Efficient emulation of relativistic heavy ion collisions with transfer learning

D. Liyanage,[1] Y. Ji,[2] D. Everett,[1] M. Heffernan,[3] U. Heinz,[1] S. Mak,[2] and J.-F. Paquet[4]

[1] *Department of Physics, The Ohio State University, Columbus OH 43210.*
[2] *Department of Statistical Science, Duke University, Durham NC 27708.*
[3] *Department of Physics, McGill University, Montréal QC H3A 2T8, Canada.*
[4] *Department of Physics, Duke University, Durham NC 27708.*

Measurements from the Large Hadron Collider (LHC) and the Relativistic Heavy Ion Collider (RHIC) can be used to study the properties of quark-gluon plasma. Systematic constraints on these properties must combine measurements from different collision systems and methodically account for experimental and theoretical uncertainties. Such studies require a vast number of costly numerical simulations. While computationally inexpensive surrogate models ("emulators") can be used to efficiently approximate the predictions of heavy ion simulations across a broad range of model parameters, training a reliable emulator remains a computationally expensive task. We use transfer learning to map the parameter dependencies of one model emulator onto another, leveraging similarities between different simulations of heavy ion collisions. By limiting the need for large numbers of simulations to only one of the emulators, this technique reduces the numerical cost of comprehensive uncertainty quantification when studying multiple collision systems and exploring different models.

I. INTRODUCTION

The RHIC and LHC collider facilities create nuclear matter under extreme conditions by colliding heavy nuclei at relativistic velocities. These high energy collisions melt the nuclei and create a strongly interacting, exotic phase of nuclear matter called quark-gluon plasma (QGP) [1]. The QGP filled the universe microseconds after the Big Bang, before it cooled down to produce atomic hydrogen, helium and other light atomic nuclei that we observe in the universe today [2]. Due to its extremely short lifetime ($\sim 10^{-23}$ s) and size ($\sim 10^{-14}$ m), the QGP created in relativistic heavy ion collisions cannot be observed directly; it can only be studied through the final particles it emits.

Modeling of relativistic nuclear collisions is a challenge that involves a succession of phases of many-body nuclear physics with different degrees of freedom; the QGP is only one of them. Realistic numerical simulations of such collisions have many physical parameters that are related to the properties of this QGP. To constrain these properties, one must effectively solve the inverse problem, i.e. find the model parameters, including their uncertainties, for which simulated observables agree well with the experimental data.

Relativistic heavy ion collision experiments have accumulated a vast body of measurements and are continuing to do so. These experimental data vary widely in the size of their uncertainties, which can also have non-trivial correlations. Theoretical simulations add additional uncertainties to the error budget, of two different types: statistical (aleatoric) uncertainties from measuring a finite number of samples from a stochastic process, and systematic (epistemic) uncertainties arising from imperfect modeling of the (not yet fully understood or only approximately implemented) physics underlying the dynamical evolution process. These experimental and theoretical uncertainties limit the precision with which the desired model parameters can be inferred.

Bayesian inference or Bayesian parameter estimation is a modern statistical method that provides a way to reliably infer the properties of QGP, by accounting methodically for both theoretical and experimental uncertainties. Tremendous progress has been made in the study of relativistic heavy ion collisions over the past decade by providing increasingly reliable constraints and error estimates for the properties of QGP using Bayesian statistical techniques [3–14]. As both the model and data have uncertainties, comparing them results in a probability distribution for the model parameters, specifying the probability for a model with a given set of parameters to provide predictions that agree with the experimental observations. A single model with n parameters will have an n-dimensional probability distribution, called in brief "the posterior", describing its agreement with a set of measurements. For a class of competing models, the dimensionality of model parameter space increases accordingly. Bayesian uncertainty quantification depends on the ability to accurately sample this posterior probability distribution, which is generally not known analytically [15]. Markov Chain Monte Carlo (MCMC) techniques provide such sampling methods [16]. They are practical only if fast approximations of otherwise expensive computer simulations are available. Emulation with surrogate models has thus become an essential component in any Bayesian inference involving a computationally expensive likelihood function.

Emulators are machine learning models that provide a computationally efficient prediction of the simulator over the parameter space when trained on a sparse set of full simulation data. While a modeler can choose from a wide range of learning models (e.g., linear regression, decision trees, neural networks) as surrogates for expensive simulations, the standard practice in relativistic nuclear physics [4–14] has been to use Gaussian Process (GP) emulators [17]. There are two reasons for this: (i) GPs provide a flexible non-parametric framework for emulation modeling and (ii) they also provide an efficient quantifica-

tion of the predictive uncertainty associated with the interpolation between training points in the n-dimensional parameter space. In Bayesian parameter estimation, the latter integrates seamlessly with the aleatoric and epistemic uncertainties to yield an accurate quantification of the total uncertainty for the inferred model parameters.

Relativistic heavy ion collision experiments have been conducted at various experimental facilities around the world, using different collision systems (ranging from p+p and p+A to U+U) and different collision energies (ranging from $\sqrt{s_{\rm NN}} = 3\,\text{GeV}$ to $13\,\text{TeV}$).[1] When studying these different systems with Bayesian parameter inference methods, one typically builds separate emulators for each individual system. Each collision is simulated using a multistage model [14, 20–29] that describes the successive dynamical evolution stages. For each stage there typically exist multiple physics models ("modules") based on different physics assumptions. Mixing-and-matching these modules leads to a plethora of theoretical models that, in principle, could all be used to simulate the collision. As recently shown using Bayesian Model Averaging [11], this ambiguity in the theoretical framework can add a significant model uncertainty in the parameter inference. But accounting for it systematically requires studying multiple models, and this generates a need for efficient emulators describing the predictions from different but typically closely related evolution models. If each model emulator needs the same number of training data, the computational cost for building the emulators scales linearly with the number of models. This quickly renders a global Bayesian parameter inference, which includes a representative set of simulation models to describe large sets of experimental data from a variety of collision systems, computationally infeasible.

We introduce here a novel emulation method that significantly reduces the computational barrier for a global Bayesian parameter estimation by requiring a smaller volume of training data for building accurate emulators. This is accomplished by realizing that physical observables from different collision systems are related to each other by common trends resulting from the uniqueness of the underlying physics, and that predictions for these observables from models based on different sets of approximations for this underlying physics also share common trends reflecting this common ancestry. We use "transfer learning" [30–32] to transfer knowledge about such trends from emulators for a specific model trained on a larger, much more expensive set of already existing training data generated for a previously analyzed system, to new emulators for a different simulation model of the same collision system or for simulations of a different collision system. We provide illustrative examples on the use of this new technique; the code[2] generating these examples, including full documentation, can be found at [34].

This work is organized as follows. Sec. II provides an introduction to transfer learning and Gaussian Process emulation. Applications of transfer learning techniques for emulation of relativistic heavy ion collisions are introduced and illustrated in Sec. III. In Sec. IV we illustrate a new way of performing sensitivity analysis offered by transfer learning. We then compare the accuracy of and computational savings from the new emulation method to the existing usage of Gaussian Processes in Sec. V. Applications of this method and its limitations in analyzing relativistic heavy ion collisions and beyond are discussed in Sec. VI. We conclude in Sec. VII with an outlook on future work. The Appendix describes the standardization process for experimental observables used in our work.

II. TRANSFER LEARNING AND GAUSSIAN PROCESS EMULATION

A. Transfer learning

Transfer learning methods (see, e.g., [32, 35]) aim to improve learning in a designated task (called the *target* task), by leveraging information from other related tasks (called *source* tasks). This is in contrast to traditional machine learning methods, which instead build separate learning models for each task in isolation. Transfer learning methods are becoming increasingly popular in the machine learning literature, since it allows for efficient learning of target systems where training data can be expensive to obtain [36].

While there are many types of transfer learning models, the one most relevant for the current study is *inductive* transfer learning [32], where the source and target problems have identical input domains but different tasks. In such problems, the training data for the target task is typically scarce, so a model trained solely on such data does not provide good predictive performance. Existing transfer learning techniques tackle this problem by learning and correcting the bias between source and target tasks. One such method is TrAdaBoost [37], which weighs each source data point by a measure of similarity to the target for better classification performance on the target task. This approach is extended for regression tasks in [38]. [39] proposes an importance-weighted approach for reweighing the source data to predict on the target task. The authors of [40] present an adaptive transfer learning model using Gaussian processes, in which a transfer kernel learns to model similarities between target and source tasks. Their model assumes the same kernel for both target and source, with a dissimilarity parameter accounting for the correlation between them. Our proposed model builds on these ideas but

[1] For readers trying to follow this rapidly-evolving field we recommend the series of proceedings for the annual to biannual *Quark Matter* conferences, the latest of which is published in [18, 19].

[2] We use the EMUKIT package [33] to implement transfer learning emulation.

takes instead an additive approach where we introduce a discrepancy function between source and target, modeled by a GP. This provides a more flexible way of transferring information and also makes it possible to analyze the differences between source and target via sensitivity analysis on the discrepancy function. A comprehensive survey on existing transfer learning techniques can be found in [41].

The proposed transfer learning emulator is based on the popular Kennedy-O'Hagan (KO) model for multi-fidelity emulation [31]. Here we address the bias between target and source by applying a correlation factor and a discrepancy function. This work provides a novel application of the KO model for modeling heavy ion collisions between different nuclear species, or for the same species using different but related dynamical evolution codes.

B. Gaussian process emulation

Gaussian processes (GPs) [42] are a popular choice for emulation of computer simulations [43] and have been exploited in diverse applications from rocket design [44] to 3D printing [45]. GPs are an essential tool for Bayesian parameter estimation of complex simulation models, where they are used to efficiently interpolate between full model runs taken on a sparse set of design points in a high-dimensional parameter space, largely due to their ability to efficiently provide a probabilistic quantification of the incurred interpolation uncertainty.

Let $f(\mathbf{x})$ denote the simulation output at parameter point $\mathbf{x} = (x_1, \cdots, x_q) \in \mathcal{X}$, where $\mathcal{X}$ is the parameter space. A Gaussian process is a stochastic process $\{f(\mathbf{x}) \in \mathbb{R} : \mathbf{x} \in \mathcal{X}\}$, for which any finite collection of points $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)$ have a joint Gaussian distribution. A GP is fully characterized by a mean function $\mu(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and a covariance function $k(\mathbf{x}, \mathbf{x}') = \mathrm{Cov}[f(\mathbf{x}), f(\mathbf{x}')]$. This will be denoted as

$$f(\cdot) \sim \mathrm{GP}\{\mu(\cdot), k(\cdot, \cdot)\}.$$

The mean function $\mu(\mathbf{x})$ denotes the mean of the process while the covariance function controls the smoothness of its sample paths.

From a Bayesian perspective, the GP model $f(\cdot)$ prior to conditioning on data from the full model runs represents a modeler's prior belief on the simulation output before observing it. In practice, the mean function $\mu(\cdot)$ prior to conditioning is typically set to be a constant μ. There are several popular choices for the covariance function $k(\cdot, \cdot)$, including Gaussian[3], Matérn, and cubic covariances [42]. In this study, we employ the anisotropic Gaussian covariance function, widely used for computer experiment emulators [17]:

$$k^{\mathrm{SE}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left[-\sum_{j=1}^{q} \frac{(x_j - x_j')^2}{2l_j^2}\right]. \tag{1}$$

Here $\sigma^2 > 0$ is a variance parameter controlling the variation of the process around its mean, while the parameters $l_j > 0$ $(j = 1, 2, \cdots, q)$ are characteristic length-scales. Larger l_j induce stronger correlations between nearby points, resulting in smoother sample paths, whereas smaller l_j result in more wiggly sample paths.

We now integrate the data obtained from the full model simulations. Suppose noisy outputs $\mathbf{y} = (y_1, \ldots, y_n)$ are simulated at parameters $\mathbf{x}_1, \ldots, \mathbf{x}_n$ via the sampling model

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \overset{i.i.d.}{\sim} N(0, \gamma^2), \tag{2}$$

where ϵ_i represents statistical uncertainty, *i.i.d.* stands for "independent and identically distributed", and $N(0, \gamma^2)$ denotes a Gaussian normal distribution with zero mean and variance γ^2. Conditioning on the data $\mathbf{y}$ (and assuming fixed parameters μ, σ^2 and l), the posterior distribution of f at a new point on the parameter space $\mathbf{x}_{\mathrm{new}}$ can be shown to be [17]

$$[f(\mathbf{x}_{\mathrm{new}})|\mathbf{y}] \sim N(\mu^*(\mathbf{x}_{\mathrm{new}}), {\sigma^2}^*(\mathbf{x}_{\mathrm{new}})), \tag{3}$$

where the posterior mean and variance are given by

$$\begin{aligned} \mu^*(\mathbf{x}_{\mathrm{new}}) &= \mu + \mathbf{k}_{\mathrm{new}}^\top (\mathbf{K} + \gamma^2 \mathbf{I}_n)^{-1} (\boldsymbol{y} - \mu \mathbf{1}_n) \\ {\sigma^2}^*(\mathbf{x}_{\mathrm{new}}) &= k(\mathbf{x}_{\mathrm{new}}, \mathbf{x}_{\mathrm{new}}) - \mathbf{k}_{\mathrm{new}}^\top (\mathbf{K} + \gamma^2 \mathbf{I}_n)^{-1} \mathbf{k}_{\mathrm{new}}. \end{aligned} \tag{4}$$

Here, $\mathbf{k}_{\mathrm{new}} = [k(\mathbf{x}_{\mathrm{new}}, \mathbf{x}_i)]_{i=1}^n$ is the covariance vector between the n existing design points of full-model runs and a new, interpolated point in the parameter space, and $\mathbf{K} = [k(\mathbf{x}_i, \mathbf{x}_j)_{i,j=1}^n]$ is the covariance matrix for the simulated data. Equations (3,4) provide the basis for emulator modeling: the posterior mean $\mu^*(\mathbf{x}_{\mathrm{new}})$ serves as the emulator model prediction at a new point $\mathbf{x}_{\mathrm{new}}$, and the posterior variance ${\sigma^2}^*(\mathbf{x}_{\mathrm{new}})$ yields a quantification of emulator model uncertainty. A key appeal of GP emulators is that both their prediction and uncertainty can be efficiently computed via such closed-form expressions. In practice, the parameters μ, σ^2 and l are first estimated using the maximum likelihood method [46], then plugged into the predictive equations (4) for emulation (see [17] for further details on plug-in predictors).

C. Emulator model specification

We now extend the above GP modeling framework to build a transfer learning emulator model. Let $f_T(\mathbf{x})$ denote the simulator output at parameter $\mathbf{x}$ for the *target* system, *i.e.*, the system for which data[4] are limited and

[3] In the statistical literature the Gaussian function is often called a "squared-exponential", indicated here by the superscript SE.

[4] Here and in the following "data" is short for "full-model simulation predictions".

emulation is desired. Let $f_S(\mathbf{x})$ denote the simulator output at parameter $\mathbf{x}$ for the *source* system, *i.e.*, the system for which a large set of simulation data is available. We assume that the source and target systems share the same parameter space.

We adopt the following transfer learning model linking the source and target systems:

$$f_T(\mathbf{x}) = \rho f_S(\mathbf{x}) + \delta(\mathbf{x}). \tag{5}$$

Here, ρ is a linear correlation coefficient linking the source system to the target and will be estimated from data using maximum likelihood methods. The function $\delta(\mathbf{x})$ models the *discrepancy* (*i.e.* systematic differences) between source and target after accounting for correlations. Since neither $f_S(\mathbf{x})$ nor $\delta(\mathbf{x})$ are known with certainty, we then place independent priors on both terms:

$$f_S(\mathbf{x}) \sim \mathrm{GP}\{\mu_S, k_S^{\mathrm{SE}}(\cdot,\cdot)\}, \quad \delta(\mathbf{x}) \sim \mathrm{GP}\{\mu_\delta, k_\delta^{\mathrm{SE}}(\cdot,\cdot)\}, \tag{6}$$

where different variance and length-scale parameters are used for the squared-exponential kernels k_S^{SE} and k_δ^{SE}. As before, the GP mean parameters μ_S and μ_δ, variances σ_S^2 and σ_δ^2, and length-scales l_S and l_δ are estimated from data using maximum likelihood methods.

Consider now the simulation data for training: for the source system, suppose noisy outputs $\mathbf{y}_S = \left(y_1^S, \ldots, y_m^S\right)$ are available at parameters $\mathbf{X}_S = (\mathbf{x}_1^S, \ldots, \mathbf{x}_m^S)$ via the sampling model

$$y_i^S = f_S(\mathbf{x}_i^S) + \epsilon_i^S, \quad \epsilon_i^S \overset{i.i.d.}{\sim} N(0, \gamma_S^2), \quad i = 1, \ldots, m. \tag{7}$$

For the target system, suppose also that noisy outputs $\mathbf{y}_T = \left(y_1^T, \ldots, y_n^T\right)$ are simulated at parameters $\mathbf{X}_T = (\mathbf{x}_1^T, \ldots, \mathbf{x}_n^T)$ via

$$y_j^T = f_T(\mathbf{x}_j^T) + \epsilon_j^T, \quad \epsilon_j^T \overset{i.i.d.}{\sim} N(0, \gamma_T^2), \quad j = 1, \ldots, n. \tag{8}$$

The goal is to to realize computational savings by keeping the sample size n for the target system much smaller than the sample size m for the source system.

Conditioning on both sets of data $\mathbf{y}_S$ and $\mathbf{y}_T$ (and assuming fixed GP model parameters), the posterior distribution for the target system f_T at a new parameter $\mathbf{x}_{\mathrm{new}}$ can be shown to be

$$[f_T(\mathbf{x}_{\mathrm{new}}) | \mathbf{y}_S, \mathbf{y}_T] \sim N(\mu_T^*(\mathbf{x}_{\mathrm{new}}), {\sigma_T^2}^*(\mathbf{x}_{\mathrm{new}})), \tag{9}$$

where the posterior mean and variance of the transfer learning emulator model are given by

$$\begin{aligned} \mu_T^*(\mathbf{x}_{\mathrm{new}}) &= \rho\mu_S + \mu_\delta \\ &\quad + \mathbf{k}_{\mathrm{new}}^\top \boldsymbol{\Sigma}^{-1} \left(\begin{bmatrix} \mathbf{y}_S \\ \mathbf{y}_T \end{bmatrix} - \begin{bmatrix} \mu_S \mathbf{1}_m \\ (\rho\mu_S + \mu_\delta)\mathbf{1}_n \end{bmatrix} \right), \\ {\sigma_T^2}^*(\mathbf{x}_{\mathrm{new}}) &= \rho^2 \mathbf{k}_S(\mathbf{x}_{\mathrm{new}}, \mathbf{x}_{\mathrm{new}}) + \mathbf{k}_\delta(\mathbf{x}_{\mathrm{new}}, \mathbf{x}_{\mathrm{new}}) \\ &\quad - \mathbf{k}_{\mathrm{new}}^\top \boldsymbol{\Sigma}^{-1} \mathbf{k}_{\mathrm{new}}, \end{aligned} \tag{10}$$

with $\mathbf{k}_{\mathrm{new}} = [\mathbf{k}_{\mathrm{new}}^S, \mathbf{k}_{\mathrm{new}}^T]$ and $\mathbf{k}_{\mathrm{new}}^S = [k(\mathbf{x}_{\mathrm{new}}, \mathbf{x}_i)]_{i=1}^m$, $\mathbf{k}_{\mathrm{new}}^T = [k(\mathbf{x}_{\mathrm{new}}, \mathbf{x}_j)]_{j=1}^n$, and

$$\boldsymbol{\Sigma} = \begin{bmatrix} \mathbf{K}_S(\mathbf{X}_S) + \gamma_S^2 \mathbf{I}_m & \rho \mathbf{K}_S(\mathbf{X}_S, \mathbf{X}_T) \\ \rho \mathbf{K}_S(\mathbf{X}_S, \mathbf{X}_T)^T & \rho^2 \mathbf{K}_S(\mathbf{X}_T) + \mathbf{K}_\delta(\mathbf{X}_T) + \gamma_T^2 \mathbf{I}_n \end{bmatrix}.$$

Equation (10) provides the predictive equations for our transfer learning emulator model: $\mu_T^*(\mathbf{x}_{\mathrm{new}})$ serves as the emulator model prediction while ${\sigma_T^2}^*(\mathbf{x}_{\mathrm{new}})$ quantifies its uncertainty. These closed-form equations enable efficient probabilistic predictions from the proposed model. As before, the parameters μ, σ^2, l and ρ are estimated using maximum likelihood [46] (first for the source, then for the discrepancy), then used in the predictive equations (10) for emulation of the target system.

The discrepancy function $\delta(\mathbf{x})$, which captures the systematic differences between the source and target, can then be estimated from equation (5) as:

$$\hat{\delta}(\mathbf{x}) = \mu_T^*(\mathbf{x}) - \rho\mu_S^*(\mathbf{x}), \tag{11}$$

where $\mu_T^*(\mathbf{x})$ is the posterior mean in equation (10) and $\mu_S^*(\mathbf{x})$ is the posterior mean of Gaussian process emulator in equation (4). A careful analysis of the estimated discrepancy function $\hat{\delta}(\mathbf{x})$ can yield useful insights on the different physics between the source and target systems. We explore this further in Section IV.

The above transfer learning emulator model is closely related to the KO model which is widely used for multi-fidelity emulation. The KO model aims to emulate a high-fidelity computer simulation, using data simulated from lower-fidelity approximations of the *same* system. The KO model is similar in spirit to Equation (5) in that the high-fidelity code is modeled as a linear autoregressive formulation of the low-fidelity code, plus a discrepancy term to account for systematic bias. The key difference for the proposed model is that instead of transferring learning from simulations of different fidelities for the *same* system, our emulator model is trained by transferring knowledge between high-fidelity simulations of *different* systems that have common traits.

III. TRANSFER LEARNING EMULATORS FOR RELATIVISTIC HEAVY ION COLLISIONS

All large scale Bayesian parameter estimations for relativistic heavy ion collisions have been made computationally feasible by using GPs as surrogates for computationally expensive simulations. The biggest computational cost associated with any such analysis is in generating training data for the GPs. In this section, we compare the accuracy and the computational cost associated with two distinct emulation methods: direct emulation with traditional Gaussian Processes and our novel transfer learning emulation method. We show that transfer learning requires significantly fewer training data from the computationally expensive simulation and thus lowers the computational barrier associated with Bayesian parameter estimation for complex problems, such as the one posed by the dynamical modeling of relativistic heavy ion collisions. Transfer learning is a particularly powerful tool for situations where (i) the training data on the target alone are insufficient to fit a good emulator, and

Observable Type	Centralities	
	Au+Au at 0.2 TeV	Pb+Pb at 2.76 TeV
Charged particle multiplicity; $dN_{ch}/d\eta$	None	[0-5], [60-70]
Pion multiplicity; dN_{π}/dy	[0-5], [40-50]	[0-5], [60-70]
Mean transverse momenta of pions; $\langle p_T \rangle_{\pi}$	[0-5], [40-50]	[0-5], [60-70]
Two-particle elliptic flow; $v_2\{2\}$	[0-5], [40-50]	[0-5], [60-70]
Fluctuation in the mean transverse momentum; $\delta p_T/p_T$	None	[0-5], [55-60]

TABLE I. Observables used for emulation

(ii) the amount of training data available on the source is much larger than that for the target.

A. Multistage model of relativistic heavy ion collision simulations

The relativistic heavy ion collision model used in the present work [14] involves the following modules describing different evolution stages:

1. T_RENTo: A phenomenological model of the initial energy deposition after the impact of the nuclei [47, 48].

2. Freestreaming: A model for weakly-coupled pre-equilibrium dynamics, covering the first fm/c or so [49–51].

3. Relativistic viscous hydrodynamics, describing the dissipative evolution of near-equilibrium QCD matter with the code MUSIC [52–56].

4. Particlization: Conversion of the fluid into particles after it cools down below the critical temperature where QGP converts back into hadrons, described by the Cooper-Frye formula [57, 58]. To parameterize the local hadron phase space distributions using only the ten components of the energy momentum tensor evolved by the hydrodynamic model, three different models with different physics assumptions are explored:

   (a) Grad viscous corrections, which expand the distribution function up to second order in hadron momenta [59];

   (b) Chapman-Enskog (CE) viscous corrections, which solve the Relaxation-Time-Approximation Boltzmann equation for linearized corrections to the distribution function [60]; and

   (c) Pratt-Torrieri-Bernhard (PTB) modified equilibrium viscous corrections [61] which uses an exponential ansatz ensuring a positive definite distribution function.

   These corrections are implemented using the iS3D sampler [62, 63].

5. Hadronic decays and re-scatterings are modeled with Boltzmann kinetic transport using the code SMASH [64, 65].

To apply and test transfer learning techniques in this setting, we use a very large set of full-model simulation data that were generated for calibrating the JETSCAPE modeling framework [14], including the following systems:

1. Pb+Pb collisions at $\sqrt{s_{\rm NN}} = 2.76\,\text{TeV}$ with

   (a) Grad viscous corrections,

   (b) Chapman-Enskog viscous corrections, and

   (c) Pratt-Torrieri-Bernhard viscous corrections;

2. Au+Au collisions at $\sqrt{s_{\rm NN}} = 0.2\,\text{TeV}$ center of mass energy with Grad viscous corrections.

All these simulations share the same set of 17 model parameters described in [11, 14]. For model calibration, full-model simulations were performed at 500 design points that uniformly cover the 17-dimensional parameter space within a finite 17-dimensional cube described in [11, 14], using maximin Latin Hypercube sampling [66].[5] For each design point and each particlization model, 2500 simulations were performed with stochastically fluctuating initial conditions and particlization results. For each design point and particlization model, a multitude of experimental observables were computed and compared with the corresponding experimental data. We use full-model predictions for only a subset of these observables (listed in Table I) to illustrate the proposed transfer learning emulator. For simplicity, we focus here on only two collision centralities, "central" ([0%-5%] centrality) and "peripheral" ([40%-50%] centrality for the Au+Au collisions at RHIC, and [55%-60%] or [60%-70%] (whichever was the most peripheral bin available) for the Pb+Pb collisions at the LHC), and also leave out the yields and mean transverse momenta of kaons and protons, charged hadron triangular flow and transverse energy (E_T) distributions.

For each choice of collision system and particlization model, we thus have a set of 473 samples of the parameter space (design points) that provide mean values and

[5] For technical reasons, only the simulation results from 473 of these 500 design points were used in the present analysis.

errors for each observable to train its emulator. To test the performance of the trained emulator we also generated additional test data sets for each model: 100 design points from a separate maximin Latin Hypercube design. Note that the emulators are not trained directly on the observables (as predicted by the simulations) listed in Table I; we first perform a standardization of each of the observables using the means and variances of the source simulation data. These transformations are slightly different from those used in [14] – see Appendix A for details.

The test data set for each model is used to evaluate the performance of each emulator by calculating the mean squared error (MSE):

$$\mathrm{MSE} = \sum_{\substack{i\in\{\text{test design}\}\\ l\in\{\text{observables}\}}} \frac{\left[\hat{Y}^l_{\mathrm{sim}}(\mathbf{x_i}) - \hat{Y}^l_{\mathrm{emu}}(\mathbf{x_i})\right]^2}{N_{\mathrm{test}} N_{\mathrm{obs}}}, \qquad (12)$$

where $\mathbf{x_i}$ are the model parameters for the i^{th} test design point and $\hat{Y}^l_{\mathrm{sim}}, \hat{Y}^l_{\mathrm{emu}}$ represent standardized (See Appendix A) simulation and emulation outputs for the l^{th} observable. We will show plots of the MSE for target emulators constructed with n target training points ($1 \le n \le 473$), using either the standard GP training protocol or the transfer learning protocol, and compare their performance as a function of n. As discussed in Sec. II C, the transfer learning emulator is trained by using these n sets of target data on top of a source emulator that has been previously trained with a larger number m of design points from the source system (here $m = 473$).

B. Transfer learning between different collision systems

As our first application of transfer learning methods, we build emulators for simulated Au+Au collisions at $\sqrt{s_{\mathrm{NN}}} = 0.2\,\mathrm{TeV}$ as the target system, using available trained emulators for Pb+Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 2.76\,\mathrm{TeV}$ as our source. The two emulation methods discussed previously are trained for each of the six observables shown in the Au+Au column of Table I, as a function of the number of design points n for which full-model simulations of the target system are available. We do this by first randomly dividing the total set of $n_{\mathrm{max}} = 473$ simulation data for the target from previous work [11, 14] into 10 roughly equal size sets (nine batches of 47 plus one batch of 50 design points). We then train the emulators using only one batch of target design points, and then repeat the training procedure after successively adding the remaining batches. After each training step, we compare the predictions for the observables from the trained emulators with the full-model test data for the 100 parameter sets in the test design, and compute its mean squared error (MSE, Eq. (12)). The result is shown in Fig. 1 as a function of the number n of target designs used for training.

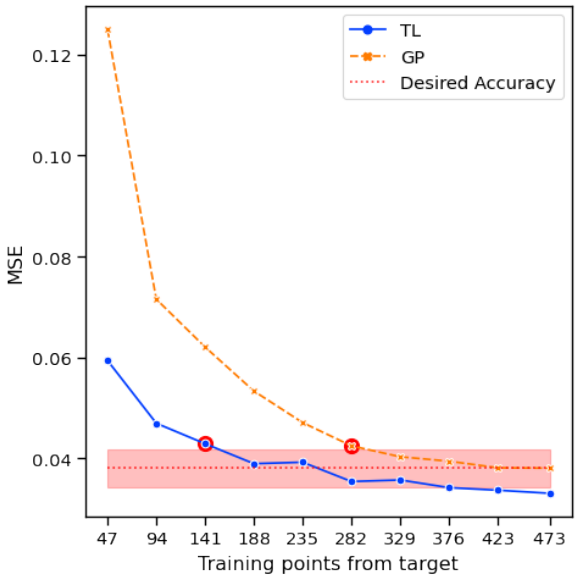

FIG. 1. Mean squared error prediction accuracy of emulators for Au+Au collisions at $\sqrt{s_{\mathrm{NN}}} = 200\,\mathrm{GeV}$ using the Grad particlization model. The transfer learning emulator uses a source emulator trained on model simulations for Pb+Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 2700\,\mathrm{GeV}$. The MSE shown is averaged over all observables, but the curves for the MSE of individual observables look all very similar. See text for discussion.

The dashed orange line in the figure (labeled GP) shows the MSE for the GP emulator of the target system using the standard training protocol, without any help from the source system emulator. The dotted red horizontal line shows the final MSE reached by this method using all 473 available target design points from the full-model simulation data, with the shaded band representing a 10% variation around this value. The solid blue line (labeled TL) shows the MSE for the proposed transfer learning emulation method, which, in addition to the n target design points, also makes use of the information from the previously trained, costly GP emulator for the source system. The two red dots indicate the smallest number n of target design points needed, for each emulator, for its MSE to come within 10% of the "asymptotic precision" (defined by the MSE at the maximally available number of target training points) shown by the dotted red line.

The solid blue curve denoting the transfer learning MSE clearly shows that the TL emulator is more accurate than the traditional GP emulator (dashed orange curve), for all values n of the number of target design points used. The relative advantage of the transfer learning emulator is particularly evident for small numbers of target system design points. For example, when using only 47 design points for Au+Au, the transfer learning emulator has approximately half the mean squared error of the traditional emulator. Note that, even in the "asymptotic limit" when all 473 target design points are

used, the proposed transfer learning emulator still yields improved precision over the standard GP emulator, by leveraging information from the source system emulator.

As expected, for both emulator models, the emulation prediction error (in terms of MSE) decreases monotonically with increasing number of target training points n. For the proposed transfer learning emulator, the rate of decrease is not always uniform, which suggests that there is a diminishing marginal decrease in MSE for each additional target design point. In other words, at a certain point, the "new" information provided by the target training data is minor compared with the "old" information already contributed by the source system emulator.

Another way to quantify the success of the proposed transfer learning emulator is via the two large red dots in Fig. 1, where it can be seen that the same Au+Au collision simulation can be emulated with the same accuracy at half the number of full-model simulations. This level of success of transfer learning is quite encouraging, considering that the target here (Au+Au at $\sqrt{s_{\rm NN}} = 200\,\text{GeV}$) involves collisions at more than an order of magnitude lower center of mass energy than the source system (Pb+Pb collisions at $\sqrt{s_{\rm NN}} = 2760\,\text{GeV}$).

C. Transfer learning between different viscous corrections at particlization

As discussed in Section III A, the multistage dynamical modelling of heavy ion collisions requires approximations and switching between different physical pictures which is associated with theoretical uncertainty: different modelling choices can be made in each collision stage, based on different assumptions or approximations of the governing physics. Different choices lead to models whose predictions differ from each other in quantitative detail but share qualitative features and common trends under variation of certain experimental control parameters, such as collision energy, collision centrality, system size etc. For each such model variant, teaching these trends to an emulator for its observables requires evaluating the full model at a large number of design points. Transfer learning offers a more computationally efficient strategy: after having spent large numerical resources on the training of sufficiently accurate emulators for the observables predicted for one such model variant (the source), equally accurate emulators for other variants (the targets) can be obtained at a fraction of the cost by transferring some of the qualitative tendencies from source to the targets.

We illustrate this idea here by considering as source and targets model variants obtained by swapping out one particular module in the multistage model, the particlization module (we refer to the discussion in Sec. III A). We consider Pb+Pb collisions at the LHC, simulated with Grad model particlization, as our "source", and the same collisions simulated with Pratt-Torrieri-Bernhard (PTB, Fig. 2) or Chapman-Enskog particlization (CE, Fig. 3) as our "targets".

The data we work with are the simulated model outputs for each of the three particlization models from the same design points discussed in the preceding subsection, a maximum of 473 points for emulator training plus a fixed number of 100 design points for emulator testing. Different from before, a larger set of observables is available for Pb+Pb collisions at the LHC than we had to emulate for Au+Au collisions at RHIC (c.f. Table I). We follow the same training strategy as described in the preceding subsection, for emulators predicting the model outputs for this larger set of observables but using the same design point batches as considered before. To zero in on the relative performance of the TL and GP emulators for the hypothetical case where only very small numbers of target model design points are available, we additionally divided the 473 total target design points to which had access randomly into smaller batches of 5 design points each, allowing studies of the evolution of the emulators' MSE with n for smaller n-values (see inset in Fig. 3).

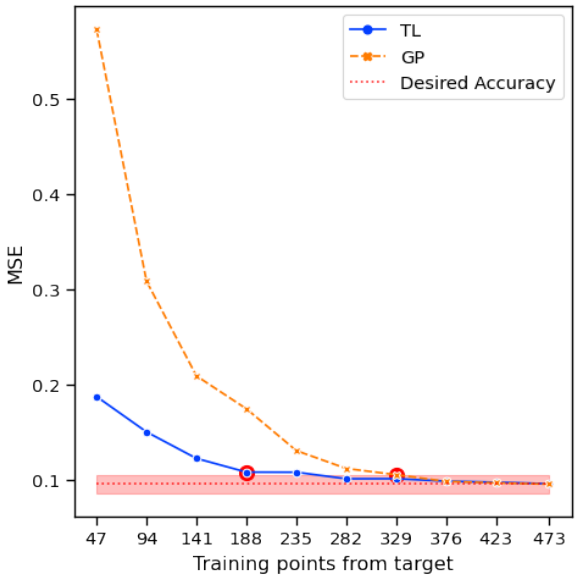

FIG. 2. Mean squared error prediction accuracy of emulators for Pb+Pb collisions at $\sqrt{s_{\rm NN}} = 2760\,\text{GeV}$ using the Pratt-Torrieri-Bernhard particlization model. The transfer learning emulator uses a source emulator trained on model simulations for Pb+Pb collisions at the same $\sqrt{s_{\rm NN}}$ using the Grad particlization model. The MSE shown is averaged over all observables, but the curves for the MSE of individual observables look all very similar. See text for discussion.

In Figs. 2 and 3 we note that the transfer learning emulators again already approach their asymptotic accuracy within 10% for a much smaller number of target design points than those generated with the standard GP training protocol, similar to the preceding subsection. We also note that for the case of different particlization routines shown in Figs. 2 and 3 the accuracy advantage of the TL emulators over their GP siblings begins to disappear once about 60% of the maximally available number of target training points ($n_{\rm mx} = 473$) have been used.

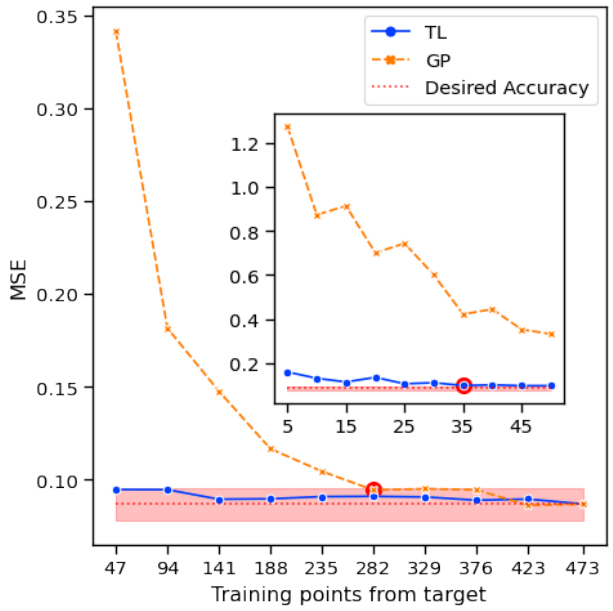

FIG. 3. Same as Fig. 2 but for a target using Chapman-Enskog particlization.

For small numbers of target design points $n \sim 50$, the TL emulators have approximately one third or less of the mean squared error of the traditional GP emulators for the targets involving a change of particlization model, compared to the factor two reduction for the target involving a lower collision energy studied in Sec. III B. Amazingly, the inset in Fig. 3 shows that for the CE particlization model the MSE prediction accuracy of the TL emulator needs only 35 target training points to reach within 10% of its asymptotic value, and is not much worse even for as few as only 5 target training points. This means that the qualitative trends of the observables predicted by the Grad and CE particlization models must be very close (much closer than between the source and the other two target models studied in this work), and teaching these trends to the target emulator via transfer learning almost completely obviates the need for additional information from full-model simulations of the target model. While this is clearly a special situation, it illustrates the huge cost-saving potential of transfer learning if ways can be found to reliably diagnose the convergence of the emulator accuracy towards its asymptotic value.

IV. SENSITIVITY ANALYSIS

There is evident interest in understanding the effect of individual model parameters on specific observables, to gain intuition about what the experimental data might tell us about the underlying physics and medium properties. This relation between parameters and observables is often explored through "sensitivity analysis", though the exact method varies. Examples from the field of heavy ion physics can be found in Refs. [5, 14, 67, 68].

Transfer learning offers an interesting new way of performing sensitivity analysis, by systematically investigating which model parameters contribute to non-trivial differences in parameter dependencies between source and target models. As described in Sec. II, Eq. (11), these differences can be characterized by the correlation coefficient ρ and its corresponding discrepancy function $\delta(\mathbf{x})$. By estimating both ρ and $\hat{\delta}(x)$ from data, we can then perform a sensitivity analysis on the *estimated* discrepancy function $\hat{\delta}(\mathbf{x})$. Below, we perform such an analysis using the proposed transfer learning emulator and the scenarios discussed in the preceding section.

There are two main types of sensitivity analysis methods from the uncertainty quantification literature [69]: local or global ones. Local sensitivity analysis can quantify the model sensitivity for an observable *at a fixed parameter value*, such as the maximum a posteriori (MAP) estimate obtained from parameter inference. On the other hand, global sensitivity analysis provides an averaged quantification of sensitivity for each parameter over the full parameter space. In what follows, we focus on the latter global sensitivity analysis of the estimated discrepancy function $\hat{\delta}(\mathbf{x})$ (11).

We first introduce the first-order Sobol' indices [70], a popular method for analyzing global sensitivity. Sobol' indices [71, 72] quantify the importance of each parameter for a given function $\delta(\mathbf{x})$, by decomposing its contribution to the variance of $\delta(\cdot)$ over the parameter space. The first-order Sobol' index for model parameter x_j is defined as:

$$\frac{\mathrm{Var}_{X_j}(\mathbb{E}_{\boldsymbol{X}_{-j}}(\delta(\boldsymbol{X})|X_j))}{\mathrm{Var}_{\boldsymbol{X}}(\delta(\boldsymbol{X}))}, \qquad j = 1, \ldots, q. \tag{13}$$

Here, X_j is an independent uniform random variable for parameter x_j over its parameter range, and $\boldsymbol{X} = (X_1, \cdots, X_q)$ is its corresponding random vector for all parameters. The term $\mathbb{E}_{\boldsymbol{X}_{-j}}(\delta(\boldsymbol{X})|x_j)$ is called the *main effect* of parameter x_j: given fixed j-th parameter $X_j = x_j$, it *averages* the function $\delta(\cdot)$ uniformly over the remaining parameters $\boldsymbol{X}_{-j} = \boldsymbol{X} \setminus X_j$. This is formally defined as

$$\mathbb{E}_{\boldsymbol{X}_{-j}}(\delta(\boldsymbol{X})|X_j) = \int_{\mathcal{X}_{-j}} \delta(x_1, \ldots, x_q)\, dU(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_q), \tag{14}$$

where $U(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_q)$ is the uniform probability measure over $\mathcal{X}_{-j}$, the parameter space $\mathcal{X}$ omitting the j-th parameter. The first-order Sobol' index (13) thus quantifies the importance of parameter x_j, by taking the ratio of $\mathrm{Var}_{X_j}(\mathbb{E}_{\boldsymbol{X}_{-j}}(\delta(\boldsymbol{X})|X_j))$, the variance accounted for by the *main effects* $\mathbb{E}_{\boldsymbol{X}_{-j}}(\delta(\boldsymbol{X})|X_j)$, over $\mathrm{Var}_{\boldsymbol{X}}(Y)$, the *total* variance of $\delta(\cdot)$ over all parameters. For costly simulations such as for heavy ion collisions, the integral in (14) can be expensive to evaluate. A standard approach [69] (which we adopt) is to replace the expensive

$\delta(\cdot)$ with the estimated discrepancy $\hat{\delta}(\cdot)$ (11) from the emulator model.

One can further modify the Sobol' indices in (13) by grouping together similar model input parameters. The grouped Sobol' indices in [70] accomplish this. The q input parameters $\mathbf{X} = (X_1, \cdots, X_q)$ (assumed again to be uniformly distributed) are first divided into J groups $(\mathbb{X}_1, \cdots, \mathbb{X}_J)$, given by:

$$(X_1, \cdots, X_q) = (\underbrace{X_1, \ldots, X_{k_1}}_{\mathbb{X}_1}, \ldots, \underbrace{X_{k_{J-1}+1}, \ldots, X_q}_{\mathbb{X}_J}).$$

The first-order *grouped* Sobol' indices can then be defined as:

$$S_j = \frac{\mathrm{Var}_{\mathbb{X}_j}(\mathbb{E}_{\mathbb{X}_{-j}}(Y|\mathbb{X}_j))}{\mathrm{Var}_{\mathbf{X}}(Y)}, \quad j = 1, \cdots, J, \tag{15}$$

where $\mathbb{X}_{-j} = \mathbf{X} \setminus \mathbb{X}_j$ consists of all parameters except for those in group j.

In our implementation, all simulation models consider the same $q = 17$ input model parameters. We group these parameters into six groups according to similarities of their functionality in our model. We employ the following parameter grouping:[6]

- N: The normalization parameter in TRENTo
- TRE: All other parameters in the TRENTo initial-state module.
- Free-streaming: Parameters controlling the free-streaming time
- η/s: All model inputs that parameterize the temperature dependence of the specific shear viscosity.
- ζ/s: All model inputs that parameterize the temperature dependence of the specific bulk viscosity.
- T_{sw}: The particlization temperature separating hydrodynamics and hadronic transport.

This grouping provides meaningful insight on the global sensitivity of the discrepancy between the source and target systems. Our grouped sensitivity analysis agrees with previous sensitivity studies, while providing more concise results with clearer implications.

The left column of Fig. 4 shows the global sensitivity of the model for Pb+Pb collisions at 2.76 TeV with Grad viscous corrections, obtained from the source model emulators discussed before. The six panels in that column correspond to three different observables, each at two different centralities. Within each panel, each of the six

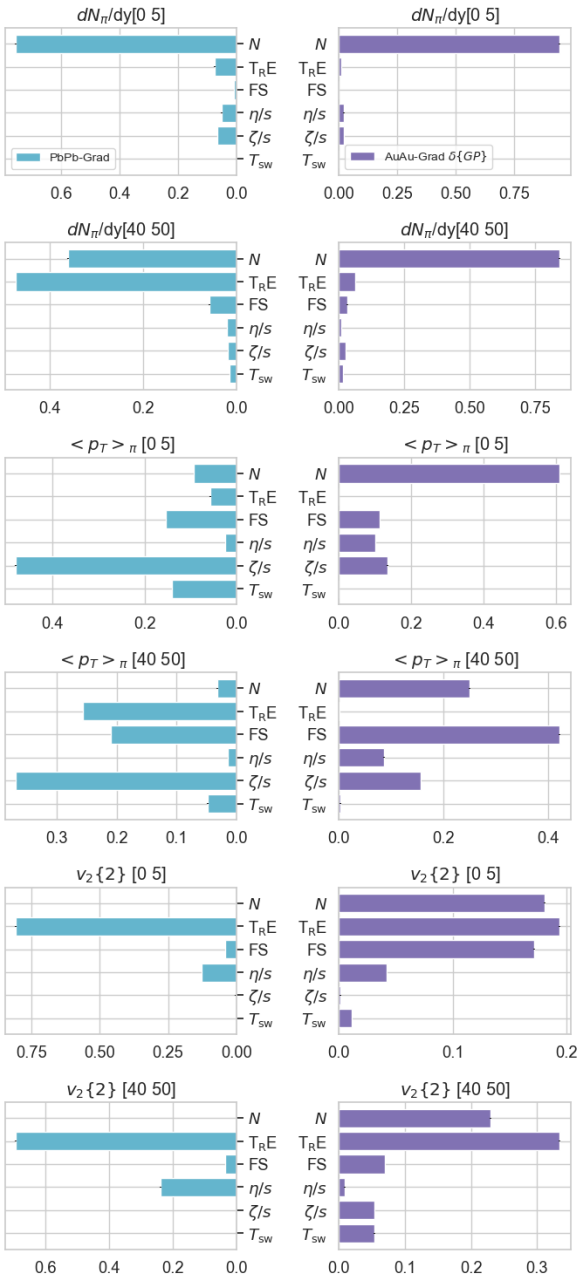

FIG. 4. First order group Sobol' sensitivities of the Pb+Pb 2.76 TeV source simulation (left) and of the discrepancy GP for Au+Au 200 GeV target simulation (right).

bars represents a different group of model parameters. In central collisions (0-5% centrality), the overall pion yield is mostly sensitive to the normalization constant N for the initial energy density profile, the pion mean transverse momentum reacts most strongly to changes in the specific bulk viscosity, and the charged hadron elliptic flow is most sensitive to TRENTo model parameters (in particular, to the granularity of the initial energy density fluctuations). At first it may seem surprising that v_2 reacts more strongly to the TRENTo parameters than to

[6] Note that the results from grouped sensitivity analysis may depend on both the grouping of parameters as well as the choice of model parameterization (*e.g.*, how $\eta/s(T)$ is parameterized), thus one must be careful about the interpretation of such analyses. Further details on this can be found in [73].

the specific shear viscosity but this becomes clearer once one remembers that η/s controls the *hydrodynamic response* to the initial-state source eccentricity ϵ_2, i.e. the ratio v_2/ϵ_2. The large sensitivity of v_2 to the TRENTo parameters really reflects their dominant effect on ϵ_2 which is bigger than that of η/s on the ratio v_2/ϵ_2. In peripheral collisions, on the other hand, the left column of Fig. 4 exhibits additional sensitivities that are much less prominent in central collisions: The overall pion yield now also exhibits sensitivity to the TRENTo parameters; this would be consistent with a stronger viscous heating effects caused by increased granularity in the smaller fireballs generated when the nuclei hit each other at larger impact parameters. The pion mean transverse momentum shows additional sensitivity to the TRENTo parameters and free-streaming time which control the early build-up of radial flow [49]. And the influence of η/s on the charged hadron v_2 grows in relative importance.

In the right column of Fig. 4 we show the sensitivity of the discrepancy GPs between Pb+Pb $\sqrt{s_{\rm NN}}$ =2.76 TeV Grad (source) and Au+Au $\sqrt{s_{\rm NN}}$ =200 GeV (target) model outputs. Clearly, for all three observables, at both collision centralities, the discrepancy GPs share a high sensitivity to the normalization parameter N. This is expected since the most striking difference between these two collision systems is their total multiplicity, driven by the much higher collision energy at the LHC compared to RHIC. We further observe that the discrepancy GPs related to mean transverse momentum ($\langle p_T\rangle_\pi$) and flow observables ($v_2\{2\}$) have a significant sensitivity to the model parameters related to the pre-equilibrium stage, both via the TRENTo initialization model and the duration of the free-streaming stage. This indicates that the pre-equilibrium dynamics depends sensitively on the center of mass energy of the collision. Interestingly, the discrepancy GPs for the mean transverse momentum observable are found to be insensitive to the TRENTo parameters and the switching temperature (which is mostly constrained by the chemical composition of the final hadronic stage [14]). Similarly, the discrepancy GPs for the elliptic flow observables show only weak sensitivity to the specific viscosities.[7] In other words, these observables share roughly the same degree of sensitivity to these parameters at both collision energies – these are the types of systematic trends in the simulations that make transfer learning efficient.

In Fig. 5 we show the analogous sensitivity plots for the discrepancy GPs for the Pb+Pb CE (left column) and Pb+Pb PTB (right column) target models.[8] Com-

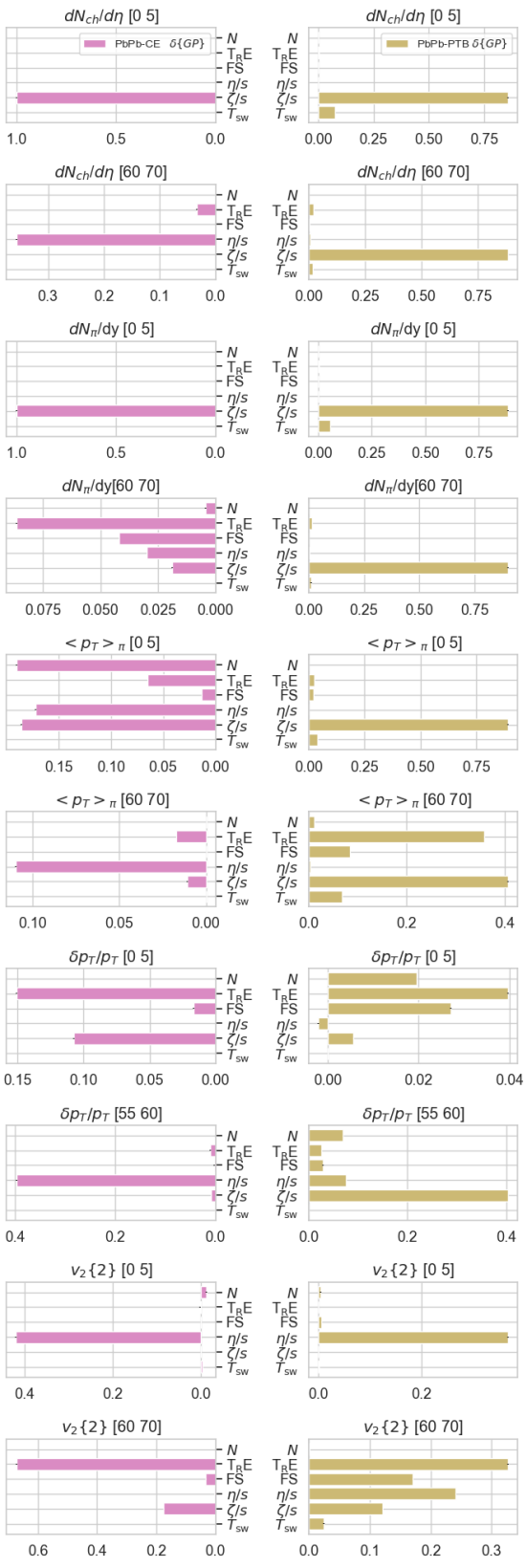

FIG. 5. First order group Sobol' sensitivities of discrepancy GPs. Pb+Pb with CE viscous corrections (left) and Pb+Pb with PTB viscous corrections (right).

[7] The non-vanishing (albeit weak) sensitivity of $v_2\{2\}$ to the parameters describing the temperature dependence of the specific shear viscosity in central collisions and to the specific bulk viscosity in peripheral collisions supports the frequently made assertion that collisions at different center of mass energies should help to constraint the temperature dependence of these viscosities.

[8] The source sensitivities shown in the left column of Fig. 4 are the same for all three targets.

pared to Fig. 4 we include two additional observables (the total charged hadron multiplicity density $dN_{\rm ch}/d\eta$ and the normalized p_T-fluctuations $\delta p_T/\langle p_T\rangle$), again for two collision centralities, resulting in ten panels for each target model. In central collisions, for both targets the majority of the discrepancy GPs (with the exception of the ones emulating the p_T fluctuations and elliptic flow) are found to be most sensitive to the bulk viscosity parameters. Remembering that here the difference between source and targets is how the viscous corrections are handled during particlization, the sensitivity to the viscosity parameterizations is not surprising. More insightful is the observation that the sensitivity to the bulk viscosity sector is mostly stronger than to the shear sector. This may be related to the fact that particlization at $T_{\rm sw}$ happens just after hadronization of the QGP, and that the bulk viscosity peaks near the hadronization phase transition. The situation is, however, more complex in peripheral collisions where the sensitivities to the bulk and shear viscous sectors of parameter space differ between the CE and PTB targets. Furthermore, the mean values and fluctuations of the pion transverse momenta show dominant sensitivities to different sectors of the parameter space than the other observables. All this suggests that Bayesian inference based on the available experimental data should allow us to discriminate between the different particlization models based on their ability to describe the full spectrum of observations, and that combining the strengths and weaknesses of these different models in the future via Bayesian Model Mixing [74, 75] may lead to overall tighter constraints on the fireball properties.

We close this section by noting that relating the source and target model emulators in the form (5) and identifying the corresponding linear correlation coefficient ρ and discrepancy $\hat{\delta}(\mathbf{x})$ may be a very broadly applicable technique for gaining valuable insights into qualitative similarities and differences between different models and into their success and/or failure in describing a given set of experimental data.

V. COMPUTATIONAL SAVINGS FROM TRANSFER LEARNING

Relativistic heavy ion collision experiments produce measurements for hundreds of observables. Since their dynamics is too complex to be described analytically, they are studied theoretically by building phenomenological models that are calibrated with the experimental data. The models have multiple parameters describing properties of the collision dynamics that can not (yet) be computed from first principles and must be inferred using the experimental measurements. After calibration the models can be tested by predicting and measuring additional observables. Since both the experimental data and simulation model outputs have uncertainties associated with them, model calibration (a.k.a. solving "the inverse problem") requires a probabilistic framework.

As already briefly summarized in the Introduction, Bayesian parameter inference is a framework that allows for a systematic probabilistic accounting for our knowledge about the model and its uncertainties. It is based on Bayes theorem,

$$\mathcal{P}(\mathbf{x}|\mathbf{y}_{\rm exp}) = \frac{\mathcal{P}(\mathbf{y}_{\rm exp}|\mathbf{x})\mathcal{P}(\mathbf{x})}{\mathcal{P}(\mathbf{y}_{\rm exp})}. \tag{16}$$

Here $\mathcal{P}(\mathbf{x})$ is prior probability for the parameters $\mathbf{x}$, and $\mathcal{P}(\mathbf{y}_{\rm exp}|\mathbf{x})$ is the likelihood function, describing the probability that model output with a given set of model parameters x agrees with the experimental data $\mathbf{y}_{\rm exp}$. It is usually assumed to be a Gaussian,

$$\mathcal{P}(\mathbf{y}_{\rm exp}|\mathbf{x}) = \frac{1}{\sqrt{|2\pi\mathbf{\Sigma}|}} \exp\Big[-\frac{1}{2}\mathbf{y}^\top\mathbf{\Sigma}^{-1}\mathbf{y}\Big], \tag{17}$$

where $\mathbf{y} \equiv [\mathbf{y}_{\rm sim}(\mathbf{x}) - \mathbf{y}_{\rm exp}]$ is the deviation between model prediction and experimental measurement, and $\mathbf{\Sigma}$ is the total uncertainty, obtained by adding the experimental and simulation uncertainties: $\mathbf{\Sigma} = \mathbf{\Sigma}_{\rm exp} + \mathbf{\Sigma}_{\rm sim}(\mathbf{x})$. For heavy ion collisions $\mathbf{y}$ is a vector that can have more than 100 components, and $\mathbf{\Sigma}$ is a quadratic matrix of the same dimensionality; $|\mathbf{\Sigma}|$ denotes its determinant.

The term $\mathcal{P}(\mathbf{x}|\mathbf{y}_{\rm exp})$ on the left hand side of Eq. (16) is called the posterior (short for "the posterior probability density"). It describes the probability of the model parameters $\mathbf{x}$ given the experimental data $\mathbf{y}_{\rm exp}$, and it is the main quantity of interest in Bayesian parameter inference. Its functional form is generally not known analytically, in particular not for heavy ion collisions. To find the most likely range for the parameters $\mathbf{x}$ and quantify their uncertainty requires numerical techniques for finely sampling the posterior in the neighborhood of the MAP values of the parameters. This is typically achieved by using Markov Chain Monte Carlo (MCMC) techniques.[9]

For each MCMC sample of the posterior (16) the likelihood function (17) must be evaluated; this requires knowledge of the model prediction $\mathbf{y}_{\rm sim}$ at the sampled parameter set $\mathbf{x}$. In a high-dimensional parameter space millions of MCMC samples are needed to explore the posterior in sufficient detail. In principle, this requires running the full-model simulation millions of times. For heavy ion collisions this is practically infeasible, due to the computational cost of each model simulation. This is where numerically cheap surrogate models (emulators) for $\mathbf{y}_{\rm sim}(\mathbf{x})$ come to the rescue. They can be trained by using *very* much smaller numbers of full-model simulations (typically hundreds, not millions). They do introduce an additional emulation (or interpolation) uncertainty which is known and can be simply added to the total simulation uncertainty $\mathbf{\Sigma}_{\rm sim}$ when evaluating the

[9] These techniques require only relative probabilities, so the normalization $\mathcal{P}(\mathbf{y}_{\rm exp})$ in the denominator on the right of Eq. (16) (which is independent of the parameters to be inferred) does not need to be calculated.

Gaussian function (17), but which we want to keep at or below the other uncertainties.

The biggest computational cost is now associated with training the emulators, which requires generating full-model simulation output at the training points. The number of training points needed to build an accurate emulator is therefore of crucial importance. For example, one of the very recent Bayesian inference attempts in relativistic heavy ion collisions [67] which went beyond the work in [14] by emulating additional observables and multiple collision systems, used 64 million CPU hours for emulator training. The authors of [67] considered only a single evolution model which does not provide access to estimating modeling uncertainties as in [11].

The analysis in [11] calibrated each of the different model variants by using the same set of training points, thus multiplying the cost of emulator training by the number of variants. For the extended set of collision systems and higher-statistics observables studied in [67] this would already no longer be practical. The transfer learning technique presented in this work lowers this barrier by reducing the number of training points for subsequent model variants once an accurate emulator has been trained for the first model.

The full-model simulations used in this paper take on average $\mathcal{O}(1000)$ CPU hours for each design point in model parameter space [14]. A majority (80%) of the CPU time is spent on the hadron transport stage after particlization; the remaining CPU time (20%) is mostly utilized by the hydrodynamic QGP evolution code. In figure 6 we show the CPU hours needed to build accurate emulators for the three target model variants discussed in this work, with or without transfer learning from a previously trained source emulator (whose training cost was about 25% higher than the middle orange bar).[10] For this plot, we decided on the required number of training samples for each emulator by requiring convergence of the mean squared error to within 10% of the "asymptotic" accuracy, as shown in Figs. 1-3. We note that the transfer learning method incurs significantly less computational cost compared to the standard GP training protocol. When the source and target models have a much in common (such as the Pb+Pb Grad and Pb+Pb CE models), the computational savings can exceed an order of magnitude (see right bars in Fig. 6).

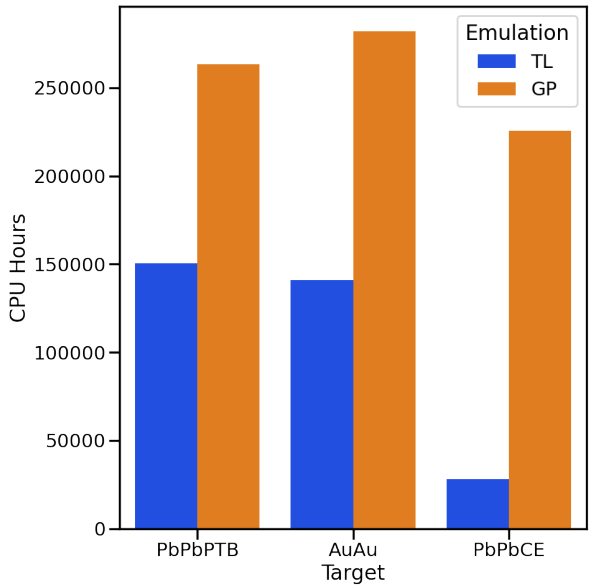

FIG. 6. Comparison between computational resources used by transfer learning (left blue bars) and the traditional GP emulation method (right orange bars).

We note, however, that the cost for the 100 full-model test samples needed to evaluate the MSE and the cost for determining its "asymptotic" value are not accounted for in Fig. 6.[11] The (possibly large) computational cost for additional test runs can be largely avoided by using a cross-validation approach [76], which randomly splits the available target data into training and validation sets multiple times. One then obtains an error estimate by fitting the emulator on the training set and testing on the validation set, cycling through the different splits. Cross-validation error estimates, however, are known to be upwardly biased [76]. This should not be a big issue when using the cross-validation MSEs as a criterion for how many full-model target simulations to use in transfer learning. For the current study, however, we were interested in a precise understanding of the convergence properties of the transfer learning method and therefore elected to use unbiased MSE estimators by running a new set of test samples for validation.

VI. IMPLICATIONS FOR THE STUDY OF HEAVY ION COLLISIONS

Theoretical progress in the phenomenological study of relativistic heavy ion collisions is made by developing increasingly accurate theoretical models of the collisions that can describe both past and future experimental data. Bayesian parameter estimation in relativistic heavy ion physics approaches this aim in two different ways: First, including more experimental data in the analysis, by using multiple collision systems and adding new observables, leads to tighter bounds on the QGP properties. Second, accounting more faithfully for theoretical uncertainties results in more robust uncertainty estimates for

[10] Different viscous corrections during particlization in the Pb+Pb system at LHC energies affect only the hadronic evolution after particlization. Since we take the source simulations as given, we exclude in the figure the computational cost incurred up to particlization. With this accounting, exploring the effects of different viscous corrections in the same collision system requires only $\mathcal{O}(800)$ CPU hours per design point on average, for both emulation methods.

[11] Accounting for the cost of generating the 100 full-model test samples would add about 100,000 CPU hours to each of the bars displayed in Fig. 6.

the QGP parameters. Accounting for model differences by Bayesian Model Averaging (BMA, as done in [11]) usually results in weaker constraints (broader posteriors) on the plasma properties, but does not account differentially for specific strengths and weaknesses of each model in different regions of parameter space. Bayesian Model Mixing [74, 75] has the potential to mitigate this shortcoming, leading to modeling uncertainties that lie between those of BMA and those of a single model analysis.

For both approaches, improved knowledge extraction comes at a steep computational cost. Mitigation calls for the development of increasingly efficient emulation techniques, to reduce as much as possible the need for computationally expensive runs of increasingly complex models. This work offers transfer learning as one such instrument in the Bayesian inference tool box with the potential for significant numerical cost savings. As shown in Sec. III, it addresses both the need for including more observables and for studying multiple variants of the theoretical model. By cutting the cost of Bayesian parameter estimation, we open the door to viable systematic analyses of measurements from heavy ion data from multiple collision systems, accounting for multiple sources of theoretical model uncertainties, and yielding increasingly accurate constraints on the properties of the plasma.

VII. CONCLUSIONS AND OUTLOOK

In this work we introduced and studied transfer learning as a novel method for training emulators for relativistic heavy ion collision simulations. We showed that this method is surprisingly effective and can significantly reduce the computational cost associated with building emulators. Furthermore, we saw that there is a wealth of information in the discrepancy GP which is a by-product of transfer learning methods and offers new ways of comparison between different simulation models. To decipher the information in the discrepancy GPs, we performed a global first order Sobol' sensitivity analysis in Sec. IV.

The transfer learning method introduced in this work has the limitation of requiring the same set of parameters in both the target and source models. We have ideas for a more general knowledge transferring framework that can handle different parameterizations of source and target, but this will have to wait for future work.

The field of relativistic heavy ion collisions has generated a multitude of different dynamical simulation models, and their number keeps growing. A systematic approach to accurately account for the theoretical uncertainties introduced by these model ambiguities is urgently needed from a statistical and information-theoretical perspective [75]. With the present contribution we hope to help lower the barrier to implementing such a paradigm change.

ACKNOWLEDGMENTS

We thank the JETSCAPE Collaboration for providing the relativistic heavy ion collision simulation data used in this work. D.L., D.E. and U.H. were supported by the NSF CSSI program under grant OAC-2004601, and within the framework of the JETSCAPE Collaboration under NSF Award No. ACI-1550223, as well as by the DOE Office of Science, Office for Nuclear Physics under Award No. DE-SC0004286. J.-F.P. acknowledges support by DOE Award No. DE-FG02-05ER41367. M.H. is supported by the Natural Sciences and Engineering Research Council of Canada.

Appendix A: Standardization of the observables

We standardize all simulation data before they are used to train the emulators. This is achieved by performing a standard normal transformation (A1) on the training and test data, using the means and variances of the predicted observables of our source model, i.e. for Pb+Pb collisions at $\sqrt{s_{\rm NN}} = 2.76\,\text{TeV}$ with Grad viscous corrections:

$$\tilde{Y}_j^l = \frac{Y_j^l - \mu_{\rm Grad}^l}{\sigma_{\rm Grad}^l}\,, \tag{A1}$$

$$\mu_{\rm Grad}^l = \sum_i \frac{Y_{i,\rm Grad}^l}{N_{\rm train}}\,, \quad (\sigma_{\rm Grad}^l)^2 = \sum_i \frac{\left(Y_{i,\rm Grad}^l - \mu_{\rm Grad}^l\right)^2}{N_{\rm train}}.$$

$Y_{i,\rm Grad}^l$ is the $l^{\rm th}$ observable from the source simulation i, and i is summed over all events in the training design.

[1] M. Gyulassy and L. McLerran, New forms of QCD matter discovered at RHIC, Nucl. Phys. A **750**, 30 (2005), arXiv:nucl-th/0405013.
[2] K. Yagi, T. Hatsuda, and Y. Miake, Quark-Gluon Plasma: From Big Bang to Little Bang, Cambridge Monogr. Part. Phys. Nucl. Phys. Cosmol. **23**, 1 (2005).
[3] H. Petersen, C. Coleman-Smith, S. A. Bass, and R. Wolpert, Constraining the initial state granularity with bulk observables in Au+Au collisions at $\sqrt{s_{\rm NN}} =$ 200 GeV, J. Phys. G **38**, 045102 (2011), arXiv:1012.4629 [nucl-th].
[4] J. Novak, K. Novak, S. Pratt, J. Vredevoogd, C. Coleman-Smith, and R. Wolpert, Determining Fundamental Properties of Matter Created in Ultrarelativistic Heavy-Ion Collisions, Phys. Rev. **C89**, 034917 (2014), arXiv:1303.5769 [nucl-th].
[5] E. Sangaline and S. Pratt, Toward a deeper understanding of how experiments constrain the underlying physics of heavy-ion collisions, Phys. Rev. **C93**, 024908 (2016), arXiv:1508.07017 [nucl-th].
[6] J. E. Bernhard, P. W. Marcy, C. E. Coleman-Smith, S. Huzurbazar, R. L. Wolpert, and S. A. Bass, Quantifying properties of hot and dense QCD matter through systematic model-to-data comparison, Phys. Rev. C **91**, 054910 (2015), arXiv:1502.00339 [nucl-th].

[7] J. E. Bernhard, J. S. Moreland, S. A. Bass, J. Liu, and U. Heinz, Applying Bayesian parameter estimation to relativistic heavy-ion collisions: simultaneous characterization of the initial state and quark-gluon plasma medium, Phys. Rev. **C94**, 024907 (2016), arXiv:1605.03954 [nucl-th].

[8] J. S. Moreland, J. E. Bernhard, and S. A. Bass, Bayesian calibration of a hybrid nuclear collision model using p-Pb and Pb-Pb data at energies available at the CERN Large Hadron Collider, Phys. Rev. C **101**, 024911 (2020), arXiv:1808.02106 [nucl-th].

[9] J. E. Bernhard, *Bayesian parameter estimation for relativistic heavy-ion collisions*, Ph.D. thesis, Duke U. (2018-04-19), arXiv:1804.06469 [nucl-th].

[10] J. E. Bernhard, J. S. Moreland, and S. A. Bass, Bayesian estimation of the specific shear and bulk viscosity of quark–gluon plasma, Nature Phys. **15**, 1113 (2019).

[11] D. Everett *et al.* (JETSCAPE), Phenomenological constraints on the transport properties of QCD matter with data-driven model averaging, Phys. Rev. Lett. **126**, 242301 (2021), arXiv:2010.03928 [hep-ph].

[12] G. Nijs, W. van der Schee, U. Gürsoy, and R. Snellings, Transverse momentum differential global analysis of heavy-ion collisions, Phys. Rev. Lett. **126**, 202301 (2021), arXiv:2010.15130 [nucl-th].

[13] G. Nijs, W. van der Schee, U. Gürsoy, and R. Snellings, Bayesian analysis of heavy ion collisions with the heavy ion computational framework Trajectum, Phys. Rev. C **103**, 054909 (2021), arXiv:2010.15134 [nucl-th].

[14] D. Everett *et al.* (JETSCAPE), Multisystem Bayesian constraints on the transport coefficients of QCD matter, Phys. Rev. C **103**, 054904 (2021), arXiv:2011.01430 [hep-ph].

[15] R. Trotta, Bayes in the sky: Bayesian inference and model selection in cosmology, Contemporary Physics **49**, 71–104 (2008).

[16] G. Peters, Markov Chain Monte Carlo: stochastic simulation for bayesian inference (2nd ed)., Statistics in Medicine **27**, 3213 (2008), https://onlinelibrary.wiley.com/doi/pdf/10.1002/sim.3240.

[17] T. J. Santner, B. J. Williams, W. I. Notz, and B. J. Williams, *The Design and Analysis of Computer Experiments* (Springer, 2003).

[18] F. Liu, E. Wang, X.-N. Wang, N. Xu, and B.-W. Zhang, eds., *The 28th International Conference on Ultra-relativistic Nucleus-Nucleus Collisions: Quark Matter 2019*, Nucl. Phys. **A1005** (2020).

[19] `https://indico.cern.ch/event/792436/`.

[20] S. Bass and A. Dumitru, Dynamics of hot bulk QCD matter: From the quark gluon plasma to hadronic freezeout, Phys. Rev. C **61**, 064909 (2000), arXiv:nucl-th/0001033.

[21] C. Nonaka and S. A. Bass, Space-time evolution of bulk QCD matter, Phys. Rev. **C75**, 014902 (2007), arXiv:nucl-th/0607018 [nucl-th].

[22] T. Hirano, U. Heinz, D. Kharzeev, R. Lacey, and Y. Nara, Mass ordering of differential elliptic flow and its violation for phi mesons, Phys. Rev. **C77**, 044909 (2008), arXiv:0710.5795 [nucl-th].

[23] H. Petersen, J. Steinheimer, G. Burau, M. Bleicher, and H. Stocker, A Fully Integrated Transport Approach to Heavy Ion Reactions with an Intermediate Hydrodynamic Stage, Phys. Rev. **C78**, 044901 (2008), arXiv:0806.1695 [nucl-th].

[24] H. Song, S. A. Bass, and U. Heinz, Viscous QCD matter in a hybrid hydrodynamic+Boltzmann approach, Phys. Rev. **C83**, 024912 (2011), arXiv:1012.0555 [nucl-th].

[25] U. Heinz, C. Shen, and H. Song, The viscosity of quark-gluon plasma at RHIC and the LHC, AIP Conf. Proc. **1441**, 766 (2012), arXiv:1108.5323 [nucl-th].

[26] H. Song, S. Bass, and U. Heinz, Spectra and elliptic flow for identified hadrons in 2.76A TeV Pb + Pb collisions, Phys. Rev. **C89**, 034919 (2014), arXiv:1311.0157 [nucl-th].

[27] X. Zhu, F. Meng, H. Song, and Y.-X. Liu, Hybrid model approach for strange and multistrange hadrons in 2.76A TeV Pb+Pb collisions, Phys. Rev. **C91**, 034904 (2015), arXiv:1501.03286 [nucl-th].

[28] S. Ryu, J.-F. Paquet, C. Shen, G. Denicol, B. Schenke, S. Jeon, and C. Gale, Effects of bulk viscosity and hadronic rescattering in heavy ion collisions at energies available at the BNL Relativistic Heavy Ion Collider and at the CERN Large Hadron Collider, Phys. Rev. **C97**, 034910 (2018), arXiv:1704.04216 [nucl-th].

[29] C. Gale, S. Jeon, and B. Schenke, Hydrodynamic modeling of heavy-ion collisions, International Journal of Modern Physics A **28**, 1340011 (2013), https://doi.org/10.1142/S0217751X13400113.

[30] M. C. Kennedy and A. O'Hagan, Bayesian calibration of computer models, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **63**, 425 (2001).

[31] M. C. Kennedy and A. O'Hagan, Predicting the output from a complex computer code when fast approximations are available, Biometrika **87**, 1 (2000).

[32] S. J. Pan and Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering **22**, 1345 (2010).

[33] A. Paleyes, M. Pullin, M. Mahsereci, N. Lawrence, and J. González, Emulation of physical processes with emukit, in *Second Workshop on Machine Learning and the Physical Sciences, NeurIPS* (2019).

[34] `https://github.com/danOSU/TransferLearningEmulation`.

[35] L. Torrey and J. Shavlik, Transfer learning, in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* (IGI global, 2010) pp. 242–264.

[36] `https://ftp.cs.wisc.edu/machine-learning/shavlik-group/torrey.handbook09.pdf`.

[37] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, Boosting for transfer learning, in *Proceedings of the 24th International Conference on Machine Learning*, ICML '07 (Association for Computing Machinery, New York, NY, USA, 2007) p. 193–200.

[38] D. Pardoe and P. Stone, Boosting for regression transfer, in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10 (Omnipress, Madison, WI, USA, 2010) p. 863–870.

[39] J. Garcke and T. Vanck, Importance weighted inductive transfer learning for regression, in *Machine Learning and Knowledge Discovery in Databases*, edited by T. Calders, F. Esposito, E. Hüllermeier, and R. Meo (Springer Berlin Heidelberg, Berlin, Heidelberg, 2014) pp. 466–481.

[40] B. Cao, S. J. Pan, Y. Zhang, D.-Y. Yeung, and Q. Yang, Adaptive transfer learning, in *proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 24 (2010).

[41] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, A comprehensive survey on transfer

learning, Proceedings of the IEEE **109**, 43 (2020).
[42] C. E. Rasmussen, Gaussian processes in machine learning, in *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*, edited by O. Bousquet, U. von Luxburg, and G. Rätsch (Springer Berlin Heidelberg, Berlin, Heidelberg, 2004) pp. 63–71.
[43] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, Design and Analysis of Computer Experiments, Statistical Science **4**, 409 (1989).
[44] S. Mak, C.-L. Sung, X. Wang, S.-T. Yeh, Y.-H. Chang, V. R. Joseph, V. Yang, and C. F. J. J. Wu, An efficient surrogate model for emulation and physics extraction of large eddy simulations, Journal of the American Statistical Association **113**, 1443 (2018).
[45] J. Chen, S. Mak, V. R. Joseph, and C. Zhang, Function-on-function kriging, with applications to three-dimensional printing of aortic tissues, Technometrics **63**, 384 (2021).
[46] G. Casella and R. L. Berger, *Statistical Inference* (Cengage Learning, 2021).
[47] J. S. Moreland, J. E. Bernhard, and S. A. Bass, Alternative ansatz to wounded nucleon and binary collision scaling in high-energy nuclear collisions, Phys. Rev. **C92**, 011901 (2015), arXiv:1412.4708 [nucl-th].
[48] `https://github.com/Duke-QCD/trento.git`.
[49] J. Liu, C. Shen, and U. Heinz, Pre-equilibrium evolution effects on heavy-ion collision observables, Phys. Rev. C **91**, 064906 (2015), [Erratum: Phys.Rev.C 92, 049904 (2015)], arXiv:1504.02160 [nucl-th].
[50] W. Broniowski, W. Florkowski, M. Chojnacki, and A. Kisiel, Free-streaming approximation in early dynamics of relativistic heavy-ion collisions, Phys. Rev. **C80**, 034902 (2009), arXiv:0812.3393 [nucl-th].
[51] `https://github.com/derekeverett/freestream-milne`.
[52] B. Schenke, S. Jeon, and C. Gale, (3+1)D hydrodynamic simulation of relativistic heavy-ion collisions, Phys. Rev. **C82**, 014903 (2010), arXiv:1004.1408 [hep-ph].
[53] B. Schenke, S. Jeon, and C. Gale, Elliptic and triangular flow in event-by-event (3+1)D viscous hydrodynamics, Phys. Rev. Lett. **106**, 042301 (2011), arXiv:1009.3244 [hep-ph].
[54] J.-F. Paquet, C. Shen, G. S. Denicol, M. Luzum, B. Schenke, S. Jeon, and C. Gale, Production of photons in relativistic heavy-ion collisions, Phys. Rev. **C93**, 044906 (2016), arXiv:1509.06738 [hep-ph].
[55] A. Kurganov and E. Tadmor, New High-Resolution Central Schemes for Nonlinear Conservation Laws and Convection-Diffusion Equations, Journal of Computational Physics **160**, 241 (2000).
[56] `http://www.physics.mcgill.ca/music/`.
[57] F. Cooper and G. Frye, Comment on the single particle distribution in the hydrodynamic and statistical thermodynamic models of multiparticle production, Phys. Rev. D **10**, 186 (1974).
[58] F. Cooper, G. Frye, and E. Schonberg, Landau's hydrodynamic model of particle production and electron positron annihilation into hadrons, Phys. Rev. **D11**, 192 (1975).
[59] H. Grad, On the kinetic theory of rarefied gases, Commun. Pure Appl. Math **2**, 331 (1949).
[60] S. Chapman, T. G. Cowling, and D. Burnett, *The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases* (Cambridge university press, 1990).
[61] S. Pratt and G. Torrieri, Coupling Relativistic Viscous Hydrodynamics to Boltzmann Descriptions, Phys. Rev. **C82**, 044901 (2010), arXiv:1003.0413 [nucl-th].
[62] M. McNelis, D. Everett, and U. Heinz, Particlization in fluid dynamical simulations of heavy-ion collisions: The iS3D module, Comput. Phys. Commun. **258**, 107604 (2021), arXiv:1912.08271 [nucl-th].
[63] `https://github.com/derekeverett/iS3D`.
[64] J. Weil *et al.*, Particle production and equilibrium properties within a new hadron transport approach for heavy-ion collisions, Phys. Rev. **C94**, 054905 (2016), arXiv:1606.06642 [nucl-th].
[65] `https://github.com/smash-transport/smash`.
[66] M. D. Morris and T. J. Mitchell, Exploratory designs for computational experiments, Journal of Statistical Planning and Inference **43**, 381 (1995).
[67] J. E. Parkkila, A. Onnerstad, F. Taghavi, C. Mordasini, A. Bilandzic, and D. J. Kim, New constraints for qcd matter from improved bayesian parameter estimation in heavy-ion collisions at lhc (2021), arXiv:2111.08145 [hep-ph].
[68] D. Everett, *Quantifying the quark-gluon plasma*, Ph.D. thesis, The Ohio State University (2021), arXiv:2107.11362 [hep-ph].
[69] B. Iooss and P. Lemaître, A review on global sensitivity analysis methods, in *Uncertainty Management in Simulation-Optimization of Complex Systems* (Springer, 2015) pp. 101–122.
[70] J. Jacques, C. Lavergne, and N. Devictor, Sensitivity analysis in presence of model uncertainty and correlated inputs, Reliability Engineering & System Safety **91**, 1126 (2006).
[71] I. M. Sobol', On sensitivity estimation for nonlinear mathematical models, Matematicheskoe modelirovanie **2**, 112 (1990).
[72] S. IM, Sensitivity estimates for nonlinear mathematical models, Math. Model. Comput. Exp **1**, 407 (1993).
[73] E. Borgonovo, S. Tarantola, E. Plischke, and M. D. Morris, Transformations and invariance in the sensitivity analysis of computer experiments, Journal of the Royal Statistical Society: Series B **76**, 925 (2014).
[74] J. R. Coleman, *Topics in Bayesian computer model emulation and calibration, with applications to high-energy particle collisions*, Ph.D. thesis, Duke University, Department of Statistical Science (2019).
[75] D. R. Phillips *et al.*, Get on the BAND Wagon: A bayesian framework for quantifying model uncertainties in nuclear dynamics, J. Phys. G **48**, 072001 (2021), arXiv:2012.07704 [nucl-th].
[76] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning* (Springer Series in Statistics, 2001).