficient, black-box methods to tune these parameters, with notable examples including the NUTS (Hoffman et al., 2014) algorithm which is widely available in probabilistic programming languages (Salvatier et al., 2016; Carpenter et al., 2017; Bingham et al., 2019; Phan et al., 2019).

Within the machine learning community, variational inference (VI; Jordan et al., 1999) has grown in popularity as an inexact but comparatively faster approach to solving the same inferential problem. As such, MCMC has lost its preferential status as the default approach for Bayesian inference for prediction and uncertainty quantification in this thriving community. Recent efforts (Hoffman et al., 2021; Hoffman and Sountsov, 2022) have focused on speeding up MCMC by focusing on widening instead of lengthening computations on modern computer architectures, e.g. utilizing GPUs or TPUs, which allow for vast parallel computations. Tuning parallel MCMC chains has proven to be a somewhat different challenge from its sequential counterpart (Radul et al., 2020) and parallel efforts need to consider the lockstep necessity of gradient evaluations of parallel chains on modern vector oriented libraries (Abadi et al., 2016; Bradbury et al., 2018; Paszke et al., 2019).

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

Figure 1: Illustration of Algorithm 1 using an exact transport map, i.e. equality in (2) holds: sampling from the Banana density $(x_1; x_2) / \exp[x_1^2/8 + x_2 - x_1^2/4 - 2]/2$ using the transport map $T(u_1; u_2) = (\sqrt{8}u_1; u_2 + 2u_1^2)$ starts by transforming the target space to the reference space via a change of variables, drawing samples from an ellipsis on the extended reference space (not pictured) and pushing samples back to the target space.

2 TRANSPORT ELLIPTICAL SLICE SAMPLER

In this paper we assume that $x \in \mathcal{X} \subseteq \mathbb{R}^d$ are model parameters and D represents our data. Our goal is to then approximate the posterior distribution, where by Bayes rule the posterior density is given by $(x) / L(D|x)\ _0(x)$, for $L(D|x)$ the likelihood function and $_0(x)$ the prior density. Our goal is to introduce a new MCMC algorithm which leverages the tuning-free nature of elliptical slice sampling with the efficient density transformation tools of normalising flows, thus creating the transport elliptical slice sampler (TESS); an adaptive mechanism that allows scientists to perform fast parallel sampling from unnormalized densities. An intuitive pictorial representation of our TESS algorithm is given in Figure 1.

2.1 Elliptical slice sampling

Introduced by Murray et al. (2010) as a simple MCMC algorithm with no tuning parameters, the elliptical slice sampler builds on a Metropolis-Hasting sampler introduced by Neal (1998), which is designed for situations where the prior $_0(x)$ is Gaussian. Without loss of generality, we can assume that the prior is a standard Gaussian density $_0(x) = (x)$[1]. The algorithm of Neal (1998) proceeds by first proposing new state of the Markov chain, $x' = \sqrt{1 - ^2}x + v$, where $v \in \mathbb{R}^d$ is an independent momentum variable following a standard Gaussian distribution. The proposal moves x along the half ellipse which connects the points x and v which pass thorough x, for values $\in [-1; 1]$.

Elliptical slice sampling, instead, uses the proposal $x' = x\cos + v\sin$ which moves on the full ellipse connecting x, $-x$, v and $-v$ for $\in [0; 2]$. Both proposals leave the prior density invariant and elliptical slice sampling uses the slice sampling algorithm (Neal, 2003) to choose a value which ensures that the likelihood $L(D|x)$ is invariant. Overall, this proposal scheme keeps the target posterior invariant (Murray et al., 2010), details of which are presented in the Supplementary material for completeness.

[1] shift and scale x if non-standard.

2.2 Normalizing flows

Normalizing flows (NF; Rezende and Mohamed, 2015) are a flexible class of transformations produced by the sequential composition of invertible and differentiable mappings. Using NF involves choosing a simple reference density, for example a standard Gaussian distribution (u), and a parameterized diffeomorphism T, with optimized parameters, to transform the reference density to our target (x) via a change of variables. In other words, we want to find a map T such that for u and $x = T(u)$ we have x. Assuming this function exists, applying a change of variable yields the following identities

$$(x) = (T^{-1}(x))|\det r\, T^{-1}(x)| =: \hat{}(x) \quad (1)$$

$$(u) = (T(u))|\det r\, T(u)| =: \hat{}(u); \quad (2)$$

where $r\, T$ and $r\, T^{-1}$ are the Jacobian matrices of T and its inverse, respectively. In the context of VI, an approximation of $\hat{}(x)$ would serve as an approximation to our target density when carrying-out inference, since this approximation is both normalized and trivial to sample from.

2.3 Fixed transport maps with elliptical slice sampling

To fulfill our requirement for a simple and cost-effective MCMC proposal, we begin by generalizing the dimension-independent, gradient-free, and tuning-free elliptical slice sampler. We will add tuning parameters to our generalized elliptical slice sampler using NF. The diffeomorphism T will be responsible for efficiently exploring the posterior target density by transforming the proposal's dynamics and tracing the contours of a standard Gaussian density to follow

the contours of an approximation of the target. Following previous works in the transport Monte Carlo field (as described in Section 3), we present TESS as a two-step procedure. Firstly, we learn the transport map between the target and reference densities. Secondly, we utilize the transport map within the elliptical slice sampler to generate samples from the target density.

1. Map optimization To estimate the parameters of our NF map we minimize a divergence between our target density $\pi(x)$ and the push-forward reference density $\hat{\pi}(x)$ (1). For our intended purpose, by the law of the unconscious statistician, this is equivalent to minimizing the divergence between the pull-back target density $\hat{\pi}(u)$ (2) and the reference density $\phi(u)$. The Kullback-Leibler divergence (KL; Kullback and Leibler, 1951) is arguably the most widely used and studied divergence, here presented in the context of approximate Bayesian inference but also studied in other branches of statistics and information theory (Joyce, 2011). It not only has a tractable Monte Carlo estimate, it is directly related to the foundation of VI and provides intuition into the connection between maximizing the likelihood of observational data and minimizing the distance between target and reference densities (Blei et al., 2017),

$$\mathrm{KL}(\phi \,||\, \hat{\pi}) = \int \log \frac{\phi(x)}{\hat{\pi}(x)} \phi(x) dx. \tag{3}$$

The optimal transport map is found by optimizing the the parameters ψ of the diffeomorphism T_ψ such that the Kullback-Leibler divergence between the target and reference densities is minimised, i.e.

$$\psi = \arg\min_{\psi \in \Psi} \mathrm{KL}(\phi \,||\, \hat{\pi}). \tag{4}$$

2. Sampling from the target Our proposed sampling method generalizes the elliptical slice sampler by targeting the extended state space $\pi(x)\,\phi(v)$ for any posterior density $\pi(x)$, regardless of the choice of prior distribution. The target density is preconditioned using a transform via a normalizing flow to map to a standard Gaussian distribution. That is, given a map T_ψ, with fixed parameters ψ, such that $\hat{\pi}(u) \approx \phi(u)$ we proceed as follows: i) from an initial state $(x, v) = (T_\psi(u), v)$, ii) move around an ellipse connecting u and v and iii) accept the new state according to a slice variable chosen uniformly on the interval $[0, \hat{\pi}(u)\,\phi(v)]$. One iteration of this method is detailed in Algorithm 1.

Algorithm 1 Transport Elliptical Slice Sampler

Require: $u, T_\psi(\cdot), \hat{\pi}(\cdot)$

$v \sim \mathcal{N}(0, I_d)$
$w \sim \mathrm{Uniform}(0, 1)$
$\log s \leftarrow \log \hat{\pi}(u) + \log \phi(v) + \log w$
$\theta \sim \mathrm{Uniform}(0, 2\pi)$
$[\theta_{\min}, \theta_{\max}] \leftarrow [\theta - 2\pi, \theta]$
$u' \leftarrow u\cos\theta + v\sin\theta$
$v' \leftarrow v\cos\theta - u\sin\theta$
if $\log \hat{\pi}(u') + \log \phi(v') > \log s$ **then**
  $x' \leftarrow T_\psi(u')$
  Return x', u'
else
  if $\theta < 0$ **then**
    $\theta_{\min} \leftarrow \theta$
  else
    $\theta_{\max} \leftarrow \theta$
  end if
  $\theta \sim \mathrm{Uniform}(\theta_{\min}, \theta_{\max})$
  Go to 6.
end if

Proposition 1 The transition kernel of the Markov chain derived from Algorithm 1 leaves the target density $\pi(x)\,\phi(v)$ invariant.

The TESS algorithm is likely to be geometrically ergodic under certain transformations, if those transformations lead to nice tail properties on the pulled back target. A sketch of this argument follows from three key components: (i) Natarovskii et al. (2021) show that the standard elliptical slice sampler is geometrically ergodic if the target density has tails which are rotationally invariant and monotonically decreasing, e.g. $\propto \exp(-c||x||)$ for $c > 0$. (ii) This implies geometric ergodicity for TESS if for a target density $\hat{\pi}$ and Markov transition kernel $P(x, \cdot)$, with $C > 0$ and $\alpha \in (0, 1)$, geometric ergodicity of the elliptical slice sampler holds when

$$||P^n(x, \cdot) - \hat{\pi}||_{TV} \leq C(1 + ||x||)^{\alpha n}, \quad \forall n \in \mathbb{N}, \forall x \in \mathbb{R}^d.$$

Then for $P_\psi(x, \cdot)$ the transition kernel of TESS we have,

$$\begin{aligned} ||P_\psi^n(u, \cdot) - \hat{\pi}||_{TV} &= ||P^n(T_\psi(u), \cdot) - \hat{\pi}||_{TV} \\ &\leq C(1 + ||T_\psi(u)||)^{\alpha n}, \end{aligned}$$

which holds only if the transformation T_ψ leads to nice tail properties for $\hat{\pi}$. (iii) Following from Theorems 2 and 3 of Johnson and Geyer (2012), if there exists a diffeomorphism which ensures that T_ψ pulls in the tails of the distribution enough, then geometric ergodicity holds on the transformed distribution. An open question is to determine the necessary conditions on T_ψ for this result to hold beyond simple transformations.

2.4 Adaptive transport maps

There are two key components to TESS, the MCMC sampling phase and the transformation function T_ψ, which so far we have treated as two independent procedures. However, the function T_ψ is parameterised by ψ and these parameters must be learnt using samples from the target $\pi(x)$. We therefore propose an adaptive version of TESS which alternates between optimizing and sampling x to produce an

accurate map between the reference measure and the target distribution.

The parameters are optimized by first running the TESS sampling procedure (Alg. 1) using k parallel Monte Carlo chains with an initial value of 0, resulting in k approximate samples from our target (x). We then run m iterations of a stochastic gradient descent algorithm on the loss function

$$\mathrm{KL}(\ (x)\|\ \hat{}(x)) \quad \frac{1}{k}\sum_{i=1}^{k} \log\frac{(x_i)}{\hat{}(x_i)}: \qquad (5)$$

The warm-up stage of the sampler repeats this process for h epochs with batches of size k, adjusting the inherited parameters from the previous epoch and finally fixing the parameters to then iterate N times Algorithm 1, generating samples from our extended target space (x) (v). This adaptive sampling algorithm is detailed in Algorithm 2.

An important property of the Kullback-Leibler divergence is that it is an asymmetric divergence, i.e. $\mathrm{KL}(\ \|\ \hat{}) \neq \mathrm{KL}(\ \hat{}\|\ )$. Minimizing $\mathrm{KL}(\ \hat{}\|\ )$ forces (x) to cover the mass of $\hat{}(x)$ thus produces a poor approximation of the tails of the posterior target density. Alternatively, minimizing $\mathrm{KL}(\ \|\ \hat{})$ forces $\hat{}(x)$ to cover the mass of (x), providing an overconfident approximation to the target density that can be corrected using a sampling method that leaves the target distribution invariant.

Algorithm 2 Adaptive TESS

Require: $u_{1:k}^{(0)}; h; m; N;$ TESS. TESS applies Algorithm 1

Set initial parameters of T and $\hat{}$.
for $t \quad 1; : : : ; h$ do . Warm-up
  for $i \quad 1; : : : ; k$ do
    $x_i^{(t)}; u_i^{(t)} \quad \mathrm{TESS}(u_i^{(t \ 1)}; T\ ; \hat{})$
  end for
  Update in T by running m iterations of gradient descent on (5) using samples $u_{1:k}^{(t)}$.
end for
$u_{1:k}^{(0)} \quad u_{1:k}^{(h)}$
for $t \quad 1; : : : ; N$ do . Sampling
  for $i \quad 1; : : : ; k$ do
    $x_i^{(t)}; u_i^{(t)} \quad \mathrm{TESS}(u_i^{(t \ 1)}; T\ ; \hat{})$
  end for
end for
Return $x_{1:k}^{(1)}; : : : ; x_{1:k}^{(N)}$

We follow the approach of Hoffman et al. (2019) and initialize the parameters of the NF using an approximation of the parameters that minimize $\mathrm{KL}(\ \hat{}\|\ )$ via a stochastic gradient descent scheme. In other words, minimizing the Monte Carlo approximation

$$\mathrm{KL}(\ (u)\|\ \hat{}(u)) \quad \frac{1}{M}\sum_{i=1}^{M} \log\frac{(u_i)}{\hat{}(u_i)}; \quad u_i \overset{\text{iid}}{} : \qquad (6)$$

2.5 Choice of transport map

There is a wide class of linear and nonlinear functions which can be used within our normalizing flow map. In this paper, we focus on the coupling architecture first introduced by Dinh et al. (2014). Consider the disjoint partition $x = (x^A; x^B) \in \mathbb{R}^p \quad \mathbb{R}^{d\ p}$ and a coupling function $t(\ ;\ ) : \mathbb{R}^p \to \mathbb{R}^p$ parameterized by some set of parameters θ. Then, one can define a transformation $G : \mathbb{R}^d \to \mathbb{R}^d$ by the formula

$$x^A = t(u^A; (\ u^B)) := e^{\ 1} \quad u^A + \ _2 \qquad (7)$$

$$x^B = u^B; \qquad (8)$$

given parameters $: \mathbb{R}^{d\ p} \to \mathbb{R}^p \quad \mathbb{R}^p$ learned only from the extended input. Here we assume an affine bijection, defined in (7), and make a dense feedforward neural network, for further generalizations and variations see Kobyzev et al. (2020). The main practical advantages of the coupling architecture with affine transformations are that it is easily inverted through a shift and scale of the transformed x^A with parameters given by the unchanged $x^B = u^B$, and that the modulus determinant of its Jacobian matrix can be easily computed as $|\det \mathrm{D}\ G(x)| = \prod_{i=1}^{d} (e^{\ 1})_i$. Furthermore, since the inverse of the transformation is of similar structure, also its constant of volume change can be easily derived as $|\det \mathrm{D}\ G^{\ 1}(x)| = \prod_{i=1}^{d} (e^{\ 1})_i$. Both of these are using parameters given by $(\ u^B) = (\ x^B)$ and we drop the absolute value from our computations since the values being multiplied are non-negative. We allow for arbitrary complexity of our NF by introducing a transformation $D : \mathbb{R}^d \to \mathbb{R}^d$ with the same structure as G but with the roles of the random variables reversed, i.e. $x^A = u^A$ and $x^B = t(u^B; (\ u^A))$. Hence making our final NF a sequential composition of $n \quad 1$ transformations $T = D_n \quad G_n \quad \quad D_1 \quad G_1$.

3 RELATED WORK

Elliptical slice sampling The original elliptical slice sampler paper (Murray et al., 2010) presented a simple algorithm that worked well on scenarios of strong prior (Gaussian) information. Nishihara et al. (2014) were the first to explore the idea of generalizing this algorithm to any target distribution, while trying to maintain a simple kernel. Their proposal used a Student-t distribution to approximate the target, under the premise that this proposal would adequately cover the tails of the target density. Their work also considered an adaptive mechanism using parallel computing architectures, which accelerated the MCMC sampler by utilizing multiple chains with fewer iterations per chain. Fagan et al. (2016) also used a generalized elliptical slice sampling proposal paired with a preconditioning step to alleviate complex geometry on their target, in their case using expectation propagation to learn correlation structures for subsets of the parameter space. The main difference between previous

elliptical slice sampling work and our methodology is the use of normalizing ows to create a transport map between a Gaussian density (for which the sampler works well) and the target density of interest. As shown in Section 4, utilizing the richness of nonlinear transport maps produces a fast and highly ef cient MCMC algorithm.

Monte Carlo transport maps Our work draws inspiration and is closely related to several threads of work that approach the problem of simulation by simplifying the structure of the target density through a preconditioning step. For general MCMC proposals, Parno and Marzouk (2018) introduced the idea of learning a diffeomorphism using samples from an MCMC algorithm to approximate(3). Their work built on El Moselhy and Marzouk (2012)'s proposal for approximate inference, adding an MCMC kernel that corrects the approximation and provides asymptotically exact samples. Their work showed that a relatively simple transformation can provide valuable information about the global structure of the target density, thus improving the ef ciency of MCMC algorithms that use local gradient information on certain, especially degenerate, test cases.

MCMC with normalizing ows The NeuTra Hamiltonian Monte Carlo (HMC) algorithm introduced in Hoffman et al. (2019) combines neural transport maps with the HMC algorithm. This builds on the earlier work of Marzouk et al. (2016), who frame the approximate inference problem as solving a two-step process, where rstly an optimization problem is solved to nd a preconditioned diffeomorphism which minimizes(6), and then the preconditioned target is sampled from using a HMC algorithm. The NeuTra algorithm relies on gradient-based proposals to explore the target density. A key difference from the transport elliptical slice sampler is that gradients of the target density are not required, this makes the algorithm faster than gradient-based MCMC algorithms, and as illustrated in Section 4, this is achieved without sacri cing sampling accuracy due to the transport mapping. Additionally, TESS can be applied in settings where it is either infeasible to calculate target gradients, or they may be unstable (e.g the Neal's funnel density (Gorinova et al., 2020)).

4 EXPERIMENTS

In this section we compare the performance of the adaptive form of TESS (Alg. 2) with the performance of several state-of-the-art MCMC algorithms designed for parallel computer architectures. Speci cally, MEADS (Hoffman and Sountsov, 2022), ChEES-HMC (Hoffman et al., 2021) and the popular NUTS algorithm (Hoffman et al., 2014) where an adaptive step size is tuned such that the average cross-chain harmonic-mean acceptance rate is approximately 0.8. We also precondition the latter NUTS method using the same NF as in TESS, which leads to the NeuTra algorithm (Hoffman et al., 2019). We compare the effect of TESS's over tted adapted transformation against an under-tted transformation (i.e. reversing the KL) which, unlike TESS, is done independently and a priori to the sampling process. Each experiment runs all algorithms on 128 parallel chains for 400 warm-up iterations per chain, during which hyperparameters are tuned, followed by 100 iterations used to produce posterior samples, with xed hyperparameters. MEADS separates the 128 chains into 4 batches of 32 chains each and tunes parameters during all 500 iterations but only the last 100 are used as posterior samples.

The transform map used in all experiments uses 2 transformations of a -parameterized dense feedforward neural network with two hidden layers of the same dimension as the input (see Sec. 2.5 for details). The Adam (Kingma and Ba, 2014) method is used to estimate , with a learning rate that decays exponentially over 400 iterations at a rate of 0.1 using a different initial learning rate for each experiment. For the adaptive TESS algorithm, we set = 1 on all experiments.

We compare the experimental results of each algorithm based on their Monte Carlo sample ef ciency, as indicated by the maximum integrated autocorrelation time ( ) with standard deviation ( ). Additionally, we present the effective sample size (ESS) in terms of the median worst case integrated autocorrelation time, both for individual chains and when all chains are grouped together. A more ef cient algorithm is indicated by lower autocorrelations and higher ESS, as this indicates that samples are closer to being independent. To demonstrate the impact of computational cost on each algorithm, we normalize the ESS by the run time in seconds. ESS/sec considers the time spent adapting and sampling, therefore provides a fair comparison between algorithms. To assess the accuracy of the posterior approximation for each algorithm, we use the kernelized Stein discrepancy with U- and V-statistics, as described in (Gorham and Mackey, 2017). Lower values of U- and V-statistics indicate a better approximation of the target posterior. Further information on these diagnostics can be found in the Supplementary Material.

4.1 Biochemical oxygen demand model

We start with an experiment from (Parno and Marzouk, 2018) designed to undermine gradient methods because of its rapidly changing posterior correlation structure, which is challenging for standard samplers to explore. Gradient methods capture local geometry, but the local geometry in this example is not representative of the global geometry of the target and thus provides insuf cient information for ef cient sampling. On the other hand, the non-linear transformation of the target space with a NF-based approach captures the global, non-Gaussian structure of the target density.

https://github.com/albcab/TESS

Algorithm	$\hat{\tau}_{max}$		ESS	ESS/chain	ESS/sec	Stein U-stat.	Stein V-stat.
TESS	0.555	1.485	11523	90	1129.199	4.269e+02	4.570e+02
MEADS	9.959	1.468	643	5	208.613	1.476e+15	1.486e+15
ChEES-HMC	6.406	2.228	999	8	224.290	1.505e+16	1.510e+16
NUTS	9.579	1.427	668	5	19.625	1.187e+15	1.192e+15
NeuTra	9.553	1.502	670	5	15.579	1.082e+15	1.087e+15

Table 1: Biochemical oxygen demand model. Algorithm diagnostics where $\hat{\tau}_{max}$ is the maximum integrated autocorrelation time over all dimensions; ESS is the corresponding minimum effective sample size. Results are averaged over multiple chains of each sampler, and is the empirical standard deviation of $\hat{\tau}_{max}$ over these runs.

Figure 2: Samples from the target density $\pi(y)$ of the Biochemical oxygen demand model acquired by the TESS algorithm, mapped to $\hat{\pi}(u)$ (1), with diffeomorphism T learned from the warm-up procedure of Algorithm 2. With an approximation that overestimates the real variance (right) of our target (left) we are able to capture its global, non-Gaussian structure and explore it using a dimension independent and gradient-free method.

The simple biochemical oxygen demand model is given by $B(t) = \theta_0(1 - \exp(-\theta_1 t))$ for times $t < 5$. In this synthetic data experiment, we set the parameters $\theta_0 = 1$ and $\theta_1 = 0.1$ and simulate $y(t_i)$ observations at times t_i evenly spaced in $[0, 5)$ for $i = 1, \dots, 20$ such that $y(t_i) = \theta_0(1 - \exp(-\theta_1 t_i)) + \epsilon_i$, where $\epsilon_i \sim N(0, \sigma_y^2)$ and fixed $\sigma_y^2 = 2 \cdot 10^{-4}$. The target posterior density is given by the likelihood $L(y \mid \theta_0, \theta_1) = \prod_i N(y(t_i); B(t_i; \theta_0, \theta_1), \sigma_y^2)$ and a flat prior $\pi_0(\theta_0, \theta_1) \propto 1$. The numerical results are shown in Table 1 and Figure 2 plots the Monte Carlo approximation of the posterior for the original and transformed densities.

It is clear from the results that local gradient information is insufficient to efficiently sample from the rapidly changing local correlation structure of the target density. On the other hand, the learned transport map from the warm-up procedure of TESS provides a mass-covering approximation of the global structure of the target, demonstrated in Figure 2 by $\hat{\pi}(u)$, which allows the algorithm to move farther away from its initial position, exploring efficiently the entire target space, and yielding not only shorter autocorrelation times, but also the correct posterior estimates of the parameter space. In this specific case, gradient-based algorithms are forced to take very small steps while still encountering large rejection probabilities, thus being inefficient at producing samples from the posterior.

4.2 Sparse logistic regression

Next, we consider a sparse logistic regression model with hierarchies. Regression parameters of the logistic likelihood are given a horseshoe prior (Carvalho et al., 2009) which induces sparsity on the regressors, i.e. variable selection. These types of hierarchies on the prior scale of a parameter create funnel geometries that are hard to efficiently explore without local or global structure of the target.

Algorithms are run on the non-centered parametrization (Papaspiliopoulos et al., 2007) of our model using the numerical version of the German credit dataset[1]. The target posterior is defined by the likelihood $L(y \mid \beta, \lambda, \tau) = \prod_i \text{Bernoulli}(y_i; \sigma(\beta^T X_i))$, with sigmoid function $\sigma(\cdot)$, and prior $\pi_0(\beta, \lambda, \tau) = \text{Gamma}(\tau; 1/2, 1/2) \prod_j N(\beta_j; 0, \lambda) \text{Gamma}(\lambda_j; 1/2, 1/2)$. Numerical results for each MCMC algorithm are shown in Table 2. Notice how NUTS and NeuTra provide the best results but long sampling times reflect their inefficiency when running in parallel: every iteration takes as long as the longest chain takes to iterate. Waiting for all chains to catch up severely slows down sampling time, the same effect can be observed in all experiments.

As the dimension of the parameter space grows ($D = 51$ in this example), TESS will require more samples, i.e. more chains, for a low variance estimate of (6). In addition, a more complicated NF is required to capture the non-Gaussian structure of the high-dimensional target space. When either of these fail, and the diffeomorphism T is unable to capture the structure of the target space, the simple sampling procedure inherited from the elliptical slice sampler will struggle to sample from the target space, even if producing uncorrelated samples. We purposely illustrate the effect of a deficient transformation on a high dimensional problem in order for the practitioner to understand the caveats of our method. Studying ways to lower the variance of (6), using control variates (Lemieux, 2014) and similar methods (Botev and Ridder, 2017), as well as alternative

[1] https://archive.ics.uci.edu/ml/datasets/statlog+\(german+credit+data)

Algorithm	τ_{max}		ESS	ESS/chain	ESS/sec	Stein U-stat.	Stein V-stat.
TESS	5.182	0.352	1235	10	34.744	1.591e+00	1.693e+00
MEADS	7.105	0.413	901	7	49.453	9.408e-01	1.079e+00
ChEES-HMC	5.666	0.380	1130	9	81.588	1.193e+00	1.312e+00
NUTS	4.734	0.833	1352	11	0.379	1.004e+00	1.138e+00
NeuTra	2.482	1.949	2579	20	0.401	3.618e-01	4.971e-01

Table 2: Sparse logistic regression. Algorithm diagnostics where τ_{max} is the maximum integrated autocorrelation time over all dimensions; ESS is the corresponding minimum effective sample size. Results are averaged over multiple chains of each sampler, and σ is the empirical standard deviation of τ_{max} over these runs.

NF schemes is left to future work.

4.3 Regime switching Hidden Markov model

A important use of inference and uncertainty quanti cation is on time series data. In this example, we analyze nancial time series, speci cally the daily difference in log price data of Google's stock, referred to as returns, for $t = 1, \dots, 431$. We shall assume that at any given time t the stock's returns will follow one of two regimes: an independent random walk regime $r_t \sim N(\mu_1, \sigma_1^2)$, or an autoregressive regime $r_t \sim N(\mu_2 + r_{t-1}, \sigma_2^2)$. We de ne the two regimes as $s_t \in \{0, 1\}$ and the probability of switching between, or remaining within a regime at time t will depend on the regime at $t-1$, i.e. p_{s_{t-1}, s_t} for $s_{t-1}, s_t \in \{0, 1\}$. The transition probabilities $p_{1,1}$ and $p_{2,2}$, and their complementary probabilities $p_{1,2} = 1 - p_{1,1}$ and $p_{2,1} = 1 - p_{2,2}$ are treated as model parameters. Since the regime at any time is unobserved, we instead carry over time the probability of belonging to either regime as $\pi_{1t} + \pi_{2t} = 1$. Finally, we de ne the initial values, both for returns r_0 and the probability of belonging to one of the two regimes π_{10}.

The regime switching model is de ned by the likelihood

$$L(r \mid \mu, \sigma^2, p, r_0, \pi_{10}) = \prod_t \pi_{1t}\,\pi_{1t} + (1 - \pi_{1t})\,\pi_{2t}, \tag{9}$$

$$\text{where } \pi_{1t} = \frac{\pi_{1t}\,\pi_{1t}}{\pi_{1t}\,\pi_{1t} + (1 - \pi_{1t})\,\pi_{2t}},$$

and $\pi_{jt} = p_{j,1} N(r_t; \mu_1, \sigma_1^2) + p_{j,2} N(r_t; \mu_2 + r_{t-1}, \sigma_2^2)$ for $j \in \{0, 1\}$. The prior distributions for the parameters are

$$\mu_1, \mu_2, r_0 \sim N(0, 1); \quad \sigma \sim N^0(1, 0.1); \tag{10}$$

$$\mu_1, \mu_2 \sim C^+(1); \tag{11}$$

$$p_{1,1}, p_{2,2} \sim \text{Beta}(10, 2); \quad \pi_{10} \sim \text{Beta}(2, 2); \tag{12}$$

where N^0 indicates a Gaussian distribution which is truncated at zero and C^+ is the half-Cauchy distribution. Numerical results are shown in Table 3.

The marginal unimodality and somewhat independent correlation structure of the parameters makes this posterior distribution easy to sample from, diagnostic results show the best performance for all algorithms with respect to other models. TESS's learned exible transformation of the target density, allowing it to propose uncorrelated sequential samples, is fundamental for its superior diagnostics. ChEES-HMC outputs the samples with the lowest Stein discrepancy, but since it uses the same step size for all target dimensions it struggles to mix well on the worst-case dimension. On the other hand, a exible transport map is able to capture the covariance structure of the target, allowing fast mixing even on the worst-case dimension. Pair density plots can be found in the Supplementary Material.

4.4 Predator-prey system

We consider a likelihood de ned as a solution of an ODE system, speci cally, the predator-prey system de ned by the Lotka-Volterra equations (Goel et al., 1971),

$$\frac{dp}{dt} = \alpha p - \beta pq, \quad \text{and} \quad \frac{dq}{dt} = -\gamma q + \delta pq, \tag{13}$$

where p and q are the prey and predator populations, respectively. We can solve the ODE system of equations numerically and account for measurement error by modelling the observations as $\log p_t \sim N(\log p(t), \sigma_p^2)$ and $\log q_t \sim N(\log q(t), \sigma_q^2)$ for all $t > 0$. Furthermore $p(0)$ and $q(0)$ are the initial values. Since we cannot analytically solve the system of equations, we approximate its solution using the Runge–Kutta method, adding an approximation error to our likelihood function. Data for the Hudson's Bay historical lynx-hare population are used as observations in the model. The likelihood is de ned as

$$L(p, q \mid \theta) = \prod_t N\left(\begin{matrix}\log p_t \\ \log q_t\end{matrix}; \begin{matrix}\log p(t) \\ \log q(t)\end{matrix}; \begin{matrix}\sigma_p^2 & 0 \\ 0 & \sigma_q^2\end{matrix}\right)$$

where $\theta = (\alpha, \beta, \gamma, \delta, \sigma_p^2, \sigma_q^2, p(0), q(0))$ and $\{p(t), q(t)\}_{t>0}$ are approximate solutions to the Lotka-Volterra system of equations initialized at $(p(0), q(0))$.

http://people.whitman.edu/~hundledr/courses/M250F03/LynxHare.txt

Algorithm	$\hat{\tau}_{max}$		ESS	ESS/chain	ESS/sec	Stein U-stat.	Stein V-stat.
TESS	0.267	0.893	23985	187	985.969	5.120e-02	1.301e-01
MEADS	1.382	1.197	4631	36	319.949	3.066e-01	3.867e-01
ChEES-HMC	3.451	1.825	1855	14	121.756	-8.203e-03	7.073e-02
NUTS	0.282	0.403	22672	177	182.255	2.222e-02	1.009e-01
NeuTra	0.441	1.020	14530	114	209.069	1.092e-01	1.880e-01

Table 3: Regime switching Hidden Markov model. Algorithm diagnostics where $\hat{\tau}_{max}$ is the maximum integrated autocorrelation time over all dimensions; ESS is the corresponding minimum effective sample size. Results are averaged over multiple chains of each sampler, and is the empirical standard deviation of $\hat{\tau}_{max}$ over these runs.

Prior distributions for parameters are

$$; \quad N^{0}(1; 1=2); \quad ; \quad N^{0}(1=20; 1=20); \tag{14}$$

$$\log_{p}; \log_{q} \quad N(\quad 1; 1); \tag{15}$$

$$\log p(0); \log q(0) \quad N(\log 10; 1); \tag{16}$$

where N^{0} is a Gaussian distribution truncated at zero.

This experiment exhibits a situation similar to Section 4.1: gradient methods, without global information on the structure of our target, lack enough information to move efficiently around its rapidly changing correlation structure. On the other hand, TESS captures the global structure of the target using a NF and is able to move purposely around it when sampling. Figure 3 illustrates the contrast: MEADS is unable to converge towards a sensible solution, exploring a region of the target space with large error variance and insignificant initial positions, both for the predator and the prey populations; on the other hand, TESS is able to converge towards reasonable initial populations and concentrate sampling around small error variance. Samples from the other gradient methods give similar results to MEADS. Gradient methods need a learned correlation matrix that captures the global correlation structure of the target and use gradient information to propose large steps locally, while TESS is able to capture both the global correlation and local structure by learning an overconfident transport map, then using this information on a cheap and gradient-free method for sampling.

TESS MEADS

Figure 3: Density plots of the approximate posterior distribution for the initial values and scale parameters from the predator-prey system model, drawn with transport elliptical slice sampling on the left and MEADS on the right.

5 DISCUSSION

In this paper we proposed TESS, an MCMC algorithm that performs dimension independent and gradient-free sampling from any unnormalized target density. We also proposed an adaptive version of our algorithm that learns a non-Gaussian approximation to the target, helping the algorithm explore complex geometries efficiently. TESS is also able to utilize parallel computer architectures to accelerate sampling from posterior distributions. We believe that this will allow practitioners to perform uncertainty quantification of their models with parallel computational resources and little time.

We found that our algorithm is able to outperform gradient-based competitors in a variety of models. However, it is important to develop flexible transport maps and low-variance Monte Carlo approximations of the KL divergence, specially for high-dimensional models. Future work will explore the role of the transport map on the algorithm's efficiency, its efficacy in capturing issues in Bayesian posterior geometries, and develop flexible transport maps for high-dimensional models.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful feedback which has significantly improved the quality of the paper. CN gratefully acknowledges the support of EPSRC grants EP/V022636/1, EP/S00159X/1 and EP/R01860X/1.

References

Iain Murray, Ryan Adams, and David MacKay. Elliptical slice sampling. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*,

pages 541–548. JMLR Workshop and Conference Proceedings, 2010.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. Physics letters B, 195(2):216–222, 1987.

Radford M Neal et al. Mcmc using hamiltonian dynamics. Handbook of Markov Chain Monte Carlo, 2(11):2, 2011.

Peter J Rossky, Jimmie D Doll, and Harold L Friedman. Brownian dynamics as smart monte carlo simulation. The Journal of Chemical Physics, 69(10):4628–4633, 1978.

Ulf Grenander and Michael I Miller. Representations of knowledge in complex systems. Journal of the Royal Statistical Society: Series B (Methodological), 56(4):549–581, 1994.

Seppo Linnainmaa. Taylor expansion of the accumulated rounding error. BIT Numerical Mathematics, 16(2):146–160, 1976.

Matthew D Hoffman, Andrew Gelman, et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. J. Mach. Learn. Res., 15(1):1593–1623, 2014.

John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. Probabilistic programming in python using pymc3. PeerJ Computer Science, 2:e55, 2016.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. Journal of statistical software, 76(1), 2017.

Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul A. Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep universal probabilistic programming. J. Mach. Learn. Res., 20:28:1–28:6, 2019.

Du Phan, Neeraj Pradhan, and Martin Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro. arXiv preprint arXiv:1912.11554, 2019.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. Machine learning, 37(2): 183–233, 1999.

Matthew Hoffman, Alexey Radul, and Pavel Sountsov. An adaptive-mcmc scheme for setting trajectory lengths in hamiltonian monte carlo. In International Conference on Artificial Intelligence and Statistics, pages 3907–3915. PMLR, 2021.

Matthew D Hoffman and Pavel Sountsov. Tuning-free generalized hamiltonian monte carlo. In International Conference on Artificial Intelligence and Statistics, pages 7799–7813. PMLR, 2022.

Alexey Radul, Brian Patton, Dougal Maclaurin, Matthew Hoffman, and Rif A Saurous. Automatically batching control-intensive programs for modern accelerators. Proceedings of Machine Learning and Systems, 2:390–399, 2020.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467, 2016.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc., 2019.

R. Neal. Regression and classification using gaussian process priors. In Bayesian statistics, volume 6, page 475. 1998.

Radford M Neal. Slice sampling. The annals of statistics, 31(3):705–767, 2003.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In International conference on machine learning, pages 1530–1538. PMLR, 2015.

Solomon Kullback and Richard A Leibler. On information and sufficiency. The annals of mathematical statistics, 22 (1):79–86, 1951.

James M. Joyce. Kullback-Leibler Divergence, pages 720–722. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-04898-2. doi: 10.1007/978-3-642-04898-2_327.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. Journal of the American statistical Association, 112(518):859–877, 2017.

Viacheslav Natarovskii, Daniel Rudolf, and Björn Sprungk. Geometric convergence of elliptical slice sampling. In International Conference on Machine Learning, pages 7969–7978. PMLR, 2021.

Leif T Johnson and Charles J Geyer. Variable transformation to obtain geometric ergodicity in the random-walk metropolis algorithm. *The Annals of Statistics*, pages 3050–3076, 2012.

Matthew Hoffman, Pavel Sountsov, Joshua V Dillon, Ian Langmore, Dustin Tran, and Srinivas Vasudevan. Neutralizing bad geometry in hamiltonian monte carlo using neural transport. *arXiv preprint arXiv:1903.03704*, 2019.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979, 2020.

Robert Nishihara, Iain Murray, and Ryan P Adams. Parallel mcmc with generalized elliptical slice sampling. *The Journal of Machine Learning Research*, 15(1):2087–2112, 2014.

Francois Fagan, Jalaj Bhandari, and John P Cunningham. Elliptical slice sampling with expectation propagation. In *UAI*, 2016.

Matthew D Parno and Youssef M Marzouk. Transport map accelerated markov chain monte carlo. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):645–682, 2018.

Tarek A El Moselhy and Youssef M Marzouk. Bayesian inference with optimal maps. *Journal of Computational Physics*, 231(23):7815–7850, 2012.

Youssef Marzouk, Tarek Moselhy, Matthew Parno, and Alessio Spantini. An introduction to sampling via measure transport. *arXiv preprint arXiv:1602.05023*, 2016.

Maria Gorinova, Dave Moore, and Matthew Hoffman. Automatic reparameterisation of probabilistic programs. In *International Conference on Machine Learning*, pages 3648–3657. PMLR, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jackson Gorham and Lester Mackey. Measuring sample quality with kernels. In *International Conference on Machine Learning*, pages 1292–1301. PMLR, 2017.

Carlos M Carvalho, Nicholas G Polson, and James G Scott. Handling sparsity via the horseshoe. In *Artificial Intelligence and Statistics*, pages 73–80. PMLR, 2009.

Omiros Papaspiliopoulos, Gareth O Roberts, and Martin Sköld. A general framework for the parametrization of hierarchical models. *Statistical Science*, pages 59–73, 2007.

Christiane Lemieux. Control variates. *Wiley StatsRef: Statistics Reference Online*, pages 1–8, 2014.

Zdravko Botev and Ad Ridder. Variance reduction. *Wiley statsRef: Statistics reference online*, pages 1–6, 2017.

Narendra S Goel, Samaresh C Maitra, and Elliott W Montroll. On the volterra and other nonlinear models of interacting populations. *Reviews of modern physics*, 43(2): 231, 1971.

Ulli Wolff, Alpha Collaboration, et al. Monte carlo errors with less errors. *Computer Physics Communications*, 156 (2):143–153, 2004.

Qiang Liu, Jason Lee, and Michael Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, pages 276–284. PMLR, 2016.

Charles Stein, Persi Diaconis, Susan Holmes, and Gesine Reinert. Use of exchangeable pairs in the analysis of simulations. *Lecture Notes-Monograph Series*, pages 1–26, 2004.

A ELLIPTICAL SLICE SAMPLING ALGORITHM

Algorithm 3 Elliptical slice sampler (Murray et al., 2010)

Require: $x; L(Dj\ )$
1: $v \quad N(0; I_d)$
2: $w \quad \text{Uniform}(0; 1)$
3: $\log s \quad \log L(Djx) + \log w$
4: $\quad \text{Uniform}(0; 2\ )$
5: $[\ _{min};\ _{max}] \quad [\quad 2\ ;\ ]$
6: $x^0 \quad x\cos\ + v\sin$
7: **if** $\log L(Djx^0) > \log s$ **then**
8: Return x^0
9: **else**
10: **if** $\ < 0$ **then**
11: $\ _{min}$
12: **else**
13: $\ _{max}$
14: **end if**
15: Uniform$(\ _{min};\ _{max})$
16: Go to 6.
17: **end if**

B PROOF

B.1 Proof of Proposition 1

As established in Murray et al. (2010) and Nishihara et al. (2014), the elliptical slice sampler and generalized elliptical sampler target the correct stationary distribution as the algorithm is reversible and produces an irreducible, aperiodic Markov chain.

The same result holds for the TESS algorithm from initial state $u = T^{-1}(x)$ and where $(u; v)$ and $(u^0; v^0)$ represent the initial and accepted transformed parameters of the sampler (steps 1 and 6-7), with s the slice variable (step 3) and $f\ _k g_{k=1}^K$ the parameters representing points in the slice expressed in radians until acceptance at K (steps 4 and 17). Let

$$\ _k^0 = \begin{cases} \ _k \quad _K; & \text{if } k < K \\ \ _K & \text{if } k = K \end{cases}; \tag{17}$$

then by the properties of the elliptical slice sampler, the transformation $(u; v; s; f\ _k g_{k=1}^K) \mapsto (u^0; v^0; s; f\ _k^0 g_{k=1}^K)$ is bijective, preserves volume and $p(f\ _k g_{k=1}^K ju; v; s) = p(f\ _k^0 g_{k=1}^K ju^0; v^0; s)$. Using the uniform density of the slice variable s it is easy to see that $p(u^0; v^0; f\ _k g_{k=1}^K; sju; v) \propto (u)\ (v) = p(u; v; f\ _k^0 g_{k=1}^K; sju^0; v^0) \propto (u^0)\ (v^0)$, and so if $(u; v) \quad \propto (u)\ (v)$ then $(u^0; v^0) \quad \propto (u^0)\ (v^0)$. Finally, as $x = T\ (u)$ we have $(x^0; v^0) \quad (T\ (u^0))j \det\ rT\ (u^0)j\ (v^0) = \quad (x^0)\ (v^0)$.

C DIAGNOSTIC TOOLS CALCULATION DETAILS

Here we describe the calculation of the maximum integrated autocorrelation time $_{\max}$ and Kernelized Stein discrepancy U- and V- statistics used throughout our results. Assume we have as output from chain c a sequence of N samples from our target $x_{c;1}; \ldots; x_{c;N}$, where each sample is on a d dimensional parameter space. Then, compute the integrated

autocorrelation time for dimension $j = 1;\dots;d$ on chain c as

$$\hat{\tau}_{c;j} = \frac{1}{2} + 2 \sum_{t=1}^{N-1} \left(1 - \frac{t}{N}\right) \frac{\hat{C}(t)}{2\hat{C}(0)} \tag{18}$$

$$\hat{C}(t) = \sum_{0<i<N-t} (x_{c;i;j} - \bar{x}_{c;;j})(x_{c;i+t;j} - \bar{x}_{c;;j}) \tag{19}$$

$$\bar{x}_{c;;j} = \frac{1}{N} \sum_{i=1}^{N} x_{c;i;j}. \tag{20}$$

The value $\hat{C}(t)$ for all $t = 1;\dots;N - 1$ is computed by applying the Fourier transform method from Wolff et al. (2004). We then define $\hat{\tau}_{\max}$ and ESS as

$$\hat{\tau}_{\max} = \max_{j=1;\dots;d} \operatorname*{median}_{c=1;\dots;C} \hat{\tau}_{c;j} \tag{21}$$

$$\text{ESS} = \min_{j=1;\dots;d} \operatorname*{median}_{c=1;\dots;C} \frac{N}{2\hat{\tau}_{c;j}}. \tag{22}$$

The Kernelized Stein discrepancy's U- and V-statistics are calculated using the inverse multi-quadratic kernel $k(x;x') = (1 + (x - x')^T(x - x'))^{-\beta}$ with $\beta = 1/2$ on all experiments as

$$\text{U-stat} = \frac{1}{CN(CN-1)} \sum_{c;i} \sum_{c'\neq c;i'\neq i} A_{x'} A^{x'} K(\mathbf{x}_{c;i};\mathbf{x}_{c';i'}) \tag{23}$$

$$\text{V-stat} = \frac{1}{C^2N^2} \sum_{c;i} \sum_{c';i'} A_{x'} A^{x'} K(\mathbf{x}_{c;i};\mathbf{x}_{c';i'}) \tag{24}$$

$$\begin{aligned} A_{x'} A^{x'} K(x;x') = {} & r_x - r_{x'} k(x;x') + r_x k(x;x') - r_{x'} \log p(x') \\ & + r_{x'} k(x;x') - r_x \log p(x) + k(x;x') r_x \log p(x) - r_{x'} \log p(x). \end{aligned} \tag{25}$$

It can be shown that the U-statistic is an unbiased estimate of $\mathbb{E}_{x;x'}[A_{x'} A^{x'} K(x;x')]$ for process generating the samples, while the V-statistic is biased but always non-negative (Liu et al., 2016). If $p = q$ then $\mathbb{E}_{x;x'}[A_{x'} A^{x'} K(x;x')] = 0$ by Stein's identity (Stein et al., 2004).