

Probabilistic Inference of Simulation Parameters via Parallel Differentiable Simulation

Eric Heiden¹, Christopher E. Denniston¹, David Millard¹, Fabio Ramos², Gaurav S. Sukhatme^{1,3}

Abstract— Reproducing real world dynamics in simulation is critical for the development of new control and perception methods. This task typically involves the estimation of simulation parameter distributions from observed rollouts through an inverse inference problem characterized by multi-modality and skewed distributions. We address this challenging problem through a novel Bayesian inference approach that approximates a posterior distribution over simulation parameters given real sensor measurements. By extending the commonly used Gaussian likelihood model for trajectories via the multiple-shooting formulation, our gradient-based particle inference algorithm, Stein Variational Gradient Descent, is able to identify highly nonlinear, underactuated systems. We leverage GPU code generation and differentiable simulation to evaluate the likelihood and its gradient for many particles in parallel. Our algorithm infers nonparametric distributions over simulation parameters more accurately than comparable baselines and handles constraints over parameters efficiently through gradient-based optimization. We evaluate estimation performance on several physical experiments. On an underactuated mechanism where a 7-DOF robot arm excites an object with an unknown mass configuration, we demonstrate how the inference technique can identify symmetries between the parameters and provide highly accurate predictions.

Website: <https://uscresl.github.io/prob-diff-sim>

I. INTRODUCTION

Simulators for robotic systems allow for rapid prototyping and development of algorithms and systems [1], as well as the ability to quickly and cheaply generate training data for reinforcement learning agents and other control algorithms [2]. In order for these models to be useful, the simulator must accurately predict the outcomes of real-world interactions. This is accomplished through both accurately modeling the dynamics of the environment as well as correctly identifying the parameters of such models. In this work, we focus on the latter problem of parameter inference.

Optimization-based approaches have been applied in the past to find simulation parameters that best match the measured trajectories [3], [4]. However, in many systems we encounter in the real world, the dynamics are highly nonlinear, resulting in optimization landscapes fraught with poor local optima where such algorithms get stuck. Global optimization approaches, such as population-based methods,

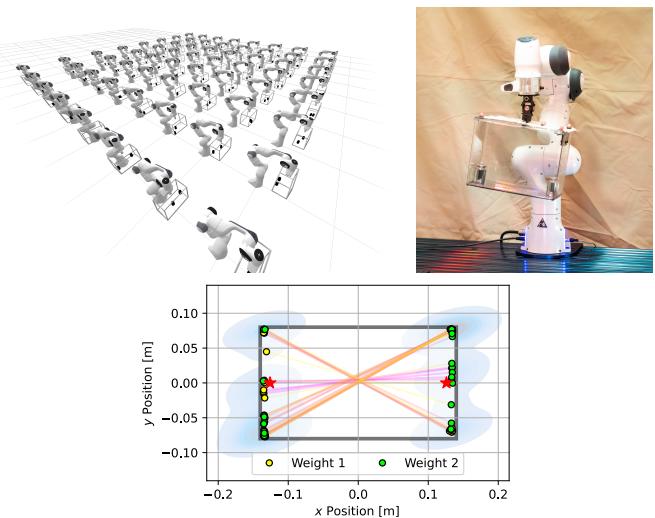


Fig. 1: Panda robot arm shaking a box with two weights in it at random locations in our parallel differentiable simulator (left), physical robot experiment (center), and the inferred particle distribution using our proposed method over the 2D positions of the two weights inside the box (right).

have been applied [5] but are sample inefficient and cannot quantify uncertainty over the predicted parameters.

In this work we follow a probabilistic inference approach and estimate belief distributions over the most likely simulation parameters given the noisy trajectories of observations from the real system. The relationship between the trajectories and the underlying simulation parameters can be highly nonlinear, hampering commonly used inference algorithms. To tackle this, we introduce a multiple-shooting formulation to the parameter estimation process which drastically improves convergence to high-quality solutions. Leveraging GPU-based parallelism of a differentiable simulator allows us to efficiently compute likelihoods and evaluate its gradients over many particles simultaneously. Based on Stein Variational Gradient Descent (SVGD), our gradient-based nonparametric inference method allows us to optimize parameters while respecting constraints on parameter limits and continuity between shooting windows. Through various experiments we demonstrate the improved accuracy and convergence of our approach.

Our contributions are as follows: first, we reformulate the commonly used Gaussian likelihood function through the multiple-shooting strategy to allow for the tractable estimation of simulation parameters from long noisy trajectories. Second, we propose a constrained optimization algorithm

¹Department of Computer Science, University of Southern California, Los Angeles, USA {heiden, cdennist, dmillard, gaurav}@usc.edu

²NVIDIA, Seattle, USA ftozotoramos@nvidia.com

³G.S. Sukhatme holds concurrent appointments as a Professor at USC and as an Amazon Scholar. This paper describes work performed at USC and is not associated with Amazon.

This work was supported by a Google PhD Fellowship and a NASA Space Technology Research Fellowship, grant number 80NSSC19K1182.

for nonparametric variational inference with constraints on parameter limits and shooting defects. Third, we leverage a **fully differentiable simulator** and GPU parallelism to automatically calculate gradients for many particles in parallel. Finally, we validate our system on a simulation parameter estimation problem from real-world data and show that our calculated posteriors are more accurate than comparable algorithms, as well as likelihood-free methods.

II. RELATED WORK

System identification methods for robotics use a dynamics model with often linearly dependent parameters in classical time or frequency domain [6], and solve for these parameters via least-squares methods [7]. Such estimation approaches have been applied, for example, to the identification of inertial parameters of robot arms [8], [9], [10], [6] with time-dependent gear friction [11], or parameters of contact models [12], [13]. Parameter estimation has been studied to determine a minimum set of identifiable inertial parameters [14] and finding exciting trajectories which maximize identifiability [15], [16]. More recently, least-squares approaches have been applied to estimate parameters of nonlinear models, such as the constitutive equations of material models [17], [18], [19], and contact models [20], [21]. In this work, we do not assume a particular type of system to identify, but propose a method for general-purpose differentiable simulators that may combine multiple models whose parameters can influence the dynamics in highly nonlinear ways.

Bayesian methods seek to infer probability distributions over simulation parameters, and have been applied to infer the parameters of dynamical systems in robotic tasks [22], [23], [24] and complex dynamical systems [25]. Our approach is a Bayesian inference algorithm which allows us to include priors to find posterior distributions over simulation parameters. The advantages of Bayesian inference approaches have been shown to be useful in the areas of uncertainty quantification [25], [26], system noise quantification [27] and model parameter inference [28], [29].

Our method is designed for differentiable simulators which have been developed recently for various areas of modeling, such as articulated rigid body dynamics [30], [31], [32], [33], [5], [21], deformables [34], [31], [35], [36], [37] and cloth simulation [38], [31], [33], as well as sensor simulation [39], [40]. Certain physical simulations (e.g. fracture mechanics) may not be analytically differentiable, so that **surrogate gradients** may be necessary [41].

Without assuming access to the system equations, likelihood-free inference approaches, such as approximate Bayesian computation (ABC), have been applied to the inference of complex phenomena [42], [43], [4], [44], [45], [46]. While such approaches do not rely on a simulator to compute the likelihood, our experiments in appendix¹ Sec. 2.4 indicate that the approximated posteriors inferred by likelihood-free

¹We provide more details and results from likelihood-free baselines in the appendix at <https://uscresl.github.io/prob-diff-sim>.

methods are less accurate for **high-dimensional parameter distributions** while requiring significantly more simulation roll-outs as training data.

Domain adaptation techniques have been proposed that close the loop between parameter estimation from real observation and improving policies learned from simulators [3], [4], [47], [48]. Achieving an accurate simulation is typically not the final objective of these methods. Instead, the performance of the learned policy derived from the calibrated simulator is evaluated which does not automatically imply that the simulation is accurate [49]. In this work, we focus solely on the calibration of the simulator where its parameters need to be inferred.

III. FORMULATION

In this work we address the parameter estimation problem via the Bayesian inference methodology. The posterior $p(\theta|D_{\mathcal{X}})$ over simulation parameters $\theta \in \mathbb{R}^M$ and a set of **trajectories** $D_{\mathcal{X}}$ is calculated using Bayes' rule:

$$p(\theta|D_{\mathcal{X}}) \propto p(D_{\mathcal{X}}|\theta)p(\theta),$$

where $p(D_{\mathcal{X}}|\theta)$ is the likelihood distribution and $p(\theta)$ is the prior distribution over the simulation parameters. We aim to approximate the distribution over true parameters $p(\theta^{\text{real}})$, which in general may be intractable to compute. These true parameters θ^{real} generate a set of trajectories $D_{\mathcal{X}}^{\text{real}}$ which may contain some observation noise. We assume that these parameters are **unchanging** during the trajectory and represent physical parameters, such as friction coefficients or link masses.

We assume that each trajectory is a **Hidden Markov Model (HMM)** [50] which has some known initial state, s_0 , and some hidden states s_t , $t \in [1..T]$. These hidden states induce an observation model $p_{\text{obs}}(x_t|s_t)$. In the simulator, we map simulation states to observations via a deterministic observation function $f_{\text{obs}} : s \mapsto x$.

The transition probability $p(s_t|s_{t-1}, \theta)$ of the HMM cannot be directly observed but, in the case of a physics engine, can be approximated by sampling from a distribution of simulation parameters and applying a deterministic simulation step function $f_{\text{step}} : (s, t, \theta) \mapsto s$. In Sec. 2.1 of the appendix, we describe our implementation of f_{step} , the discrete dynamics simulation function. The function f_{sim} rolls out multiple steps via f_{step} to produce a trajectory of T states given a parameter vector θ and an initial state s_0 : $f_{\text{sim}}(\theta, s_0) = [s]_{t=1}^T$. To compute measurements from such state vectors, we use the observation function $f_{\text{obs}} : \mathcal{X} = f_{\text{obs}}([s]_{t=1}^T)$. Finally, we obtain a set of simulated trajectories $D_{\mathcal{X}}^{\text{sim}} = [f_{\text{obs}}(f_{\text{sim}}(\theta, s_0^{\text{real}}))]$ for each initial state s_0^{real} from the trajectories in $D_{\mathcal{X}}^{\text{real}}$. The initial state s_0^{real} from an observed trajectory may be acquired via state estimation techniques, e.g. methods that use inverse kinematics to infer joint positions from tracking measurements.

We aim to minimize the **Kullback-Leibler (KL)** divergence between the trajectories generated from forward simulating our learned parameter particles and the ground-truth trajectories , while taking into account the priors over simulation

parameters:

$$d_{\text{KL}} \left[p(D_{\mathcal{X}}^{\text{sim}} | \theta^{\text{sim}}) p(\theta^{\text{sim}}) \| p(D_{\mathcal{X}}^{\text{real}} | \theta^{\text{real}}) p(\theta^{\text{real}}) \right].$$

We choose the *exclusive KL* divergence instead of the opposite direction since it was shown for our choice of particle-based inference algorithm in [51] that the particles exactly approximate the target measure when an infinitely dimensional functional optimization process minimizes this form of KL divergence.

IV. APPROACH

A. Stein Variational Gradient Descent

A common challenge in statistics and machine learning is the approximation of intractable posterior distributions. In the domain of robotics simulators, the inverse problem of inferring high-dimensional simulation parameters from trajectory observations is often nonlinear and non-unique as there are potentially a number of parameters values that equally well produce simulated roll-outs similar to the real dynamical behavior of the system. This often results in non-Gaussian, multi-modal parameter estimation problems. Markov Chain Monte-Carlo (MCMC) methods are known to be able to find the true distribution, but require an enormous amount of samples to converge, which is exacerbated in high-dimensional parameter spaces. Variational inference approaches, on the other hand, approximate the target distribution by a simpler, tractable distribution, which often does not capture the full posterior over parameters accurately [52].

Our solution based on the Stein Variational Gradient Descent (SVGD) algorithm [53] that approximates the posterior distribution $p(\theta|D_{\mathcal{X}}) = \frac{p(D_{\mathcal{X}}|\theta)p(\theta)}{\int p(D_{\mathcal{X}}|\theta)p(\theta)d\theta}$ by a set of particles $q(\theta|D_{\mathcal{X}}) = \frac{1}{N} \sum_{i=1}^N \delta(\theta_i - \theta)$ where $\delta(\cdot)$ is the Dirac delta function, and makes use of differentiable likelihood and prior functions to be efficient. SVGD avoids the computation of the intractable marginal likelihood in the denominator by only requiring the computation of $\nabla_{\theta} \log p(\theta|D_{\mathcal{X}}) = \frac{\nabla_{\theta} p(\theta|D_{\mathcal{X}})}{p(\theta|D_{\mathcal{X}})}$ which is independent of the normalization constant. The particles are adjusted according to the steepest descent direction to reduce the KL divergence in a reproducing kernel Hilbert space (RKHS) between the current set of particles representing $q(\theta|D_{\mathcal{X}})$ and the target $p(\theta|D_{\mathcal{X}})$.

As derived in [53], the set of particles $\{\theta_i\}_{i=1}^N$ is updated by the following function:

$$\theta_i \leftarrow \theta_i + \epsilon \phi(\theta_i), \quad (1)$$

$$\phi(\cdot) = \frac{1}{N} \sum_{j=1}^N [k(\theta_j, \theta) \nabla_{\theta_j} \log p(D_{\mathcal{X}}|\theta_j) p(\theta_j) + \nabla_{\theta_j} k(\theta_j, \theta)],$$

where $k(\cdot, \cdot) : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}$ is a positive definite kernel and ϵ is the step size. In this work, we use the radial basis function kernel, which is a common choice for SVGD [53] due to its smoothness and infinite differentiability. To tune the kernel bandwidth, we adopt the median heuristic, which has been shown to provide robust performance on large learning problems [54].

SVGD requires that the likelihood function be differentiable in order to calculate Eq. 1, which in turn requires

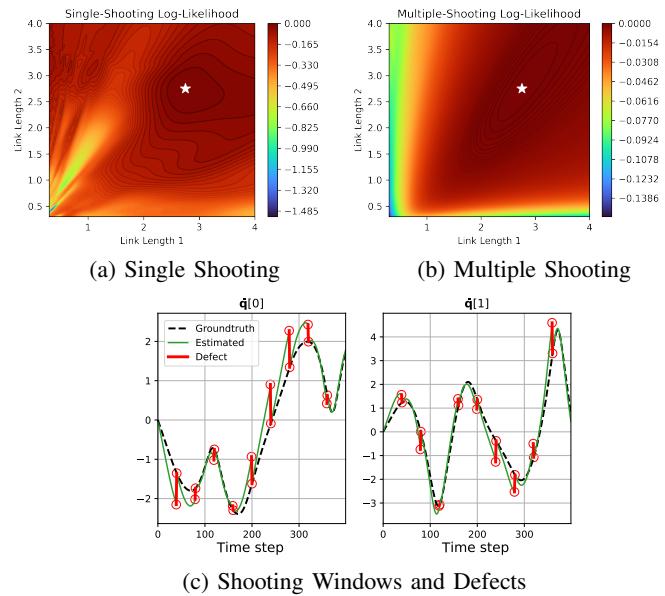


Fig. 2: The two heatmaps plots on the left show the landscape of the log-likelihood function for an inference problem where the two link lengths of a double pendulum are estimated (ground-truth parameters indicated by a white star). In (a), the likelihood is evaluated over a 400-step trajectory; in (b), the trajectory is split into 10 shooting windows and the likelihood is computed via Eq. 6. In (c), the shooting intervals and defects are visualized for an exemplary parameter guess.

that f_{sim} and f_{obs} be differentiable. To enable this, we use a fully-differentiable simulator, observation function and likelihood function, and leverage automatic differentiation with code generation to generate CUDA kernel code [55]. Because of this CUDA code generation, we are able to calculate $\nabla_{\theta_j} \log p(D_{\mathcal{X}}|\theta) p(\theta)$ for each particle in parallel on the GPU.

B. Likelihood Model for Trajectories

We define $p(\mathcal{X}^{\text{sim}} | \mathcal{X}^{\text{real}})$ as the probability of an individual trajectory \mathcal{X}^{sim} simulated from parameters θ with respect to a matched ground-truth trajectory $\mathcal{X}^{\text{real}}$. Following the HMM assumption from Sec. III, we treat $p(\mathcal{X}^{\text{sim}} | \mathcal{X}^{\text{real}})$ as the product of probabilities over observations, shown in Eq. 2. This assumption is justified because the next state s_{t+1} is fully determined by the position q_t and velocity \dot{q}_t of the current state s_t in articulated rigid body dynamics (see Sec. 2.1 of the website). To directly compare observations we use a Gaussian likelihood:

$$\begin{aligned} p(\mathcal{X}^{\text{real}} | \mathcal{X}^{\text{sim}}) &= \prod_{t \in T} p(x_t^{\text{real}}, x_t^{\text{sim}}) \\ &= \prod_{t \in T} \mathcal{N}(x_t^{\text{real}} | x_t^{\text{sim}}, \sigma_{\text{obs}}^2). \end{aligned} \quad (2)$$

This likelihood model for observations is used to compute the likelihood for a trajectory:

$$\begin{aligned} p_{ss}(\mathcal{X}^{\text{real}} | \theta) &= p(\mathcal{X}^{\text{real}} | f_{\text{obs}}(f_{\text{sim}}(\theta, s_0^{\text{real}}))) \\ &= p(\mathcal{X}^{\text{real}} | \mathcal{X}^{\text{sim}}), \end{aligned} \quad (3)$$

where s_0^{real} is (an estimate of) the first state of $\mathcal{X}^{\text{real}}$. This formulation is known as a *single-shooting* estimation problem, where a trajectory is compared against another only by varying the initial conditions of the modeled system. To evaluate the likelihood for a collection of ground-truth trajectories, $D_{\mathcal{X}}^{\text{real}}$, we use an equally weighted Gaussian Mixture Model likelihood:

$$p_{\text{obs}}(D_{\mathcal{X}}^{\text{real}} | \theta) = \sum_{x^{\text{real}} \in D_{\mathcal{X}}^{\text{real}}} p_{ss}(\mathcal{X}^{\text{real}} | \theta). \quad (4)$$

C. Multiple Shooting

Estimating parameters from long trajectories can prove difficult in the face of observation noise, and for systems in which small changes in initial conditions produce large changes in trajectories, potentially resulting in poor local optima [56]. We adopt the *multiple-shooting* method which significantly improves the convergence and stability of the estimation process. Multiple shooting has been applied in system identification problems [57] and biochemistry [58].

Multiple shooting divides up the trajectory into n_s shooting windows over which the likelihood is computed. To be able to simulate such shooting windows, we require their start states s^s to be available for f_{sim} to generate a shooting window trajectory. Since we only assume access to the first true state s_0^{real} from the real trajectory $\mathcal{X}^{\text{real}}$, we augment the parameter vector θ by the start states of the shooting windows, which we refer to as *shooting variables* s_t^s (for $t = h, 2h, \dots, n_s \cdot h$). We define an augmented parameter vector as $\bar{\theta} = [\theta \ s_h^s \ \dots \ s_{n_s \cdot h}^s]$. A shooting window of h time steps starting from time t is then simulated via $\mathcal{X}_t = f_{\text{obs}}(f_{\text{sim}}(\theta, s_t^s)[0 : h])$, where $[i : j]$ denotes a selection operation of the sub-array between the indices i and j . Analogous to Eq. 2, we evaluate the likelihood $p(\mathcal{X}^{\text{real}}[t:t+h] | \mathcal{X}_t^{\text{sim}})$ for a single shooting window as a product of state-wise likelihoods.

To impose continuity between the shooting windows, *defect constraints* are imposed as a Gaussian likelihood term between the last simulated state s_t from the previous shooting window at time t and the shooting variable s_t^s at time t :

$$p_{\text{def}}(s_t^s, s_t) = \mathcal{N}(s_t^s | s_t, \sigma_{\text{def}}^2) \quad t \in [h, 2h, \dots] \quad (5)$$

where σ_{def}^2 is a small variance so that the MCMC samplers adhere to the constraint. Including the defect likelihood allows the extension of the likelihood defined in Eq. 3 to a multiple-shooting scenario:

$$p_{ms}(\mathcal{X}^{\text{real}} | \bar{\theta}) = \prod_{t \in H} p_{\text{def}}(s_t^s, s_t) p(\mathcal{X}^{\text{real}}[t:t+h] | \mathcal{X}_t^{\text{sim}}), \\ s_0^s = s_0^{\text{real}} \quad H = [0, h, 2h, \dots] \quad (6)$$

As for the single-shooting case, Eq. 4 with p_{ms} as the trajectory-wise likelihood function gives the likelihood p_{obs} for a set of trajectories.

In Fig. 2, we provide a parameter estimation example where the two link lengths of a double pendulum must be inferred. The single-shooting likelihood from Eq. 3 (shown in Fig. 2a) exhibits a rugged landscape where many estimation

algorithms will require numerous samples to escape from poor local optima, or a much finer choice of parameter prior to limit the search space. The multiple-shooting likelihood (shown in Fig. 2) is significantly smoother and therefore easier to optimize.

D. Parameter Limits as a Uniform Prior

Simulators may not be able to handle simulating trajectories from any given parameter and it is often useful to enforce some limits on the parameters. To model this, we define a uniform prior distribution on the parameter settings $p_{\text{lim}}(\theta) = \prod_{i=1}^M U(\theta_i | \theta_{\min_i}, \theta_{\max_i})$, where $\theta_{\min_i}, \theta_{\max_i}$ denote the upper and lower limits of parameter dimension i .

E. Constrained Optimization for SVGD

Directly optimizing SVGD for the unnormalized posterior on long trajectories, with a uniform prior, can be difficult for gradient based optimizers. The uniform prior has discontinuities at the extremities, effectively imposing constraints, which produce non-differentiable regions in the parameter space domain. We propose an alternative solution to deal with this problem and treat SVGD as a constrained optimization on p_{obs} with p_{def} and p_{lim} as constraints.

A popular method of constrained optimization is the Modified Differential Multiplier Method (MDMM) [59] which augments the cost function to penalize constraint violations. In contrast to the basic penalty method, MDMM uses Lagrange multipliers in place of constant penalty coefficients that are updated automatically during the optimization:

$$\begin{aligned} & \text{maximize} && \log p_{\text{obs}}(D_{\mathcal{X}} = D_{\mathcal{X}}^{\text{real}} | \theta) \\ & \text{s.t.} && g(\bar{\theta}) = 0. \end{aligned} \quad (7)$$

MDMM formulates this setup into an unconstrained minimization problem by introducing Lagrange multipliers λ (initialized with zero):

$$\mathcal{L}_c(\bar{\theta}, \lambda) = -\log p_{\text{obs}}(D_{\mathcal{X}} = D_{\mathcal{X}}^{\text{real}} | \theta) + \lambda g(\bar{\theta}) + \frac{c}{2} [g(\bar{\theta})]^2, \quad (8)$$

where $c > 0$ is a constant damping factor that improves convergence in gradient descent algorithms. To accommodate these Lagrange multipliers per-particle we again extend the parameter set $(\bar{\theta})$ introduced in Sec. IV-C to store the multiple shooting variables and Lagrange multipliers, $\bar{\theta} = (\theta, s^s, \lambda_{\text{def}}, \lambda_{\text{lim}})$. The following update equations for θ and λ are used to minimize Eq. 8:

$$\begin{aligned} \dot{\theta} &= \frac{\partial \log p_{\text{obs}}(D_{\mathcal{X}}^{\text{real}} | \theta)}{\partial \theta} - \lambda \frac{\partial g(\bar{\theta})}{\partial \theta} - cg(\bar{\theta}) \frac{\partial g(\bar{\theta})}{\partial \theta} \\ \dot{\lambda} &= g(\bar{\theta}) \end{aligned}$$

We include parameter limit priors from Sec. IV-D as the following equality constraints (where $\text{clamp}(x, a, b)$ clips the value x to the interval $[a, b]$): $g_{\text{lim}}(\theta) = \text{clamp}(\theta, \theta_{\min}, \theta_{\max}) - \theta$. Other constraints are the defect constraints from Eq. 5 which are included as equality constraints as well: $g_{\text{def}}(\theta, s_t^s) = \log p_{\text{def}}(s_t^s, s_t) = \|s_t^s - s_t\|^2 / \sigma_{\text{def}}^2$.

The overall procedure of our Constrained SVGD (CSVGD) algorithm is given in Algorithm 1.

Algorithm 1: Constrained SVGD

Inputs: differentiable simulator $f_{\text{sim}} : (\theta, \mathbf{s}_0) \mapsto [\mathbf{s}]$, observation function $f_{\text{obs}} : \mathbf{s} \mapsto \mathbf{x}$, start states \mathbf{s}_0^i for each ground-truth trajectory $\mathcal{X}_i^{\text{real}} \in D_{\mathcal{X}}^{\text{real}}$, learning rate scheduler (e.g. Adam), kernel choice (e.g. RBF)

```

for  $i = 1 \dots \text{max\_iterations}$  do
    Roll out simulated observations
     $D_{\mathcal{X}}^{\text{sim}} = [f_{\text{obs}}(f_{\text{sim}}(\theta, \mathbf{s}_0^{\text{real}})) \forall \mathcal{X}^{\text{real}} \in D_{\mathcal{X}}^{\text{real}}]$ 
    Compute  $\phi(\theta)$  via Eq. 1 and
     $\log p_{\text{obs}}(D_{\mathcal{X}} = D_{\mathcal{X}}^{\text{real}} \mid \theta)$ 
    Update  $\theta$  via  $\dot{\theta} = \phi(\theta) - \lambda_{\text{lim}} \frac{\partial g_{\text{lim}}}{\partial \theta} - c g_{\text{lim}} \frac{\partial g_{\text{lim}}}{\partial \theta} - \lambda_{\text{def}} \frac{\partial g_{\text{def}}}{\partial \theta} - c g_{\text{def}} \frac{\partial g_{\text{def}}}{\partial \theta}$ 
    Update  $\lambda_{\text{lim}}, \lambda_{\text{def}}, \mathbf{s}_t^s$  via
     $\lambda_{\text{lim}} = g_{\text{lim}}(\theta), \lambda_{\text{def}} = g_{\text{def}}(\theta), \mathbf{s}_t^s = g_{\text{def}}(\theta, \mathbf{s}_t^s)$ 
    for  $t \in [h, 2h, \dots]$ 
end
return  $\theta$ 

```

F. Performance Metrics for Particle Distributions

In most cases we do not know the underlying ground-truth parameter distribution $p(\theta^{\text{real}})$, and only have access to a finite set of ground-truth trajectories. Therefore, we measure the discrepancy between trajectories rolled out from the estimated parameter distribution, $D_{\mathcal{X}}^{\text{sim}}$, and the reference trajectories, $D_{\mathcal{X}}^{\text{real}}$.

One measure, the KL divergence, is the expected value of the log likelihood ratio between two distributions. Although the KL divergence cannot be calculated from samples of continuous distributions, methods have been developed to estimate it from particle distributions using the k -nearest neighbors distance [60]. The KL divergence is non-symmetric and this estimate can be poor in two situations. The first is estimating $d_{\text{KL}}(D_{\mathcal{X}}^{\text{sim}} \parallel D_{\mathcal{X}}^{\text{real}})$ when the particles are all very close to one trajectory, causing a low estimated divergence, yet the posterior is of poor quality. The opposite can happen when the particles in the posterior are overly spread out when estimating $d_{\text{KL}}(D_{\mathcal{X}}^{\text{real}} \parallel D_{\mathcal{X}}^{\text{sim}})$.

We additionally measure the maximum mean discrepancy (MMD) [61], which is a metric used to determine if two sets of samples are drawn from the same distribution by calculating the square of the distance between the embedding of the two distributions in an RKHS.

V. EXPERIMENTS

We compare our method against commonly used parameter estimation baselines. As an algorithm comparable to our particle-based approach, we use the Cross Entropy Method (CEM) [62]. In addition, we evaluate the Markov chain Monte-Carlo techniques Emcee [63], Stochastic Gradient Langevin Dynamics (SGLD) [64], and the No-U-Turn-Sampler (NUTS) [65], which is an adaptive variant of the gradient-based Hamiltonian MC algorithm. For Emcee, we use a parallel sampler that implements the “stretch move” ensemble method [66]. For BayesSim, we report results from the best performing instantiation using a mixture density

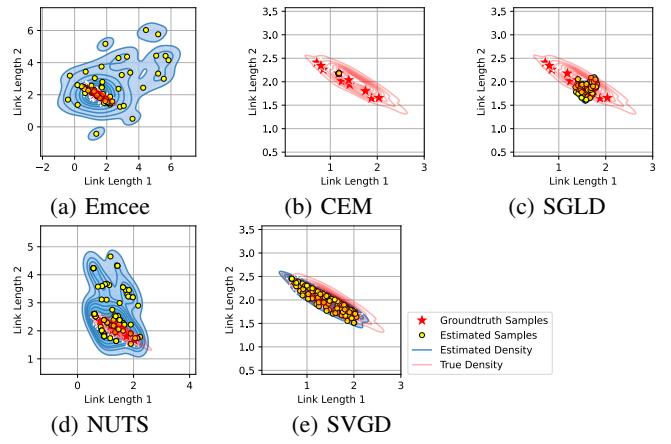


Fig. 3: Estimated posterior distributions from synthetic data generated from a known multivariate Gaussian distribution (Sec. V-A). The 100 last samples were drawn from the Markov chains sampled by Emcee, SGLD, and NUTS, while CEM and SVGD used 100 particles.

random Fourier features model (MDRFF) with Matérn kernel in Tab. I. We present further details and extended results from our experiments on our project website.

A. Parameter Estimation Accuracy

We create a synthetic dataset of 10 trajectories from a double pendulum which we simulate by varying its two link lengths. These two uniquely identifiable [13] parameters are drawn from a Gaussian distribution with a mean of (1.5 m, 2 m) and a full covariance matrix (density visualized by the red contour lines in Fig. 3). We show the evolution of the consistency metrics in Fig. 4. Trajectories generated by evaluating the posteriors of the compared methods are compared against 50 test trajectories rolled out from the ground-truth parameter distribution.

We find that SVGD produces a density very close to the density that matches the principal axes of the ground-truth posterior, and outperforms the other methods in all metrics except log likelihood. CEM collapses to a single high-probability estimate but does not accurately represent the full posterior, as can be seen in Fig. 4d being maximal quickly but poor performance on the other metrics which measure the spread of the posterior. Emcee and NUTS represent the spread of the posterior but do not sharply capture the high-likelihood areas, shown by their good performance in Fig. 4b, Fig. 4b and Fig. 4c. SGLD captures a small amount of the posterior around the high likelihood points but does not fully approximate the posterior.

B. Identify Real-world Double Pendulum

We leverage the dataset from [67] containing trajectories from a physical double pendulum. While in our previous synthetic data experiment the parameter space was reduced to only the two link lengths, we now define 11 parameters to estimate. The parameters for each link are the mass, inertia I_{xx} , the center of mass in the x and y direction, and the joint friction. We also estimate the length of the second link. Note that parameters, such as the length of the first link, are not explicitly included since they are captured by the remaining

Experiment	Metric	Emcee	CEM	SGLD	NUTS	SVGD	BayesSim	CSVGD (Ours)
Double Pendulum	$d_{KL}(D_X^{\text{real}} \parallel D_X^{\text{sim}})$	8542.2466	8911.1798	8788.0962	9196.7461	8803.5683	8818.1830	5204.5336
	$d_{KL}(D_X^{\text{sim}} \parallel D_X^{\text{real}})$	4060.6312	8549.5927	7876.0310	6432.2131	10283.6659	3794.9873	2773.1751
	MMD	1.1365	0.9687	2.1220	0.5371	0.7177	0.6110	0.0366
Panda Arm	$\log p_{\text{obs}}(D_X^{\text{real}} \parallel D_X^{\text{sim}})$	-16.1185	-17.3331	-17.3869	-17.9809	-17.7611	-17.6395	-15.1671

TABLE I: Consistency metrics of the posterior distributions approximated by the different estimation algorithms. Each metric is calculated across simulated and real trajectories. Lower is better on all metrics except $\log p_{\text{obs}}(D_X^{\text{real}} \parallel D_X^{\text{sim}})$.

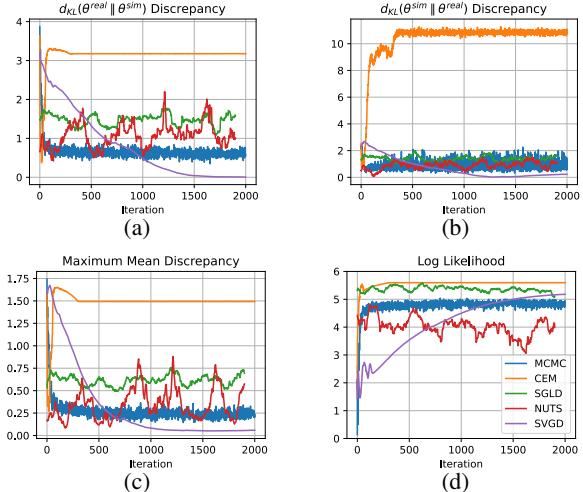


Fig. 4: Accuracy metrics for the estimated parameter posteriors shown in Fig. 3. The estimations were done on the synthetic dataset of a multivariate Gaussian over parameters (Sec. V-A). Fig. 4d shows the single-shooting likelihood using the equation described in Eq. 3.

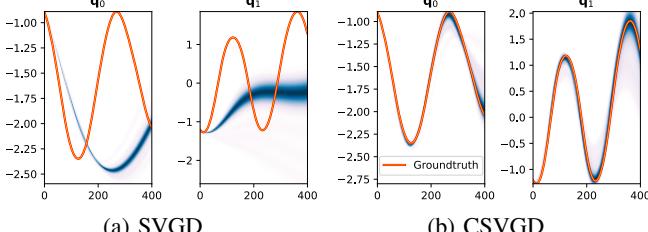


Fig. 5: Trajectory density plots (only joint positions \mathbf{q} are shown) obtained by simulating the particle distribution found by SVGD with the single-shooting likelihood (Fig. 5a) and the multiple-shooting likelihood (Fig. 5b) on the real-world double pendulum (Sec. V-B).

parameters, which we validated through sensitivity analysis. Like before, the state space is completely measurable, i.e. $\mathbf{s} \approx \mathbf{x}$, except for observation noise.

In this experiment we find that CSVGD outperforms all other methods in KL divergence (both ways) as well as MMD, shown in Tab. I. We believe this is because of the complex relationship between the parameters of each link and the resulting observations. This introduces many local minima which are hard to escape from (see Fig. 5). The multiple-shooting likelihood improves the convergence significantly by simplifying the optimization landscape.

C. Identify Inertia of an Articulated Rigid Object

In our final experiment, we investigate a more complicated physical system which is underactuated. Through an

uncontrollable universal joint, we attach an acrylic box to the end-effector of a physical 7-DOF Franka Emika Panda robot arm. We fix two 500 g weights at the bottom inside the box, and prescribe a trajectory which the robot executes through a PD controller. By tracking the motion of the box via a VICON motion capture system, we aim to identify the 2D locations of the two weights (see Fig. 1). The system only has access to the proprioceptive measurements from the robot arm (seven joint positions and velocities), as well as the 3D pose of the box at each time step. In the first phase, we identify the inertia properties of the empty box, as well as the joint friction parameters of the universal joint from real-robot trajectories of shaking an empty box. Given our best estimate, in the actual parameter estimation setup we infer the two planar positions of the weights.

The particles from SVGD and CSVGD quickly converge in a way that the two weights are aligned opposed to each other. If the weights were not at locations symmetrical about the center, the box would tilt and yield a large discrepancy to the real observations. MCMC, on the other hand, even after more than ten times the number of iterations, only rarely approaches configurations in which the box remains balanced. In Fig. 1 (right) we visualize the posterior over weight locations found by CSVGD (blue shade). The found symmetries are clearly visible when we draw lines (orange) between the inferred positions of both weights (yellow, green), while the true weight locations (red) are contained in the approximated distribution. As can be seen in Tab. I, the log likelihood is maximized by CSVGD. The results indicate the ability of our method to accurately model difficult posteriors over complex trajectories because of the symmetries underlying the simulation parameters.

VI. CONCLUSION

We have presented Constrained Stein Variational Gradient Descent (CSVGD), a new method for estimating the distribution over simulation parameters that leverages Bayesian inference and parallel, differentiable simulators. By segmenting the trajectory into multiple shooting windows via hard defect constraints, and effectively using the likelihood gradient, CSVGD produces more accurate posteriors and exhibits improved convergence over previous estimation algorithms.

In future work, we plan to leverage the probabilistic predictions from our simulator for uncertainty-aware control applications. Similar to [68], the particle-based uncertainty information can be leveraged by a model-predictive controller that takes into account the multi-modality of future outcomes.

REFERENCES

- [1] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, Sep. 2004, pp. 2149–2154 vol.3.
- [2] O. M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, Jan. 2020, publisher: SAGE Publications Ltd STM. [Online]. Available: <https://doi.org/10.1177/0278364919887447>
- [3] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8973–8979.
- [4] F. Ramos, R. Possas, and D. Fox, "BayesSim: Adaptive domain randomization via probabilistic inference for robotics simulators," in *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, June 2019.
- [5] E. Heiden, D. Millard, E. Coumans, Y. Sheng, and G. S. Sukhatme, "NeuralSim: Augmenting differentiable simulators with neural networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021. [Online]. Available: <https://github.com/google-research/tiny-differentiable-simulator>
- [6] P. Vandajon, M. Gautier, and P. Desbats, "Identification of robots inertial parameters by means of spectrum analysis," in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 3, 1995, pp. 3033–3038 vol.3.
- [7] K. R. Kozłowski, *Modelling and Identification in Robotics*, ser. Advances in Industrial Control. London: Springer-Verlag, 1998. [Online]. Available: <https://www.springer.com/gp/book/9783540762409>
- [8] P. K. Khosla and T. Kanade, "Parameter identification of robot dynamics," in *IEEE Conference on Decision and Control*, Dec. 1985, pp. 1754–1760.
- [9] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 101–119, 1986. [Online]. Available: <https://doi.org/10.1177/027836498600500306>
- [10] F. Caccavale and P. Chiacchio, "Identification of Dynamic Parameters for a Conventional Industrial Manipulator," *IFAC Proceedings Volumes*, vol. 27, no. 8, pp. 871–876, Jul. 1994. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667017478190>
- [11] M. Grotjahn, M. Daemi, and B. Heimann, "Friction and rigid body identification of robot dynamics," *International Journal of Solids and Structures*, vol. 38, no. 10, pp. 1889–1902, 2001. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0020768300001414>
- [12] D. Verscheure, I. Sharf, H. Bruyninckx, J. Swevers, and J. D. Schutter, "Identification of contact parameters from stiff multi-point contact robotic operations," *The International Journal of Robotics Research*, vol. 29, no. 4, pp. 367–385, 2010. [Online]. Available: <https://doi.org/10.1177/0278364909336805>
- [13] N. Fazeli, R. Tedrake, and A. Rodriguez, "Identifiability analysis of planar rigid-body frictional contact," in *Robotics Research*. Springer, 2018, pp. 665–682.
- [14] M. Gautier and W. Khalil, "Direct calculation of minimum set of inertial parameters of serial robots," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 368–373, Jun. 1990, conference Name: IEEE Transactions on Robotics and Automation.
- [15] G. Antonelli, F. Caccavale, and P. Chiacchio, "A systematic procedure for the identification of dynamic parameters of robot manipulators," *Robotica*, vol. 17, no. 4, pp. 427–435, Jul. 1999, publisher: Cambridge University Press.
- [16] M. Gautier and W. Khalil, "Exciting trajectories for the identification of base inertial parameters of robots," in *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*, Dec. 1991, pp. 494–499 vol.1.
- [17] R. Mahnken, "Identification of material parameters for constitutive equations," *Encyclopedia of Computational Mechanics Second Edition*, pp. 1–21, 2017.
- [18] D. Hahn, P. Banzet, J. M. Bern, and S. Coros, "Real2sim: Visco-elastic parameter estimation from dynamic motion," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–13, 2019.
- [19] Y. S. Narang, K. Van Wyk, A. Mousavian, and D. Fox, "Interpreting and Predicting Tactile Signals via a Physics-Based and Data-Driven Framework," *arXiv:2006.03777 [cs]*, Jun. 2020, arXiv: 2006.03777. [Online]. Available: <http://arxiv.org/abs/2006.03777>
- [20] S. Kolev and E. Todorov, "Physically consistent state estimation and system identification for contacts," *International Conference on Humanoid Robots*, pp. 1036–1043, 2015.
- [21] Q. Lelidec, I. Kalevatykh, I. Laptev, C. Schmid, and J. Carpentier, "Differentiable simulation for physical system identification," *IEEE Robotics and Automation Letters*, pp. 1–1, 2021.
- [22] Y. Wang, H. Wu, and H. Handros, "Markov Chain Monte Carlo (MCMC) methods for parameter estimation of a novel hybrid redundant robot," *Fusion Engineering and Design*, vol. 86, no. 9, pp. 1863–1867, Oct. 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0920379611000743>
- [23] F. Muratore, C. Eilers, M. Gienger, and J. Peters, "Bayesian domain randomization for sim-to-real transfer," *ArXiv*, vol. abs/2003.02471, 2020.
- [24] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohéz, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *ArXiv*, vol. abs/1804.10332, 2018.
- [25] B. Ninness and S. Henriksen, "Bayesian system identification via Markov chain Monte Carlo techniques," *Automatica*, vol. 46, no. 1, pp. 40–51, Jan. 2010. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0005109809004762>
- [26] V. Peterka, "Bayesian approach to system identification," in *Trends and Progress in System Identification*, P. Eykhoff, Ed. Pergamon, 1981, pp. 239–304. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780080256832500132>
- [27] J.-A. Ting, A. D'Souza, and S. Schaal, "Bayesian robot system identification with input and output noise," *Neural Networks*, vol. 24, no. 1, pp. 99–108, Jan. 2011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608010001620>
- [28] S. S. Qian, C. A. Stow, and M. E. Borsuk, "On monte carlo methods for bayesian inference," *Ecological Modelling*, vol. 159, no. 2, pp. 269–277, 2003. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0304380002002995>
- [29] K. Cranmer, J. Brehmer, and G. Louppe, "The frontier of simulation-based inference," *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30 055–30 062, 2020. [Online]. Available: <https://www.pnas.org/content/117/48/30055>
- [30] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J. Z. Kolter, "End-to-end differentiable physics for learning and control," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 7178–7189. [Online]. Available: <http://papers.nips.cc/paper/7948-end-to-end-differentiable-physics-for-learning-and-control.pdf>
- [31] Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand, "DiffTaichi: Differentiable programming for physical simulation," *ICLR*, 2020.
- [32] M. Geilinger, D. Hahn, J. Zehnder, M. Bächer, B. Thomaszewski, and S. Coros, "ADD: Analytically differentiable dynamics for multi-body systems with frictional contact," *ACM Trans. Graph.*, vol. 39, no. 6, Nov. 2020. [Online]. Available: <https://doi.org/10.1145/3414685.3417766>
- [33] Y.-L. Qiao, J. Liang, V. Koltun, and M. C. Lin, "Scalable differentiable physics for learning and control," in *ICML*, 2020.
- [34] Y. Hu, J. Liu, A. Spielberg, J. B. Tenenbaum, W. T. Freeman, J. Wu, D. Rus, and W. Matusik, "ChainQueen: A real-time differentiable physical simulator for soft robotics," *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2019.
- [35] K. M. Jatavallabhula, M. Macklin, F. Golemo, V. Voleti, L. Petrini, M. Weiss, B. Considine, J. Parent-Levesque, K. Xie, K. Erleben, L. Paull, F. Shkurti, D. Nowrouzezahrai, and S. Fidler, "gradSim: Differentiable simulation for system identification and visuomotor control," *International Conference on Learning Representations (ICLR)*, 2021. [Online]. Available: https://openreview.net/forum?id=c_E8kFWfhpo
- [36] E. Heiden, M. Macklin, Y. S. Narang, D. Fox, A. Garg, and F. Ramos, "DiSECt: A differentiable simulation engine for autonomous robotic cutting," *Robotics: Science and Systems*, 2021.

- [37] Z. Huang, Y. Hu, T. Du, S. Zhou, H. Su, J. Tenenbaum, and C. Gan, "PlasticineLab: A soft-body manipulation benchmark with differentiable physics," *ArXiv*, vol. abs/2104.03311, 2021.
- [38] J. Liang, M. Lin, and V. Koltun, "Differentiable cloth simulation for inverse problems," in *Advances in Neural Information Processing Systems*, 2019, pp. 771–780.
- [39] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob, "Mitsuba 2: A retargetable forward and inverse renderer," *Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, vol. 38, no. 6, Nov. 2019.
- [40] E. Heiden, Z. Liu, R. K. Ramachandran, and G. S. Sukhatme, "Physics-based simulation of continuous-wave LIDAR for localization, calibration and tracking," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.
- [41] J. Han and Q. Liu, "Stein variational gradient descent without gradient," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1900–1908.
- [42] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. Stumpf, "Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems," *Journal of The Royal Society Interface*, vol. 6, no. 31, p. 187–202, Jul 2008.
- [43] G. Papamakarios and I. Murray, "Fast epsilon-free inference of simulation models with Bayesian conditional density estimation," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.
- [44] K. Hsu and F. Ramos, "Bayesian learning of conditional kernel mean embeddings for automatic likelihood-free inference," in *Proceedings of Machine Learning Research*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and M. Sugiyama, Eds., vol. 89, 16–18 Apr 2019, pp. 2631–2640.
- [45] C. Matl, Y. Narang, R. Bajcsy, F. Ramos, and D. Fox, "Inferring the Material Properties of Granular Media for Robotic Tasks," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 2770–2777, iSSN: 2577-087X.
- [46] C. Matl, Y. Narang, D. Fox, R. Bajcsy, and F. Ramos, "STReSSD: Sim-To-Real from Sound for Stochastic Dynamics," *arXiv:2011.03136 [cs]*, Nov. 2020, arXiv: 2011.03136. [Online]. Available: <http://arxiv.org/abs/2011.03136>
- [47] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, "Active domain randomization," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 1162–1176. [Online]. Available: <https://proceedings.mlr.press/v100/mehta20a.html>
- [48] Y. Du, O. Watkins, T. Darrell, P. Abbeel, and D. Pathak, "Auto-Tuned Sim-to-Real Transfer," *arXiv:2104.07662 [cs]*, May 2021, arXiv: 2104.07662. [Online]. Available: <http://arxiv.org/abs/2104.07662>
- [49] N. O. Lambert, B. Amos, O. Yadan, and R. Calandra, "Objective mismatch in model-based reinforcement learning," in *Conference on Learning for Dynamics and Control (L4DC)*, ser. Proceedings of Machine Learning Research, A. M. Bayen, A. Jadbabaie, G. J. Pappas, P. A. Parrilo, B. Recht, C. J. Tomlin, and M. N. Zeilinger, Eds., vol. 120. PMLR, 2020, pp. 761–770. [Online]. Available: <http://proceedings.mlr.press/v120/lambert20a.html>
- [50] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. USA: Prentice Hall Press, 2009.
- [51] Q. Liu, "Stein variational gradient descent as gradient flow," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/file/17ed8abedc255908be746d245e50263a-Paper.pdf>
- [52] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational Inference: A Review for Statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, Apr. 2017, publisher: Taylor & Francis eprint: <https://doi.org/10.1080/01621459.2017.1285773>. [Online]. Available: <https://doi.org/10.1080/01621459.2017.1285773>
- [53] Q. Liu and D. Wang, "Stein variational gradient descent: A general purpose bayesian inference algorithm," *Advances in Neural Information Processing Systems*, vol. 29, 2016.
- [54] D. Garreau, W. Jitkrittum, and M. Kanagawa, "Large sample analysis of the median heuristic," 2018.
- [55] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Addison-Wesley Professional, Jul. 2010.
- [56] O. Aydogmus and A. H. TOR, "A Modified Multiple Shooting Algorithm for Parameter Estimation in ODEs Using Adjoint Sensitivity Analysis," *Applied Mathematics and Computation*, vol. 390, p. 125644, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0096300320305981>
- [57] H. G. Bock, "Recent Advances in Parameteridentification Techniques for O.D.E." in *Numerical Treatment of Inverse Problems in Differential and Integral Equations: Proceedings of an International Workshop, Heidelberg, Fed. Rep. of Germany, August 30—September 3, 1982*, ser. Progress in Scientific Computing, P. Deufhard and E. Hairer, Eds. Boston, MA: Birkhäuser, 1983, pp. 95–121. [Online]. Available: https://doi.org/10.1007/978-1-4684-7324-7_7
- [58] M. Peifer and J. Timmer, "Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting," *IET Systems Biology*, vol. 1, no. 2, pp. 78–88, Mar. 2007. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-syb_20060067
- [59] J. C. Platt and A. H. Barr, "Constrained differential optimization for neural networks," 1988.
- [60] Q. Wang, S. R. Kulkarni, and S. Verdu, "Divergence Estimation for Multidimensional Densities Via k -Nearest-Neighbor Distances," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2392–2405, May 2009. [Online]. Available: <http://ieeexplore.ieee.org/document/4839047>
- [61] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, p. 723–773, Mar. 2012.
- [62] R. Y. Rubinstein, "Optimization of computer simulation models with rare events," *European Journal of Operational Research*, vol. 99, no. 1, pp. 89–112, 1997.
- [63] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman, "emcee: The MCMC hammer," *Publications of the Astronomical Society of the Pacific*, vol. 125, no. 925, p. 306–312, Mar 2013. [Online]. Available: <http://dx.doi.org/10.1086/670067>
- [64] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient Langevin dynamics," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ser. ICML'11. Madison, WI, USA: Omnipress, Jun. 2011, pp. 681–688.
- [65] M. D. Hoffman and A. Gelman, "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo," *Journal of Machine Learning Research*, vol. 15, no. 47, pp. 1593–1623, 2014. [Online]. Available: <http://jmlr.org/papers/v15/hoffman14a.html>
- [66] J. Goodman and J. Weare, "Ensemble samplers with affine invariance," *Communications in Applied Mathematics and Computational Science*, vol. 5, no. 1, pp. 65–80, Jan. 2010, publisher: Mathematical Sciences Publishers. [Online]. Available: <https://msp.org/camcos/2010/5-1/p04.xhtml>
- [67] A. Asseman, T. Kornuta, and A. Ozcan, "Learning beyond simulated physics," in *Modeling and Decision-making in the Spatiotemporal Domain Workshop*, 2018. [Online]. Available: <https://openreview.net/forum?id=HylajWsRF7>
- [68] A. Lambert, A. Fishman, D. Fox, B. Boots, and F. Ramos, "Stein Variational Model Predictive Control," *Conference on Robot Learning (CoRL)*, 2020. [Online]. Available: https://corlconf.github.io/corl2020/paper_282/