MESSY Estimation: Maximum-Entropy based Stochastic and Symbolic densitY Estimation

Anonymous authors
Paper under double-blind review

Abstract

We introduce **MESSY** estimation, a **M**aximum-**E**ntropy based **S**tochastic and **S**ymbolic densit**Y** estimation method. The proposed approach recovers probability density functions symbolically from samples using moments of a Gradient flow in which the ansatz serves as the driving force. In particular, we construct a gradient-based drift-diffusion process that connects samples of the unknown distribution function to a guess symbolic expression. We then show that when the guess distribution has the maximum entropy form, the parameters of this distribution can be found efficiently by solving a linear system of equations constructed using the moments of the provided samples. Furthermore, we use Symbolic regression to explore the space of smooth functions and find optimal basis functions for the exponent of the maximum entropy functional leading to good conditioning. The cost of the proposed method for each set of selected basis functions is linear with the number of samples and quadratic with the number of basis functions. However, the underlying acceptance/rejection procedure in finding optimal and well-conditioned bases adds to the computational cost. We validate the proposed MESSY estimation method against other benchmark methods for the case of a bi-modal and a discontinuous density, as well as a density at the limit of physical realizability. We find that the addition of a symbolic search for basis functions improves the accuracy of the estimation at a reasonable additional computational cost. Our results suggest that the proposed method outperforms existing density recovery methods in the limit of a small to moderate number of samples by providing a low-bias and tractable symbolic description of the unknown density at a reasonable computational cost.

1 Introduction

Recovering probability density functions from samples is one of the fundamental problems in statistics with many applications. For example, the traditional task of discovering the underlying dynamics governing the corresponding distribution function is strongly dependent on the quality of the density estimator (Rudy et al., 2017). Applications include particle physics (Patrignani et al., 2016), boundary conditions for multi-scale kinetic problems (Frezzotti et al., 2005; Kon et al., 2014), and machine learning (Song et al., 2020).

Broadly speaking, two categories of methods have been developed for this task: parametric and non-parametric estimators. While parametric methods assume a restrictive ansatz for the underlying distribution function, non-parametric methods provide a more flexible density estimate by performing a kernel integration locally using nearby samples. Although non-parametric methods do not need any prior knowledge of the underlying distribution, they suffer from the unclear choice of kernel and its support leading to bias and lack of moment conservation. Examples of non-parametric density estimators include histogram and Kernel Density Estimation (KDE) (Rosenblatt, 1956; Jones et al., 1996; Sheather, 2004).

On the other hand, parametric density estimators may allow conservation of moments while introducing modeling error, since a guess for the distribution is required. Parametric distributions include Gaussian, orthogonal expansion with respect to Gaussian using Hermite polynomials (also known as Grad's ansatz in kinetic theory) (Hermite, 1864; Grad, 1949; Cai et al., 2015), wavelet density estimation Donoho et al. (1996), and Maximum Entropy Distribution (MED) (Kapur, 1989; Tagliani, 1999; Khinchin, 2013; Hauck et al., 2008) function among others. Given only the mean and variance, information theory provides us with the Gaussian distribution function as the least biased density, which has been used intensively in the literature as it appears in many applications. However, including higher order moments in a similar way, i.e. *moment problem*, raises further complications. For example in the context of kinetic theory, Grad proposed a closure that incorporates higher-order moments by considering a deviation from Gaussian using Hermite polynomials. Even though the information from higher moments is incorporated as the parameters of the polynomial expansion in Grad's ansatz, such a formulation suffers from not guaranteeing positivity of the estimated density along with the introduction of bias.

Among parametric density estimators, the Maximum Entropy Distribution (MED) function has been proposed in information theory as the least biased density estimate given a number of moments of the unknown distribution (Kapur, 1989). While MED provides the least biased density estimate, it suffers from two limitations. First, the distribution parameters (Lagrange multipliers) can only be found by solving a convex optimization problem with ill-conditioned Hessian (Dreyer, 1987; Levermore, 1996). The condition number increases either by increasing the order of the matching moments or approaching the limit of physical realizability which motivated the use of adaptive basis functions (Abramov, 2007; 2009). Second, MED only exists and is unique in bounded domains. While existence/uniqueness is guaranteed for recovering the distribution in the subspace occupied by the samples, the computational complexity associated with the direct computation of Lagrange multipliers has prevented researchers from deploying MED in practice.

Related methods. The problem of recovering a distribution function from samples has been investigated and studied before. We briefly review some of the work most relevant to our paper:

Data-driven maximum entropy distribution function: Several attempts have been made in the literature to speed up the computation of Lagrange multipliers for MED using Neural Networks (Sadr et al., 2021; Porteous et al., 2021; Schotthöfer et al., 2022) and Gaussian process regression (Sadr et al., 2020). Unfortunately, these approaches are data-dependent with support only on the trained subspace of distributions. Similar to the standard MED and other related closures, the data-driven MED can only handle polynomial moments as input, even though the data may be better represented with moments of other basis functions.

Learning an invertible map: The idea is to train an invertible neural network that maps the samples to a known distribution function. Then the unknown distribution function is found by inverting the trained map with the known distribution as the input. This procedure is called the normalizing flow technique (Rezende & Mohamed, 2015; Dinh et al., 2016; Kingma & Dhariwal, 2018; Durkan et al., 2019; Tzen & Raginsky, 2019; Kobyzev et al., 2020; Wang & Marzouk, 2022). This method has been used for re-sampling unknown distributions, e.g. Boltzmann generators (Noé et al., 2019), as well as density recovery such as AI-Feynmann (Udrescu & Tegmark, 2020; Udrescu et al., 2020). We note that AI-Feynman does not obtain the density from the samples directly; instead it first fits a density to the samples using the normalizing flow technique, constructs an input/output data set, then finds a simpler expression using symbolic regression. While invertible maps can be used to accurately predict densities, they can become expensive since for each problem one has to learn the parameters of the considered map via optimization.

Diffusion map: Instead of training for an invertible map, the diffusion map (Coifman et al., 2005; Coifman & Lafon, 2006) constructs coordinates using eigenfunctions of Markov matrices. Using pairwise distances between samples, in this method a kernel matrix is constructed as a generator of the underlying Langevin diffusion process. As shown by Li & Marzouk (2023), one can generate samples of the target distribution using Laplacian-adjusted Wasserstein gradient descent (Chewi et al., 2020). Unfortunately, this approach can become computationally expensive since it requires singular value decomposition of matrices of size equal to the number of samples.

Gradient flow: The gradient flow method has gained attention in recent years (Villani, 2009; Song et al., 2020; Song & Ermon, 2020). In particular, a class of sampling methods has been devised for drawing samples from a given distribution function using Langevin dynamics with the gradient of log-density as the driving force (Liu, 2017; Garbuno-Inigo et al., 2020a;b). Yet, this approach does not provide the density of the samples by itself. In our paper, we benefit from this formulation to recover the parameters of a density ansatz.

KDE via diffusion: In this method, the bandwidth of the kernel density estimation is computed using the minimum of mean integrated squared error and the fact that the KDE is the fundamental solution to a heat (more precisely Fokker-Planck) equation (Botev, 2007; Botev et al., 2010). While improvement has been achieved in this direction, we note that the KDE-diffusion method suffers from smoothing effects which introduce bias. Moreover moments of the unknown distribution are not necessarily matched.

Symbolic regression: Symbolic regression (SR) is a challenging task in machine learning that aims to identify analytical expressions that best describe the relationship between inputs and outputs of a given dataset. SR does not require any prior knowledge about the model structure. Traditional regression methods such as least squares (Wild & Seber, 1989), likelihood-based (Edwards, 1984; Pawitan, 2001), and Bayesian regression techniques (Lee, 1997; Leonard & Hsu, 2001; Tohme et al., 2020) use fixed parametric model structure and only optimize for model parameters. SR optimizes for model structure and parameters simultaneously and hence is thought to be NP-hard, i.e. Non-deterministic Polynomial-time hard, (Udrescu & Tegmark, 2020; Petersen et al., 2021; Virgolin & Pissis, 2022). The SR problem has gained significant attention over recent years (Orzechowski et al., 2018; La Cava et al., 2021), and several approaches have been suggested in the literature. Most methods adopt genetic algorithms (Koza & Koza, 1992; Schmidt & Lipson, 2009; Tohme et al., 2023). Lately, researchers proposed using machine learning algorithms (e.g. Bayesian optimization, nonlinear least squares, neural networks, transformers, etc.) to solve the SR problem (Sahoo et al., 2018; Jin et al., 2019; Udrescu et al., 2020; Cranmer et al., 2020; Kommenda et al., 2020; Burlacu et al., 2020; Biggio et al., 2021; Mundhenk et al., 2021; Petersen et al., 2021; Valipour et al., 2021; Zhang et al., 2022; Kamienny et al., 2022). While most SR methods are concerned with finding a map from the input to the output, very few have addressed the problem of discovering probability density functions from samples (Udrescu et al., 2020).

Our Contributions. Our work improves the efficiency in determining the maximum entropy result for the unknown distribution. We specifically develop a new method for determining the unknown parameters (Lagrange multipliers) of this distribution without solving the optimization problem associated with this approach. This is achieved by relating the samples to the MED using Gradient flow, with the grad-log of the MED guess distribution serving as the drift. This results in a linear inverse problem for the Lagrange multipliers that is significantly easier to solve compared to the aforementioned optimization problem. We also propose a Monte Carlo search in the space of smooth functions for finding an optimal basis function for describing (the exponent of) the maximum entropy ansatz. As a selection criterion, we rate randomly created basis functions according to the condition number associated with the coefficient (Hessian) matrix of the inverse problem for the Lagrange multipliers. This helps to maintain good conditioning, which allows us to incorporate more degrees of freedom and recover the unknown density accurately. Discontinuous density functions are treated by considering only the domain supported by data and using a multi-level solution process.

The paper is organized as follows. In Section 2 we review the concept of Gradient flow with grad-log of a known density as the drift. In Section 3, we show how parameters of a guess MED may be found by computing the relaxation rates of the corresponding Gradient flow. Using the maximum entropy ansatz, in Section 4 we derive a linear inverse problem for finding the Lagrange multipliers without the need for solving an optimization problem. In Section 5, we propose a symbolic regression method for finding basis functions that can be used to increase degrees of freedom while maintaining good conditioning of the problem by construction. In Section 6, we propose a generalization of the maximum entropy ansatz that allows including further degrees of freedom in a multi-level fashion. Section 7 presents the complete MESSY algorithm. In Section 8, we validate MESSY by comparing its predictions to those of benchmark density estimators in recovering distributions featuring discontinuities and bi-modality, as well as distributions close to the limit of realizability. Finally, in Section 9, we offer our conclusions and outlook.

2 Gradient flow and theoretical motivation

Consider a set of samples of a random variable $\boldsymbol{X}$ from an unknown density distribution function $f(\boldsymbol{x})$. Let our guess for this distribution function, the "ansatz", be denoted by $\hat{f}(\boldsymbol{x})$.

Instead of constructing a non-parametric approximation of the target density numerically from samples of $\boldsymbol{X}$ (like histogram or KDE) and then calculating its difference from the guess density $\hat{f}$, in this work we suggest measuring the distance using transport. In particular, we use the fact that the steady-state distribution of $\boldsymbol{X}(t)$ which follows the stochastic differential equation (SDE)

$$d\boldsymbol{X} = \nabla_{\boldsymbol{x}}\big[\log\big(\hat{f}\,\big)\big]dt + \sqrt{2}d\boldsymbol{W}_t \tag{1}$$

is the distribution $\hat{f}$. Here, $\boldsymbol{W}_t$ is the standard Wiener process of dimension $\dim(\boldsymbol{x})$. We note that Eq. 1 is known as the gradient flow (or score-based generative model) with grad-log of density as the force (Liu, 2017; Song et al., 2020).

The distance of f from $\hat{f}$ may be measured by the time required for the SDE with $\boldsymbol{X}(t=0) \sim f$ to reach steady state. Alternatively, one may compare the moments computed from the solution to $\boldsymbol{X}(t)$ against the input samples to measure this distance. Both these approaches are subject to numerical and statistical noise associated with the numerical scheme deployed in integrating Eq. 1. In the next section, we derive an efficient way of computing the parameters of our approximation $\hat{f}$ based on these ideas. We also show that the transition from f to $\hat{f}$ is monotonic.

3 Ansatz as the target density of Gradient flow

According to Ito's lemma (Platen & Bruti-Liberati, 2010) the transition of f to $\hat{f}$ is governed by the Fokker-Planck equation

$$\frac{\partial f}{\partial t} = \nabla_{\boldsymbol{x}}\Big[\hat{f}\,\nabla_{\boldsymbol{x}}[f/\hat{f}\,]\Big] \tag{2}$$

$$= -\nabla_{\boldsymbol{x}}\cdot\Big[\nabla_{\boldsymbol{x}}\big[\log\big(\hat{f}\,\big)\big]f\Big] + \nabla^2_{\boldsymbol{x}}\Big[f\Big]. \tag{3}$$

Proposition 3.1. *The distribution function $f(t)$ governed by the Fokker-Planck Eq. 2 converges to $\hat{f}$ as $t \to \infty$. Furthermore, the cross entropy distance between f and $\hat{f}$ monotonically decreases during this transition.*

Proof. Let us multiply both sides of Eq. 2 by $\log(f/\hat{f})$ and take the integral with respect to $\boldsymbol{x}$ in order to obtain the evolution of the cross-entropy $S = \int f\log(f/\hat{f})d\boldsymbol{x}$. It follows that

$$\begin{aligned}
\frac{dS}{dt} &= \int \log(f/\hat{f})\,\nabla_{\boldsymbol{x}}\Big[\hat{f}\,\nabla_{\boldsymbol{x}}[f/\hat{f}]\Big]\,d\boldsymbol{x}\\
&= \int \nabla_{\boldsymbol{x}}\Big[\hat{f}\log(f/\hat{f})\nabla_{\boldsymbol{x}}[f/\hat{f}]\Big]d\boldsymbol{x} - \int \hat{f}\,\nabla_{\boldsymbol{x}}[\log(f/\hat{f})]\cdot\nabla_{\boldsymbol{x}}[f/\hat{f}]d\boldsymbol{x}\\
&= \underbrace{\int \nabla_{\boldsymbol{x}}\Big[\hat{f}\log(f/\hat{f})\,\frac{f}{\hat{f}}\nabla_{\boldsymbol{x}}[\log(f/\hat{f})]\Big]d\boldsymbol{x}}_{=0} - \int \hat{f}\,\nabla_{\boldsymbol{x}}[\log(f/\hat{f})]\cdot\frac{f}{\hat{f}}\,\nabla_{\boldsymbol{x}}[\log(f/\hat{f})]d\boldsymbol{x}\\
&= -\sum_{i=1}^{\dim(\boldsymbol{x})}\int f\Big(\nabla_{x_i}[\log(f/\hat{f})]\Big)^2 d\boldsymbol{x} \le 0\ .
\end{aligned} \tag{4}$$

Here, we use the regularity condition that $f\log(f/\hat{f})\nabla_{\boldsymbol{x}}\log(f/\hat{f}) \to 0$ as $\boldsymbol{x} \to \infty$. Therefore, given any initial condition for f at $t=0$, the cross-entropy distance between f and $\hat{f}$ following the Fokker-Planck in Eq. 2 monotonically decreases until it reaches the steady-state with the trivial fixed point $f \to \hat{f}$ as $t \to \infty$. For details, see (Liu, 2017). ☐

With applications to high-dimensional problems in mind, instead of looking for solutions of Eq. 2 we choose to work with appropriate empirical moments of this equation, which can be evaluated from the available samples. As will be seen below, this approach lends itself to a very effective method for determining $\hat{f}$.

Let us denote a vector of basis functions in $\mathbb{R}^{\dim(\boldsymbol{x})}$ by $\boldsymbol{H}(\boldsymbol{x})$. By multiplying both sides of Eq. 3 by $\boldsymbol{H}(\boldsymbol{x})$ and integrating with respect to $\boldsymbol{x}$, we obtain the evolution equation for the moments, also known as the relaxation rates,

$$\frac{d}{dt}\Big[\int \boldsymbol{H} f d\boldsymbol{x}\Big] = -\int \boldsymbol{H}\nabla_{\boldsymbol{x}} \cdot \Big[\nabla_{\boldsymbol{x}}[\log(\hat{f})]f\Big] d\boldsymbol{x} + \int \boldsymbol{H}\nabla^2_{\boldsymbol{x}}\Big[f\Big] d\boldsymbol{x} \ . \tag{5}$$

Assuming that the underlying density f is integrable in $\mathbb{R}^{\dim(\boldsymbol{x})}$ and $f\boldsymbol{H} \to \boldsymbol{0}$ as $\boldsymbol{x} \to \infty$, which is implied by the existence of moments, we use integration by parts to obtain

$$\frac{d}{dt}\Big[\int \boldsymbol{H} f d\boldsymbol{x}\Big] = \int \nabla_{\boldsymbol{x}}[\boldsymbol{H}] \cdot \nabla_{\boldsymbol{x}}[\log(\hat{f})] f d\boldsymbol{x} + \int \nabla^2_{\boldsymbol{x}}[\boldsymbol{H}] f d\boldsymbol{x} \ . \tag{6}$$

Given samples of f, one can compute the relaxation rates of moments represented by Eq. 6 as a measure of the difference between $\hat{f}$ and f. These relaxation rates can be used as the gradient in the search for parameters of a given ansatz, i.e.

$$\boldsymbol{g}(t) = \frac{d}{dt}\Big\langle \boldsymbol{H}(\boldsymbol{X}(t))\Big\rangle = \Big\langle \nabla_{\boldsymbol{x}}[\boldsymbol{H}(\boldsymbol{X}(t))] \cdot \nabla_{\boldsymbol{x}}[\log(\hat{f}(\boldsymbol{X}(t)))]\Big\rangle + \Big\langle \nabla^2_{\boldsymbol{x}}[\boldsymbol{H}(\boldsymbol{X}(t))]\Big\rangle \ . \tag{7}$$

In the above, $\langle \phi(\boldsymbol{X})\rangle$ denotes the unbiased empirical measure for the expectation of $\phi(\boldsymbol{X})$ which is computed using samples of $\boldsymbol{X}_i$, for $i = 1, ..., N$ via $\langle \phi(\boldsymbol{X})\rangle = \frac{1}{N}\sum_{i=1}^{N} \phi(\boldsymbol{X}_i)$.

Here we note that the Hessian for this optimization can be obtained (Liu, 2017) using the samples of the unknown distribution by taking the derivative of this gradient with respect to the parameters $\boldsymbol{\theta}$ of the ansatz $\hat{f}$, namely

$$\boldsymbol{L}(t) = \nabla_{\boldsymbol{\theta}}[\boldsymbol{g}] = \Big\langle \nabla_{\boldsymbol{x}}[\boldsymbol{H}(\boldsymbol{X}(t))] \cdot \nabla_{\boldsymbol{\theta}}\Big[\nabla_{\boldsymbol{x}}[\log(\hat{f}(\boldsymbol{X}(t)))]\Big]\Big\rangle \ . \tag{8}$$

In what follows we develop an approach that uses this observation to bring computational benefits to the solution of the maximum entropy problem.

4 Maximum Entropy Distribution as an ansatz for the gradient flow

In this work, we use the maximum entropy distribution function as our parameterized ansatz for $\hat{f}$, i.e.

$$\hat{f}(\boldsymbol{x}) = Z^{-1}\exp\big(\boldsymbol{\lambda}\cdot\boldsymbol{H}(\boldsymbol{x})\big) \tag{9}$$

where $Z = \int \exp(\boldsymbol{\lambda}\cdot\boldsymbol{H}(\boldsymbol{x}))d\boldsymbol{x}$ is the normalization constant. The motivation for choosing this family of distributions is the fact that this is the least-biased distribution for the moment problem, provided the given moments are matched.

Definition 4.1. *Moment problem*

The problem of finding a distribution function $f(\boldsymbol{x})$ given its moments $\int \boldsymbol{H}(\boldsymbol{x})f(\boldsymbol{x})d\boldsymbol{x} = \boldsymbol{\mu}$ for the vector of basis functions $\boldsymbol{H}(\boldsymbol{x})$ will be referred to as the moment problem.

In particular, the density in Eq. (9) is the extremum of the loss functional that minimizes the Shannon entropy with constraints on moments $\boldsymbol{\mu}$ using the method of Lagrange multipliers, i.e.

$$\hat{f}(\boldsymbol{x}) = \underset{\mathcal{F}\in\mathcal{K}}{\arg\min}\ \mathcal{C}[\mathcal{F}(\boldsymbol{x})] \tag{10}$$

$$\text{where} \quad \mathcal{C}[\mathcal{F}(\boldsymbol{x})] := \int \mathcal{F}(\boldsymbol{x})\log(\mathcal{F}(\boldsymbol{x}))d\boldsymbol{x} + \sum_{i=1}^{N_b}\lambda_i\left(\int H_i(\boldsymbol{x})\mathcal{F}(\boldsymbol{x})d\boldsymbol{x} - \mu_i(\boldsymbol{x})\right) \ . \tag{11}$$

Here $\mathcal{K}$ denotes the space of probability density functions with measurable moments; see (Kapur, 1989) and Appendix A for more details. In this paper, we denote the number of considered basis functions by N_b, while N_m denotes the highest order of these basis functions. For instance, in the case of traditional one-dimensional random variable where polynomial basis functions are deployed, i.e. $\boldsymbol{H} = \left[x, x^2, ..., x^{N_m}\right]$, we have $N_m = N_b$. Here, we use the following definition for the growth rate of a basis function.

Definition 4.2. *Growth rate of n-th order*

A function $\psi(x)$ has the growth-rate of n-th order if $|\psi(x)| \leq Cx^n$ for all $x \geq x_0$ where $C \in \mathbb{R}^+$ and $x_0 \in \mathbb{R}$. This is often denoted by $\psi(x) = \mathcal{O}(x^n)$.

Substituting Eq. 9 for $\hat{f}$ in Eq. 7 results in the relaxation rate

$$\boldsymbol{g}(t) = \sum_{i=1}^{\dim(\boldsymbol{x})} \left\langle \nabla_{x_i}\left[\boldsymbol{H}\big(\boldsymbol{X}(t)\big)\right] \otimes \nabla_{x_i}\left[\boldsymbol{H}\big(\boldsymbol{X}(t)\big)\right] \right\rangle \boldsymbol{\lambda} + \sum_{i=1}^{\dim(\boldsymbol{x})} \left\langle \nabla^2_{x_i}\left[\boldsymbol{H}\big(\boldsymbol{X}(t)\big)\right] \right\rangle , \tag{12}$$

where $\otimes$ indicates the outer product. Let us define the matrix $\boldsymbol{L}^{\mathrm{ME}}$ as

$$\boldsymbol{L}^{\mathrm{ME}}(t) := \sum_{i=1}^{\dim(\boldsymbol{x})} \left\langle \nabla_{x_i}\left[\boldsymbol{H}\big(\boldsymbol{X}(t)\big)\right] \otimes \nabla_{x_i}\left[\boldsymbol{H}\big(\boldsymbol{X}(t)\big)\right] \right\rangle . \tag{13}$$

We note that the matrix $\boldsymbol{L}^{\mathrm{ME}}$ is the Hessian of the optimization problem with gradient given by Eq. 12 which is positive definite, making the underlying optimization problem convex.

Proposition 4.3. *The Hessian matrix $\boldsymbol{L}^{\mathrm{ME}}$ is symmetric positive definite. As a result, the optimization problem with gradient given by Eq. (12) and Hessian matrix given by Eq. 13 is strictly convex.*

Proof. Clearly, the Hessian matrix defined by Eq. 13 is symmetric, i.e. $L^{\mathrm{ME}}_{i,j} = L^{\mathrm{ME}}_{j,i}\ \forall i, j = 1, ..., N_b$. We further note that this matrix is positive definite, i.e. for any non-zero vector $\boldsymbol{w} \in \mathbb{R}^{N_b}$ we can write

$$\begin{aligned} \boldsymbol{w}^T \boldsymbol{L}^{\mathrm{ME}}(t) \boldsymbol{w} &= \sum_{i=1}^{\dim(\boldsymbol{x})} \left\langle \boldsymbol{w}^T \nabla_{x_i}\left[\boldsymbol{H}\big(\boldsymbol{X}(t)\big)\right]\ \nabla_{x_i}\left[\boldsymbol{H}\big(\boldsymbol{X}(t)\big)\right]^T \boldsymbol{w} \right\rangle & (14) \\ &= \sum_{i=1}^{\dim(\boldsymbol{x})} \left\langle \Big(\boldsymbol{w}^T \nabla_{x_i}\left[\boldsymbol{H}\big(\boldsymbol{X}(t)\big)\right]\Big)^2 \right\rangle > 0 . & (15) \end{aligned}$$

Given the Hessian is symmetric positive definite, we conclude that the underlying optimization problem is convex (Chong & Zak, 2013). ☐

When the matrix $\boldsymbol{L}^{\mathrm{ME}}$ is well-conditioned, we can directly compute the Lagrange multipliers using samples without the need for solving an optimization problem. This can be achieved by solving Eq. 12 for the Lagrange multipliers

$$\boldsymbol{L}^{\mathrm{ME}}(t)\boldsymbol{\lambda} = \boldsymbol{g}(t) - \sum_{i=1}^{\dim(\boldsymbol{x})} \left\langle \nabla^2_{x_i}\left[\boldsymbol{H}\big(\boldsymbol{X}(t)\big)\right] \right\rangle \tag{16}$$

for a given relaxation rate $\boldsymbol{g}$.

We proceed by noting that a convenient way for determining the parameters of $\hat{f}$ is to set $\hat{f} = f(t = 0)$ in the above formulation, or in other words, require that the given samples are also samples of $\hat{f}$ as given. This corresponds to the steady solution of Eq. 12 , namely $\boldsymbol{g} \to \boldsymbol{0}$, which implies the remarkably simple result

$$\boldsymbol{\lambda} = -\left(\boldsymbol{L}^{\mathrm{ME}}\right)^{-1} \left(\sum_{i=1}^{\dim(\boldsymbol{x})} \left\langle \nabla^2_{x_i}\left[\boldsymbol{H}\big(\boldsymbol{X}(t=0)\big)\right] \right\rangle \right) , \tag{17}$$

which implies a closed-form solution for the Lagrange multipliers through the above linear problem.

While Eq. 17 analytically recovers the Lagrange multipliers $\boldsymbol{\lambda}$ directly from samples of $\boldsymbol{X}$, it still requires inverting the matrix $\boldsymbol{L}^{\mathrm{ME}}$ which may be ill-conditioned (Abramov, 2010; Alldredge et al., 2014). This means that the resulting Lagrange multipliers may become sensitive to noise in the samples and the choice of the basis functions. In order to cope with this issue, we propose computing $\boldsymbol{\lambda}$ as outlined below.

Orthonormalizing the basis functions. We construct an orthonormal basis function with respect to $\boldsymbol{X} \sim f$ using the modified Gram-Schmidt algorithm as described in Algorithm 1. We deploy the orthonormal basis functions from the Gram-Schmidt procedure to construct $\nabla_{\boldsymbol{x}}[\boldsymbol{H}]^{\perp}$, i.e. $\nabla_{\boldsymbol{x}}[\boldsymbol{H}]$ is the input to Algorithm 1, and by integration we obtain $\boldsymbol{H}^{\perp}$. This leads to a well-conditioned matrix $\boldsymbol{L}^{\mathrm{ME}}$, since the resulting matrix should be close to identity $\boldsymbol{L}^{\mathrm{ME}} \approx \boldsymbol{I}$ with condition number $\mathrm{cond}\big(\boldsymbol{L}^{\mathrm{ME}}\big) \approx 1$ subject to round-off error. We note that the cost of this algorithm is quadratic with the number of basis functions and linear with the number of samples.

Algorithm 1: Modified Gram-Schmidt: Given a vector of basis functions $\boldsymbol{\phi}$, this algorithm constructs an orthonormal basis functions $\boldsymbol{\phi}^{\perp}$ with respect to f such that $\langle \boldsymbol{\phi}^{\perp}(\boldsymbol{X}) \otimes \boldsymbol{\phi}^{\perp}(\boldsymbol{X}) \rangle \approx \boldsymbol{I}$ using the modified Gram-Schmidt procedure (Giraud et al., 2002; Abramov, 2010).

```
Input: $\boldsymbol{\phi}$
Initialize $\boldsymbol{\phi}^{\perp} \leftarrow \boldsymbol{\phi}$;
for $i = 1, ..., \dim(\boldsymbol{\phi})$ do
    $\phi_i^{\perp} = \phi_i^{\perp} / \sqrt{\langle (\phi_i^{\perp}(\boldsymbol{X}))^2 \rangle}$;
    for $j = i+1, ..., \dim(\boldsymbol{\phi})$ do
        $\phi_j^{\perp} \leftarrow \phi_j^{\perp} - \langle \phi_i^{\perp}(\boldsymbol{X}) \phi_j^{\perp}(\boldsymbol{X}) \rangle \phi_i^{\perp}$;
    end
end
Return $\boldsymbol{\phi}^{\perp}$
```

4.1 Comparing the proposed formulation to standard Maximum Entropy Distribution

Here we point out several advantages of using the proposed loss function compared to the standard maximum entropy closure.

- **A closed-form solution:** By setting the relaxation rate of the moments to zero, the Lagrange multipliers can be computed directly from samples $\boldsymbol{X} \sim f$ without the need for the line-search associated with the Newton method.
- **Avoiding the curse of dimensionality:** The proposed method takes full advantage of having access to the samples of the unknown distribution function. In particular, we compute the orthonormal basis function, gradient, and Hessian using the samples of $\boldsymbol{X}$. This use of the Monte Carlo integration method avoids the curse of high dimensionality, as the cost scales linearly with the number of dimensions. This is a considerable advantage compared to the standard MED where the integrals need to be computed accurately, e.g. using the quadrature rule.
- **Relaxed existence requirements:** The search for the Lagrange multiplier through the SDE process does not place existence requirements on intermediate iterates of the distribution function. In other words, there is no need for the intermediate iterates of the Lagrange multipliers to be realizable. This is a significant advantage compared to the standard MED, where the line search may fail as the distribution associated with the intermediate $\boldsymbol{\lambda}$ may not exist (not integrable).
- **Reducing the condition number:** For the case where $\boldsymbol{H}$ is a vector of polynomial basis functions, we expect a smaller condition number compared to the standard moment problem. This is because of the order reduction in the moments of the Hessian where moments of $\nabla_{\boldsymbol{x}}[H_i] \nabla_{\boldsymbol{x}}[H_j]^T$ are computed rather than $H_i H_j$.

5 Symbolic-Based Maximum Entropy Distribution

In the standard moment problem it is common to consider polynomials for the moment functions in $\boldsymbol{H}$, i.e. $\boldsymbol{H} = \left[x, x^2, \ldots\right]$, even though other basis functions may better represent the unknown distribution. Additionally, such polynomial basis functions are notorious for resulting in ill-conditioned solution processes. For these reasons, we introduce a symbolic regression approach to introduce some diversity and ultimately optimize over our use of basis functions. As we will see in the next section, adding the symbolic search to our MED description improves the accuracy, convergence, and robustness of the density recovery problem.

Before diving into the proposed method, we first briefly review the general task of symbolic regression.

Definition 5.1. *Symbolic Regression (SR) problem*

Given a metric $\mathcal{L}$ and a dataset $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$ consisting of N independent identically distributed (i.i.d.) paired samples, where $\boldsymbol{x}_i \in \mathbb{R}^{\dim(\boldsymbol{x})}$ and $y_i \in \mathbb{R}$, the SR problem searches in the space of functions $\mathcal{S}$ for a function $\psi^*(\boldsymbol{x})$ which minimizes $\sum_{i=1}^N \mathcal{L}\big(y_i, \psi(\boldsymbol{x}_i)\big)$ where $\psi \in \mathcal{S}$.

In order to deploy the SR method for the density recovery, we need to restrict the space of functions $\mathcal{S}$ to those which satisfy non-negativity, normalization and existence of moments with respect to the vector of linearly independent (polynomial) basis functions $\boldsymbol{R}$. The space of such distributions can be defined as

$$\mathcal{S}_{f|\boldsymbol{R}} := \left\{ f(\boldsymbol{x}) \in \mathcal{S} \;\middle|\; f(\boldsymbol{x}) \geq 0 \; \forall \boldsymbol{x} \in \mathbb{R}^{\dim(\boldsymbol{x})}, \int_{\mathbb{R}^{\dim(\boldsymbol{x})}} f(\boldsymbol{x}) \; d\boldsymbol{x} = 1, \int_{\mathbb{R}^{\dim(\boldsymbol{x})}} \boldsymbol{R}(\boldsymbol{x}) f(\boldsymbol{x}) \; d\boldsymbol{x} < +\infty \right\} . \tag{18}$$

In order to ensure non-negativity, motivated by the MED formulation, we consider $\hat{f}$ to be exponential, i.e.

$$\hat{f}(\boldsymbol{x}) \propto \exp\left(\mathcal{G}(\boldsymbol{x})\right) \qquad \Longleftrightarrow \qquad \log\left(\hat{f}(\boldsymbol{x})\right) \propto \mathcal{G}(\boldsymbol{x}) \; , \tag{19}$$

where $\mathcal{G}(\boldsymbol{x})$ is an analytical (or symbolic) function of $\boldsymbol{x} = \left[x_1, x_2, \ldots, x_{\dim(\boldsymbol{x})}\right]$. While the non-negativity is guaranteed, existence of moments needs to be verified when a test function for $\mathcal{G}(\boldsymbol{x})$ is considered. As our focus in this paper is on the maximum entropy distribution function given by Eq. 9, we consider $\mathcal{G}(\boldsymbol{x})$ to have the form

$$\mathcal{G}(\boldsymbol{x}) = \boldsymbol{\lambda} \cdot \boldsymbol{H}(\boldsymbol{x}) = \sum_{i=1}^{N_b} \lambda_i H_i(\boldsymbol{x}) \; . \tag{20}$$

Now we proceed to provide a modified formulation for SR tailored to our MED problem.

Definition 5.2. *Symbolic Regression for the Maximum Entropy Distribution (SR-MED) problem*

Given a measure of difference between distributions $\mathcal{L}$ (e.g. KL Divergence) and a dataset $\mathcal{D} = \{\boldsymbol{X}_i\}_{i=1}^N$ consisting of N i.i.d. samples, where $\boldsymbol{X}_i \in \mathbb{R}^{\dim(\boldsymbol{x})}$, the SR-MED problem searches in the space $\mathcal{S}^{N_b}$ for N_b basis functions subject to $\hat{f} \in \mathcal{S}_{f|\boldsymbol{R}}$ which minimizes $\mathcal{L}$.

Here, we deploy continuous functions consisting of *binary* operators (e.g. +, −, ×, ÷) or *unary* functions (e.g. cos, sin, exp, log) to fill the space S^{N_b}. As in most of the SR methods, we encode mathematical expressions using symbolic expression trees, a type of binary tree, where internal nodes contain operators or functions and terminal nodes (or leaves) contain input variables of constants. For instance, the expression tree in Figure 1 represents $x^2 \cos(x)$. In this paper, we perform a Monte Carlo symbolic search in the space of smooth functions (by generating random expression trees) to find a vector of basis functions $\boldsymbol{H}$ that guarantees acceptable $\mathrm{cond}(\boldsymbol{L}^{\mathrm{ME}})$, by rejecting candidates that do not satisfy this condition. In our search, we do not consider test basis functions with odd growth rates which lead to non-realizable distributions.

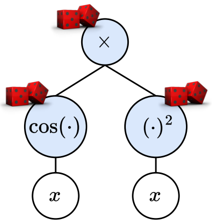


Figure 1: Expression tree for $x^2 \times \cos(x)$.

6 Multi-level density recovery

We further improve our proposed method by introducing a multi-level process that improves our prediction as the distribution becomes more detailed. The goal is to obtain a more generalized MED estimate with the form

$$\hat{f}(\boldsymbol{x}) = \sum_{l=1}^{N_L} m^{[l]}\, \hat{f}^{[l]}(\boldsymbol{x}) \tag{21}$$

$$\text{where} \quad \hat{f}^{[l]}(\boldsymbol{x}) = \frac{1}{Z^{[l]}} \exp\Big(\boldsymbol{\lambda}^{[l]} \cdot \boldsymbol{H}^{[l]}(\boldsymbol{x})\Big) \ , \tag{22}$$

$(.)^{[l]}$ denotes the level index, $Z^{[l]}$ is the normalization factor of density at level l, N_L is the number of levels considered and $m^{[l]}$ indicates the portion of total mass that is covered by $\hat{f}^{[l]}$. We note that this multi-level approach is recursive and can be described as follows:

- **Step 1: Find MED estimate $\hat{f}^{[l]}$ at level l:** At level l, first we pick a basis function $\boldsymbol{H}^{[l]}$ by solving the SR-MED problem detailed in def. 5.2. Then, we orthonormalize the basis function with respect to the distribution of the samples using Gram–Schmidt's procedure as outlined in Algorithm 1.
- **Step 2: Removing subset of samples covered by $\hat{f}^{[l]}$:** Here, we attempt to find and remove a subset of samples $\mathcal{D}^{[l]}_{\text{mask}}$ – representing a fraction of the *mass*, i.e. $m^{[l]} = |\mathcal{D}^{[l]}_{\text{mask}}|/|\mathcal{D}|$ – that can be estimated by our estimated $\hat{f}^{[l]}$ at this level. To this end, we deploy acceptance/rejection with probability $\hat{f}^{[l]}/\hat{f}^{\text{hist}}$ to find and remove $\mathcal{D}^{[l]}_{\text{mask}}$ from the remaining samples $\mathcal{D}^{[l]}$.
- **Step 3: Repeat steps 1-2 for the next level $l+1$ until almost no samples are left:** Repeat steps 1-2 with the remaining uncovered samples (which constitutes the next level) until there are (almost) no uncovered samples. The resulting total distribution is a weighted sum of the estimates from each level.

In Algorithm 2, we detail a pseudocode for our devised multi-level process. As we will see in the next section, our proposed multi-level recursive mechanism improves overall performance, and elegantly describe details of multi-mode distributions.

Algorithm 2: Multi-level, symbolic and recursive algorithm for density recovery. Here, $\mathcal{D}^{[l]}$ denotes the set of samples at level l and $\boldsymbol{u}$ is a random variable that is uniformly distributed in $(0, 1)$, i.e. $\boldsymbol{u} \sim \mathcal{U}([0, 1])$.

```
Input: $\mathcal{D}^{[1]} = \mathcal{D} = \{\boldsymbol{X}_i\}_{i=1}^{N}$, $N_L^{\text{tot}} = N_L$
for $l = 1, ..., N_L$ do
    Sample random basis functions $\boldsymbol{H}^{[l]}$ that satisfies def. 5.2 starting from polynomials in level $l = 1$;
    Compute $\hat{f}^{[l]}(\boldsymbol{x})$ given $\mathcal{D}^{[l]}$ using Algorithm 1;
    $\mathcal{D}^{[l]}_{\text{mask}} \leftarrow \{\mathcal{D}^{[l]} \mid \hat{f}^{[l]}(\boldsymbol{X})/\hat{f}^{\text{hist}}(\boldsymbol{X}) > \boldsymbol{u}\}$ where $\boldsymbol{u} \sim \mathcal{U}([0, 1])$;
    $m^{[l]} \leftarrow |\mathcal{D}^{[l]}_{\text{mask}}|/|\mathcal{D}|$;
    if $\sum_{j=1}^{l} |\mathcal{D}^{[j]}_{\text{mask}}| \approx |\mathcal{D}|$ then
        $\mathcal{D}^{[l]}_{\text{mask}} \leftarrow \mathcal{D}^{[l]}$;        // Mask all available samples
        $m^{[l]} \leftarrow |\mathcal{D}^{[l]}_{\text{mask}}|/|\mathcal{D}|$;
        $N_L^{\text{tot}} \leftarrow l$;
        break;        // Terminate the process
    else
        $\mathcal{D}^{[l+1]} \leftarrow \mathcal{D}^{[l]} \backslash \mathcal{D}^{[l]}_{\text{mask}}$;        // The uncovered samples are left for the next level
    end
end
Return $\hat{f}(\boldsymbol{x}) = \sum_{l=1}^{N_L^{\text{tot}}} \hat{f}^{[l]}(\boldsymbol{x})\ |\mathcal{D}^{[l]}_{\text{mask}}|/|D|$;        // $N_L^{\text{tot}}$ is the total number of recursive calls
```

7 Algorithm for MESSY estimation

The complete MESSY estimation algorithm is summarized in algorithm 3. Within the iteration loop, following each application of the multi-level, symbolic, and recursive density recovery summarized in algorithm 2, we introduce a maximum-cross entropy distribution (MxED) correction step (see Appendix B for details) to reduce any bias in our prediction for $\hat{f}$ from the former.

Finally, after completing the desired number of iterations, the algorithm returns the candidate density with the smallest KL Divergence given by

$$\mathrm{KL}\big(f\,||\,\hat{f}\big) = \int f(\boldsymbol{x}) \log\left(\frac{f(\boldsymbol{x})}{\hat{f}(\boldsymbol{x})}\right) d\boldsymbol{x} \tag{23}$$

$$= -\int f(\boldsymbol{x}) \log\big(\hat{f}(\boldsymbol{x})\big) d\boldsymbol{x} + \int f(\boldsymbol{x}) \log\big(f(\boldsymbol{x})\big) d\boldsymbol{x} \tag{24}$$

$$\approx -\big\langle \log\big(\hat{f}(\boldsymbol{X})\big)\big\rangle + \underbrace{\int f(\boldsymbol{x}) \log\big(f(\boldsymbol{x})\big) d\boldsymbol{x}}_{\text{constant with respect to } \hat{f}} . \tag{25}$$

In other words, we use $-\big\langle \log\big(\hat{f}(\boldsymbol{X})\big)\big\rangle$ as our selection criterion.

Algorithm 3: Pseudocode of the proposed `MESSY` estimation method. Here, $\boldsymbol{R}$ is the vector of linearly independent (polynomial) basis functions used in the moment matching procedure of MxED. Here, for MESSY-S the number of basis functions N_b is sampled uniformly from the sample space Ω_{N_b}, e.g. here we use $\Omega_{N_b} = \{2, ..., 8\}$ unless mentioned otherwise.

```
Input: $\mathcal{D} = \{\boldsymbol{X}_i\}_{i=1}^N$, $\Omega_{N_b}$, $N_m$, $N_{\text{iters}}$
Initialize $\hat{f}^{(i)} = 0$ for $i = 1, ..., N_{\text{iters}}$;
for $i = 1$ to $N_{\text{iters}}$ do
    if $i > 1$ then
        Sample $N_b \sim \mathcal{U}(\Omega_{N_b})$;
    end
    Find $\hat{f}$ using multi-level, symbolic and recursive Algorithm 2 for density recovery;
    Generate samples of $\boldsymbol{Y} \sim \hat{f}$;
    Apply boundary condition (bounded/unbounded) to $\hat{f}$;
    Correct $\hat{f}$ using MxED (Algorithm 5) given samples $\boldsymbol{Y}$ as prior and $\mathbb{E}[\boldsymbol{R}(\boldsymbol{X})]$ as target moments;
    $\hat{f}^{(i)} \leftarrow \hat{f}$;
end
$\hat{f}^{\mathrm{MESSY-P}} = \hat{f}^{(1)}$;
$\hat{f}^{\mathrm{MESSY-S}} = \mathrm{argmin}_{\hat{f} \in \{\hat{f}^{(i)}\}_{i=1}^{N_{\text{iters}}}} \Big(\mathrm{KL}(f\,||\,\hat{f})\Big)$ ;
Return $\hat{f}^{\mathrm{MESSY-P}}$ and $\hat{f}^{\mathrm{MESSY-S}}$.
```

The MESSY algorithm comes in two flavors: MESSY-P, which considers only polynomial basis functions for $\boldsymbol{H}$, and MESSY-S which includes optimization over basis functions using the SR algorithms outlined above. In fact, by convention, the SR algorithm in MESSY-S starts its first iteration using polynomial basis functions up to order N_m as the sample space of smooth functions. In other words, MESSY-P is a special case of MESSY-S with $N_{\text{iter}} = 1$. In the remaining iterations of MESSY-S, we perform the symbolic search in the space of smooth functions of order N_m to find N_b bases that provide manageable cond($\boldsymbol{L}^{\mathrm{ME}}$), as discussed in Section 6.

In addition, we provide the option to enforce boundedness of $\hat{f}$ on the support that is specified by the user, i.e. letting $\hat{f}(\boldsymbol{x}) = 0$ for all $\boldsymbol{x}$ outside the domain of interest. This allows us to recover distributions with discontinuity at the boundary which may have application in image processing.

We also provide an option to further reduce the bias by minimizing the cross-entropy given samples of bounded/unbounded multi-level estimate as prior and moments of input samples as the target moments (see Appendix B for more details on the cross-entropy calculation). For this optional step, we generate samples of $\hat{f}$ and match the moments of polynomial basis functions up to order N_m. Since the solution at each level of $\hat{f}$ is close to the exact MED solution, the optimization problem associated with the moment matching procedure of the cross-entropy algorithm converges very quickly, i.e. in a few iterations, providing us with a correction that minimizes bias along with the weighted samples of our estimate as the by-product. We note that in general the order of the randomly created basis function during the MESSY-S procedure may be different from the one used in the cross-entropy moment matching procedure.

8 Results

In this section we demonstrate the effectiveness of the proposed MESSY estimation method in recovering distributions given samples, using a number of numerical experiments, involving a range of distributions ranging from multi-mode to discontinuous. For validation, we provide comparisons with the standard KDE using the Silverman rule for the bandwidth h (Silverman, 1986), i.e.,

$$h = \left(\frac{4\hat{\sigma}^5}{3N}\right)^{1/5} \tag{26}$$

where $\hat{\sigma}$ denotes the standard deviation computed from the samples and cross-entropy closure with Gaussian as the prior (MxED) using Newton's method. We note that while the standard maximum entropy distribution function differs from MxED as the latter incorporates a prior, we intentionally use MxED as a benchmark instead because the standard approach can be extremely expensive.

Unless mentioned otherwise, we report error, time, and KL Divergence by ensemble averaging over 25 for different sets of samples. Furthermore, in the case of MESSY-S we perform $N_{\text{iters}} = 10$ iterations. Here we report the execution time using a single-core CPU for each method. Typical symbolic expressions of density functions recovered by MESSY for the test cases considered here can be found in Appendix C.

8.1 Bi-modal distribution function

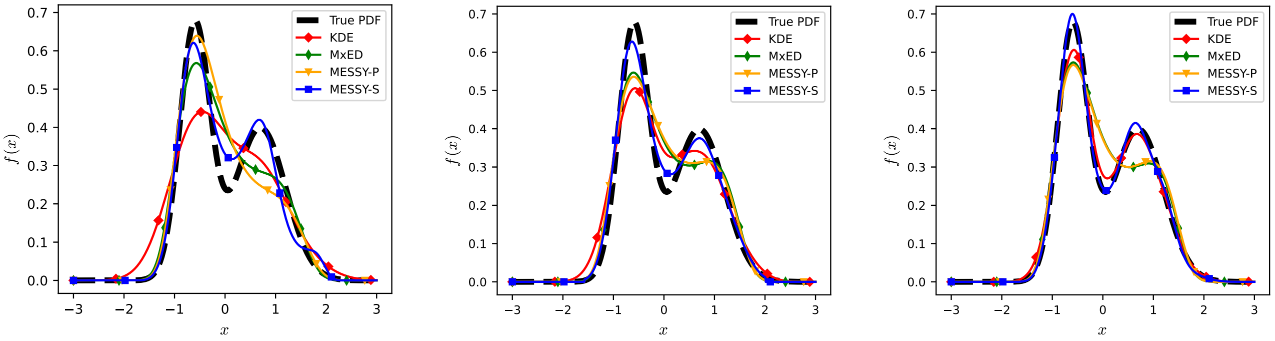
For our first test case, we consider a one-dimensional bi-modal distribution function constructed by mixing two Normal distribution functions $\mathcal{N}(x \mid \mu, \sigma)$, i.e.

$$f(x) = \alpha \mathcal{N}(x \mid \mu_1, \sigma_1) + (1-\alpha)\mathcal{N}(x \mid \mu_2, \sigma_2), \tag{27}$$

with $\alpha = 0.5$, means $\mu_1 = -0.6$ and $\mu_2 = 0.7$, and standard deviations $\sigma_1 = 0.3$ and $\sigma_2 = 0.5$.

Figure 2 compares results from MESSY, KDE and MxED for three different sample sizes, namely 100, 1000, and $10,000$ samples of f. For MxED and MESSY-P, we use $N_b = N_m = 4$. In the case of MESSY-S, we randomly create N_b basis functions which are $\mathcal{O}(x^4)$ (where N_b is sampled uniformly within $\{2, \ldots, 8\}$). Both MESSY results are subject to a cross-entropy correction step with $N_b = 4$ polynomial moments. Clearly, the MxED and MESSY methods provide a better estimate compared to KDE when a small number of samples is available; KDE suffers from bias introduced by the smoothing kernel.

In order to analyze the error further, Fig 3 presents the relative error in low and high order moments, KL Divergence, and single-core CPU time as the measure of computational cost for considered methods. The KDE error can only be reduced by increasing the number of samples. However, maximum entropy based estimators such as MxED and MESSY provide more robust estimate when less samples are available. We point out that the convergence of the cases where only moments of polynomial basis functions are considered, i.e. MxED and MESSY-P, relies on the degree of the polynomials and not the number of samples. On the other hand, the additional search associated with MESSY-S returns more appropriate basis functions for a given upper bound on the order of the basis functions.



(a) 100 samples (b) 1,000 samples (c) 10,000 samples

Figure 2: Density estimation using KDE, MxED, MESSY-P, and MESSY-S given (a) 100, (b) 1,000, and (c) 10,000 samples.

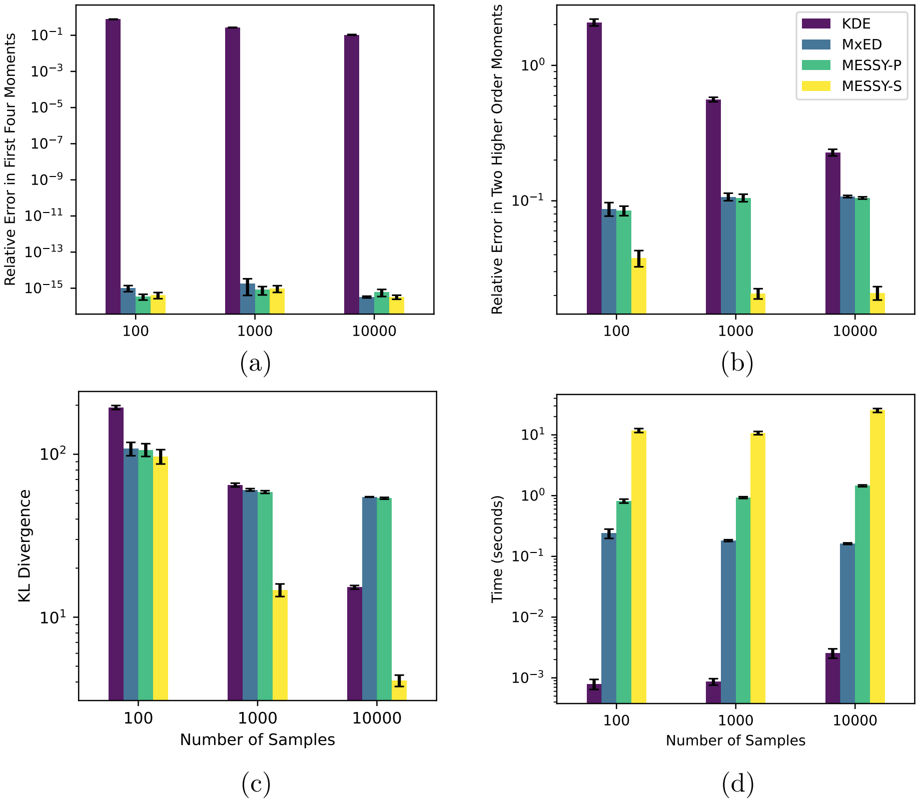
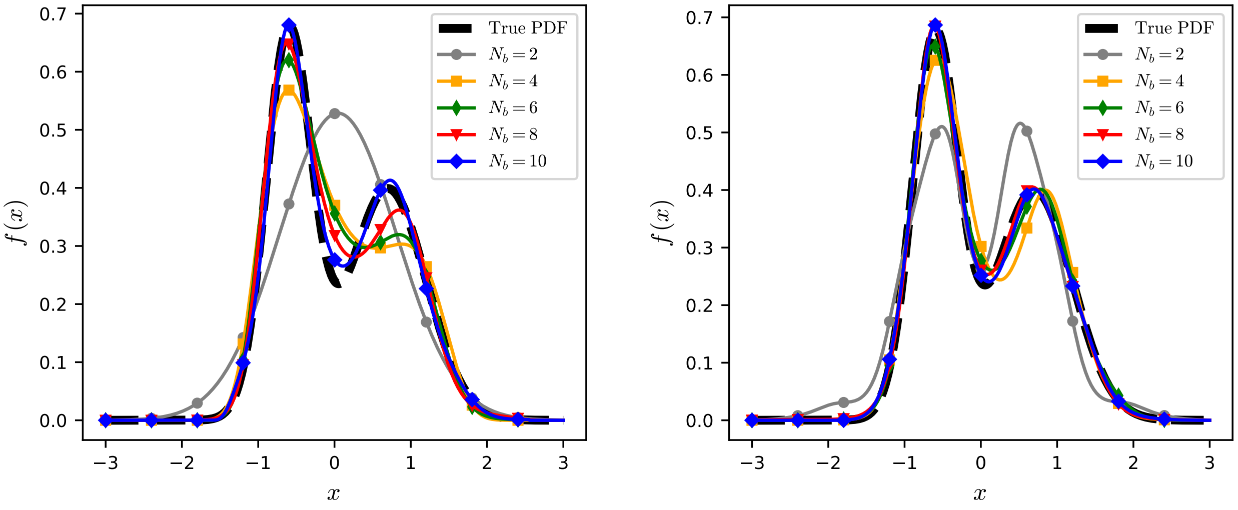


Figure 3: Comparing the relative error in (a) the first four moments, (b) two higher order moments (i.e. fifth and sixth moments), (c) KL Divergence, and (d) the execution time for KDE, MxED, MESSY-P, and MESSY-S in recovering distribution function for different sample sizes. Here, the error bar (in black) corresponds to the standard error of the empirical measurements.

Next, we perform a convergence study on 10,000 samples and show that the parametric description converges to the solution when its degrees of freedom are increased. In Fig. 4, we show that both MESSY-P and MESSY-S converge to the true solution by increasing either the order of polynomial basis function, or the number of basis functions, respectively. In the case of MESSY-S, we generated symbolic expressions that are $\mathcal{O}(x^2)$. The improved agreement compared to the MESSY-P case highlights the benefit derived from non-traditional basis functions that may better represent the data.

As shown in Fig. 5, the MESSY-S procedure results in better-conditioned $\boldsymbol{L}^{\text{ME}}$ matrices than the MESSY-P for the same degrees of freedom. However, the search for a *good* basis function increases the computational cost. In each iteration of the search for basis functions, the MESSY-S algorithm may reject symbolic basis candidates based on the condition number of the matrix $\boldsymbol{L}^{\text{ME}}$. In other words, the improved performance associated with MESSY-S comes at some increased computational cost.



(a) MESSY-P

(b) MESSY-S

Figure 4: Convergence of MESSY estimation to target distribution function by (a) increasing the order of polynomial basis functions for MESSY-P or (b) increasing the number of randomly selected symbolic basis functions with $N_m = 2$ for MESSY-S.

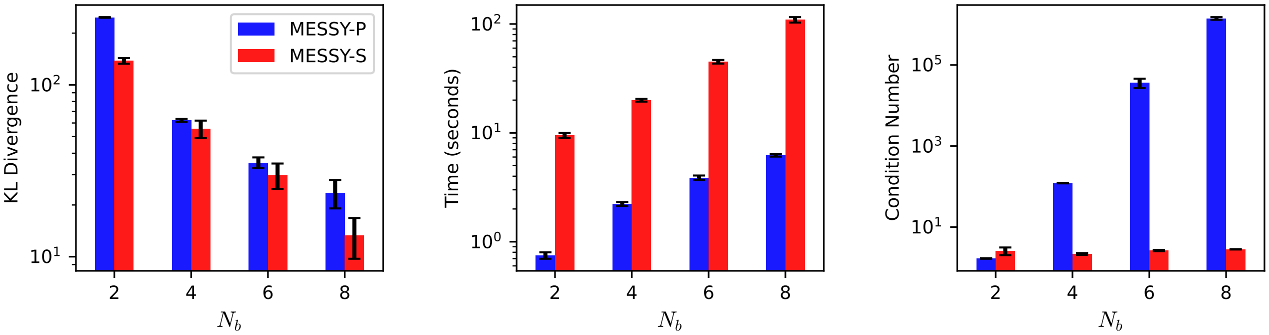


Figure 5: KL Divergence, execution time, and condition number against the degrees of freedom, i.e. the order of polynomial basis functions for MESSY-P or the number of symbolic basis functions with $N_m = 2$ for the MESSY-S estimate.

8.2 Limit of realizability

One of the challenging moment problems for maximum entropy methods is the one involving distributions near the border of physical realizability. In the one-dimensional case with moments of the first four monomials $[x, x^2, x^3, x^4]$ as the input, the moment problem is physically realizable when

$$\int x^4 f(x)dx \geq \left(\int x^3 f(x)dx\right)^2 + 1. \tag{28}$$

The moment problem with moments approaching the equality in Eq. 28 is called *limit of realizablity* (McDonald & Torrilhon, 2013; Akhiezer & Kemmer, 1965). We consider samples from a distribution in this limit as our test case here, since the standard MED cannot be solved due to an ill-conditioned Hessian (see Abramov (2007); Alldredge et al. (2014)).

In Fig. 6, we depict the estimated density of a bi-modal distribution in this limit given its samples with moments $\langle X \rangle = 0$, $\langle X^2 \rangle = 1$, $\langle X^3 \rangle = -2.10$ and $\langle X^4 \rangle = 5.42$. Here, we compare the density obtained using KDE, MxED, MESSY-P, and MESSY-S to the histogram of samples. In this example, we obtained the MESSY-S estimate by searching in the space of smooth functions with $N_b \in \{2, ..., 8\}$ basis functions and compare polynomial and symbolic basis functions of order 2 and 4.

In Fig. 7, we compare the KL Divergence, the execution time, and the condition number for each method. While KDE suffers from over-smoothing and MxED/MESSY-P require at least $N_b = 4$ (and consequently $N_m = 4$, resulting in a stiff problem with large condition number), MESSY-S can obtain accurate density estimates by using unconventional basis functions with $N_m = 2$, thus maintaining a manageable condition number.

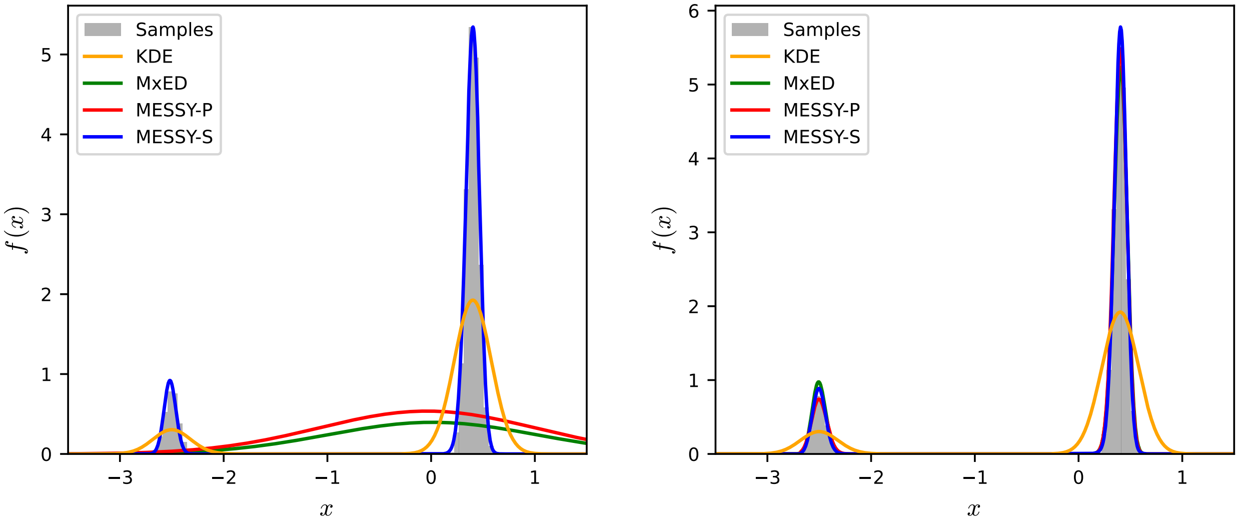


Figure 6: Estimating density for a case of distribution near the limit of realizability using KDE, MxED, MESSY-P, and MESSY-S. The solutions of MxED, MESSY-P, and MESSY-S are obtained using basis functions of second (left) and fourth (right) order.

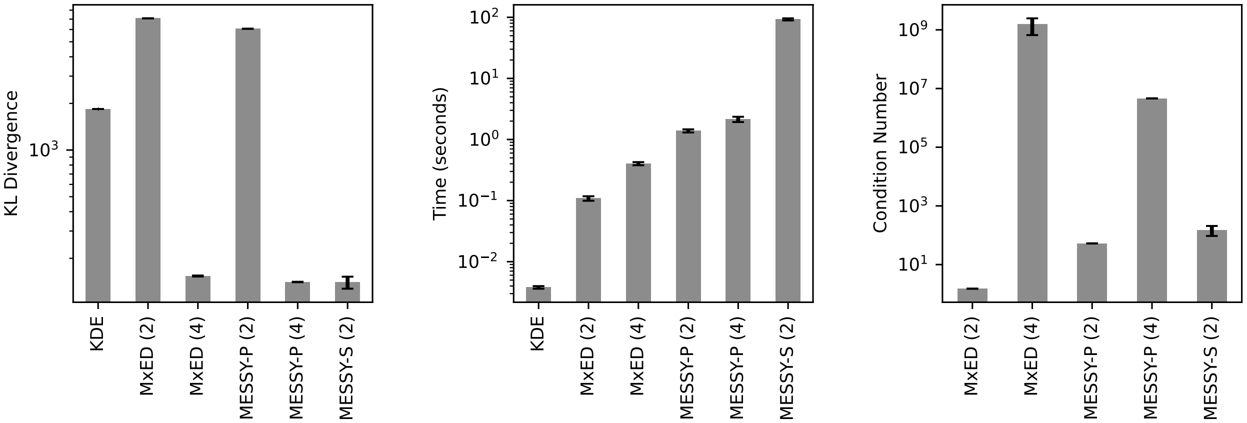


Figure 7: Comparing KL Divergence, execution time, and condition number of KDE, MxED, MESSY-P, and MESSY-S for an unknown distribution near the limit of the realizability. Here, we consider polynomial basis functions of second and fourth order for MxED and MESSY-P denoted by MxED (2), MxED (4), MESSY-P (2) and MESSY-P (4), respectively. In MESSY-S, we consider symbolic basis functions of second order only which we denote by MESSY-S (2).

8.3 Discontinuous distributions

We now highlight the benefits of using MESSY estimation with *piecewise continuous* capability for recovering distributions with a discontinuity at the boundary. As an example, let us consider the exponential distribution with a probability density function given by

$$f(x) = \begin{cases} ae^{-ax} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{29}$$

with $a = 1$.

Given $10,000$ samples of this distribution, in Fig. 8 we compare KDE, MxED, and the proposed MESSY-P and MESSY-S methodologies. In the case of MxED and MESSY-P we consider second-order polynomial basis functions, and for MESSY-S we search the space of smooth functions for $N_b \in \{2, ..., 8\}$ symbolic basis functions of order $\mathcal{O}(x^2)$. For MESSY-P and MESSY-S, we apply the boundary condition

$$\hat{f}(x) = 0 \quad \forall x < \min(X). \tag{30}$$

By providing information about the boundedness of the expected distribution, we enable MESSY to accurately predict densities with discontinuity near the boundary. As it can be seen clearly from Fig. 8, in contrast to KDE and MxED, both MESSY-P and MESSY-S provide accurate predictions by taking advantage of the information about the boundedness of the target density.

The KL Divergence score and execution time for each method is shown in Fig. 9. These figures show that MESSY-P and MESSY-S provide a more accurate description compared to the KDE estimate, albeit at a higher computational cost.

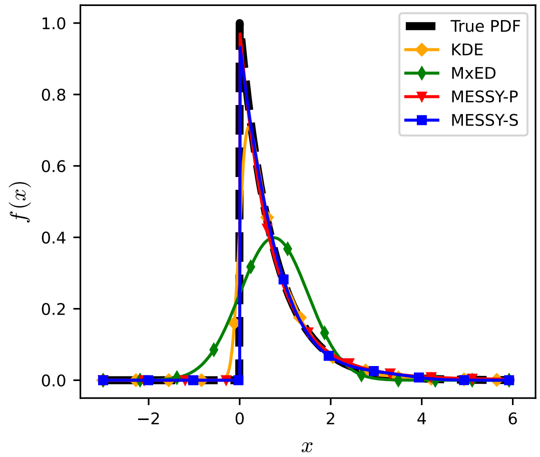


Figure 8: Estimating density of exponential distribution function from its samples using KDE, MxED, MESSY-P, and MESSY-S. For MxED, MESSY-P and MESSY-S, with $N_m = 2$.

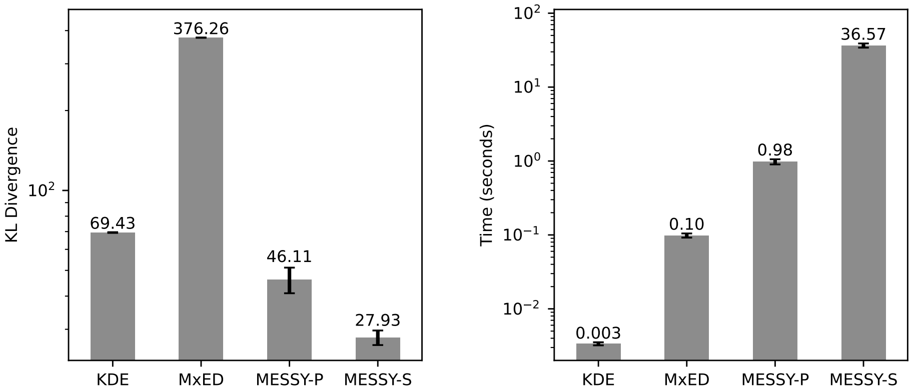


Figure 9: KL Divergence and execution time for KDE, MxED, MESSY-P, and MESSY-S estimation of exponential distribution function given 10,000 samples.

9 Conclusion and Outlook

We present a new method for symbolically recovering the underlying probability density function of a given finite number of its samples. The proposed method uses the maximum entropy distribution as an ansatz thus guaranteeing positivity and least bias. We devise a method for finding parameters of this ansatz by considering a Gradient flow in which the ansatz serves as the driving force. One main takeaway from this work is that the parameters of the MED ansatz can be computed efficiently by solving a linear problem involving moments calculated from the given samples.

The second main takeaway from this work is that accurate density recovery does not necessarily require the use of high-order moments. In fact, increasing the number of complex but low-order basis functions leads to superior expressiveness and better assimilation of the data. For this reason, the proposed method is equipped with a Monte Carlo search in the space of smooth functions for finding basis functions to describe the exponent of the MED ansatz, using KL Divergence, calculated from the unknown-distribution samples, as an optimality criterion. Discontinuous densities are treated by considering piece-wise continuous functions with support on the space covered by samples.

We validate and test the proposed MESSY estimation approach against benchmark non-parametric (KDE) and parametric (MxED) density estimation methods. In our experiments, we consider three canonical test cases; a bi-modal distribution, a distribution close to the limit of realizability, and a discontinuous distribution function. Our results suggest that MESSY estimation exhibits several positive attributes compared to existing methods. Specifically, while retaining some of the most desirable features associated with MED, namely non-negativity, least bias, and matching the moments of the unknown distribution, it outperforms standard maximum-entropy-based approaches for two reasons. First, it uses samples of the target distribution in the evaluation of the Hessian, which has a linear cost with respect to the dimension of the random variable. Second, the resulting linear problem for finding the Lagrange multipliers from moments is significantly more efficient than the Newton line search used by the classical MED approach. Moreover, our multi-level algorithm allows for recovery of more complex distributions compared to the standard MED approach. Combining the efficient approach of finding maximum entropy density via a linear system with the symbolic exploration for the optimal basis functions paves the way for achieving low bias, consistent, and expressive density recovery from samples.

Possible directions for future work include: (i) application of the proposed methodology to high-dimensional distribution functions, including applications to recovering governing dynamical laws from samples; and (ii) applications to variance reduction.

References

Rafail V Abramov. An improved algorithm for the multidimensional moment-constrained maximum entropy problem. *Journal of Computational Physics*, 226(1):621–644, 2007.

Rafail V Abramov. The multidimensional moment-constrained maximum entropy problem: A BFGS algorithm with constraint scaling. *Journal of Computational Physics*, 228(1):96–108, 2009.

Rafail V Abramov. The multidimensional maximum entropy moment problem: A review of numerical methods. *Communications in Mathematical Sciences*, 8(2):377–392, 2010.

Naum Il'ich Akhiezer and N Kemmer. *The classical moment problem: and some related questions in analysis*, volume 5. Oliver & Boyd Edinburgh, 1965.

Graham W Alldredge, Cory D Hauck, Dianne P O'Leary, and André L Tits. Adaptive change of basis in entropy-based moment closures for linear kinetic equations. *Journal of Computational Physics*, 258: 489–508, 2014.

Luca Biggio, Tommaso Bendinelli, Alexander Neitz, Aurelien Lucchi, and Giambattista Parascandolo. Neural symbolic regression that scales. In *International Conference on Machine Learning*, 2021.

Zdravko I Botev, Joseph F Grotowski, and Dirk P Kroese. Kernel density estimation via diffusion. *The Annals of Statistics*, 38(5), 2010.

ZI Botev. Nonparametric density estimation via diffusion mixing. 2007.

Bogdan Burlacu, Gabriel Kronberger, and Michael Kommenda. Operon C++: An efficient genetic programming framework for symbolic regression. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, GECCO '20, pp. 1562–1570, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371278. doi: 10.1145/3377929.3398099.

Zhenning Cai, Yuwei Fan, and Ruo Li. A framework on moment model reduction for kinetic equation. *SIAM Journal on Applied Mathematics*, 75(5):2001–2023, 2015.

Sinho Chewi, Thibaut Le Gouic, Chen Lu, Tyler Maunu, and Philippe Rigollet. SVGD as a kernelized Wasserstein gradient flow of the chi-squared divergence. *Advances in Neural Information Processing Systems*, 33:2098–2109, 2020.

Edwin KP Chong and Stanislaw H Zak. *An introduction to optimization*, volume 75. John Wiley & Sons, 2013.

Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21 (1):5–30, 2006.

Ronald R Coifman, Stephane Lafon, Ann B Lee, Mauro Maggioni, Boaz Nadler, Frederick Warner, and Steven W Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the national academy of sciences*, 102(21):7426–7431, 2005.

Miles Cranmer, Alvaro Sanchez-Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. *arXiv preprint arXiv:2006.11287*, 2020.

Kristian Debrabant, Giovanni Samaey, and Przemysław Zielinski. A micro-macro acceleration method for the Monte Carlo simulation of stochastic differential equations. *SIAM Journal on Numerical Analysis*, 55 (6):2745–2786, 2017.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *arXiv preprint arXiv:1605.08803*, 2016.

David L Donoho, Iain M Johnstone, Gérard Kerkyacharian, and Dominique Picard. Density estimation by wavelet thresholding. *The Annals of statistics*, pp. 508–539, 1996.

Wolfgang Dreyer. Maximisation of the entropy in non-equilibrium. *Journal of Physics A: Mathematical and General*, 20(18):6505, 1987.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

Anthony William Fairbank Edwards. *Likelihood*. CUP Archive, 1984.

Aldo Frezzotti, Livio Gibelli, and Silvia Lorenzani. Mean field kinetic theory description of evaporation of a fluid into vacuum. *Physics of Fluids*, 17(1):012102, 2005. doi: 10.1063/1.1824111.

Alfredo Garbuno-Inigo, Franca Hoffmann, Wuchen Li, and Andrew M Stuart. Interacting Langevin diffusions: Gradient structure and ensemble Kalman sampler. *SIAM Journal on Applied Dynamical Systems*, 19(1): 412–441, 2020a.

Alfredo Garbuno-Inigo, Nikolas Nusken, and Sebastian Reich. Affine invariant interacting Langevin dynamics for Bayesian inference. *SIAM Journal on Applied Dynamical Systems*, 19(3):1633–1658, 2020b.

Luc Giraud, Julien Langou, and Miroslav Rozloznık. On the round-off error analysis of the gram-schmidt algorithm with reorthogonalization. Technical report, Technical Report TR/PA/02/33, CERFACS, Toulouse, France, 2002.

Harold Grad. Note on N-dimensional Hermite polynomials. *Communications on Pure and Applied Mathematics*, 2(4):325–330, 1949.

Cory D Hauck, C David Levermore, and André L Tits. Convex duality and entropy-based moment closures: Characterizing degenerate densities. *SIAM Journal on Control and Optimization*, 47(4):1977–2015, 2008.

M Hermite. *Sur un nouveau développement en série des fonctions*. Imprimerie de Gauthier-Villars, 1864.

Ying Jin, Weilin Fu, Jian Kang, Jiadong Guo, and Jian Guo. Bayesian symbolic regression. *arXiv preprint arXiv:1910.08892*, 2019.

M Chris Jones, James S Marron, and Simon J Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American statistical association*, 91(433):401–407, 1996.

Pierre-Alexandre Kamienny, Stéphane d'Ascoli, Guillaume Lample, and François Charton. End-to-end symbolic regression with transformers. *arXiv preprint arXiv:2204.10532*, 2022.

Jagat Narain Kapur. *Maximum-entropy models in science and engineering*. John Wiley & Sons, 1989.

A Ya Khinchin. *Mathematical foundations of information theory.* Courier Corporation, 2013.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

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

Michael Kommenda, Bogdan Burlacu, Gabriel Kronberger, and Michael Affenzeller. Parameter identification for symbolic regression using nonlinear least squares. *Genetic Programming and Evolvable Machines*, 21 (3):471–501, 2020.

Misaki Kon, Kazumichi Kobayashi, and Masao Watanabe. Method of determining kinetic boundary conditions in net evaporation/condensation. *Physics of Fluids*, 26(7):072003, 2014.

John R Koza and John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabrício Olivetti de França, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason H Moore. Contemporary symbolic regression methods and their relative performance. *arXiv preprint arXiv:2107.14351*, 2021.

Peter M Lee. *Bayesian statistics.* Arnold Publication, 1997.

Thomas Leonard and John SJ Hsu. *Bayesian methods: an analysis for statisticians and interdisciplinary researchers*, volume 5. Cambridge University Press, 2001.

C David Levermore. Moment closure hierarchies for kinetic theories. *Journal of statistical Physics*, 83(5-6): 1021–1065, 1996.

Fengyi Li and Youssef Marzouk. Diffusion map particle systems for generative modeling. *arXiv preprint arXiv:2304.00200*, 2023.

Qiang Liu. Stein variational gradient descent as gradient flow. *Advances in neural information processing systems*, 30, 2017.

James McDonald and Manuel Torrilhon. Affordable robust moment closures for CFD based on the maximum-entropy hierarchy. *Journal of Computational Physics*, 251:500–523, 2013.

T Nathan Mundhenk, Mikel Landajuela, Ruben Glatt, Claudio P Santiago, Daniel M Faissol, and Brenden K Petersen. Symbolic regression via neural-guided genetic programming population seeding. *arXiv preprint arXiv:2111.00053*, 2021.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

Patryk Orzechowski, William La Cava, and Jason H Moore. Where are we now? a large benchmark study of recent symbolic regression methods. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1183–1190, 2018.

Claudia Patrignani, K Agashe, G Aielli, C Amsler, M Antonelli, DM Asner, H Baer, Sw Banerjee, RM Barnett, T Basaglia, et al. Review of particle physics. *CHINESE PHYSICS C*, 2016.

Yudi Pawitan. *In all likelihood: statistical modelling and inference using likelihood.* Oxford University Press, 2001.

Brenden K Petersen, Mikel Landajuela Larma, Terrell N. Mundhenk, Claudio Prata Santiago, Soo Kyung Kim, and Joanne Taery Kim. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. In *International Conference on Learning Representations*, 2021.

Eckhard Platen and Nicola Bruti-Liberati. *Numerical solution of stochastic differential equations with jumps in finance*, volume 64. Springer Science & Business Media, 2010.

William A Porteous, M Paul Laiu, and Cory D Hauck. Data-driven, structure-preserving approximations to entropy-based moment closures for kinetic equations. *arXiv preprint arXiv:2106.08973*, 2021.

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

Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The annals of mathematical statistics*, pp. 832–837, 1956.

Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science advances*, 3(4):e1602614, 2017.

Mohsen Sadr, Manuel Torrilhon, and M Hossein Gorji. Gaussian process regression for maximum entropy distribution. *Journal of Computational Physics*, 418:109644, 2020.

Mohsen Sadr, Qian Wang, and M Hossein Gorji. Coupling kinetic and continuum using data-driven maximum entropy distribution. *Journal of Computational Physics*, 444:110542, 2021.

Subham Sahoo, Christoph Lampert, and Georg Martius. Learning equations for extrapolation and control. In *International Conference on Machine Learning*, pp. 4442–4450. PMLR, 2018.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324 (5923):81–85, 2009.

Steffen Schotthöfer, Tianbai Xiao, Martin Frank, and Cory D Hauck. Neural network-based, structure-preserving entropy closures for the Boltzmann moment system. *arXiv preprint arXiv:2201.10364*, 2022.

Simon J Sheather. Density estimation. *Statistical science*, pp. 588–597, 2004.

Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Aldo Tagliani. Hausdorff moment problem and maximum entropy: a unified approach. *Applied Mathematics and Computation*, 105(2-3):291–305, 1999.

Tony Tohme, Kevin Vanslette, and Kamal Youcef-Toumi. A generalized Bayesian approach to model calibration. *Reliability Engineering & System Safety*, 204:107141, 2020.

Tony Tohme, Dehong Liu, and Kamal Youcef-Toumi. GSR: A generalized symbolic regression approach. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=lheUXtDNvP`.

Belinda Tzen and Maxim Raginsky. Neural stochastic differential equations: Deep latent Gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*, 2019.

Silviu-Marian Udrescu and Max Tegmark. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

Silviu-Marian Udrescu, Andrew Tan, Jiahai Feng, Orisvaldo Neto, Tailin Wu, and Max Tegmark. AI Feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. *Advances in Neural Information Processing Systems*, 33:4860–4871, 2020.

Mojtaba Valipour, Bowen You, Maysum Panju, and Ali Ghodsi. Symbolicgpt: A generative transformer model for symbolic regression. *arXiv preprint arXiv:2106.14131*, 2021.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

Marco Virgolin and Solon P Pissis. Symbolic regression is NP-hard. *arXiv preprint arXiv:2207.01018*, 2022.

Sven Wang and Youssef Marzouk. On minimax density estimation via measure transport. *arXiv preprint arXiv:2207.10231*, 2022.

CJ Wild and GAF Seber. *Nonlinear regression.* New York: Wiley, 1989.

Hengzhe Zhang, Aimin Zhou, Hong Qian, and Hu Zhang. PS-Tree: A piecewise symbolic regression tree. *Swarm and Evolutionary Computation*, 71:101061, 2022.

A Maximum entropy distribution function

The maximum entropy distribution (MED) function finds the least-biased closure for the moment problem. Given N_b realizable moments $\boldsymbol{\mu} \in \mathbb{R}^{N_b}$ associated with polynomial basis functions $\boldsymbol{H}$ of the unknown distribution function, MED is obtained by minimizing the Shannon entropy with constraint moments using the method of Lagrange multipliers as

$$C[\mathcal{F}(\boldsymbol{x})] := \int \mathcal{F}(\boldsymbol{x}) \log\big(\mathcal{F}(\boldsymbol{x})\big) d\boldsymbol{x} + \sum_{i=1}^{N_b} \lambda_i \left(\int H_i(\boldsymbol{x}) \mathcal{F}(\boldsymbol{x}) d\boldsymbol{x} - \mu_i(\boldsymbol{x}) \right) . \tag{A.1}$$

By taking the variational derivative of functional A.1, the extremum is found as

$$\mathcal{F}(\boldsymbol{x}) = \frac{1}{Z} \exp\left(\sum_{i=1}^{N_b} \lambda_i H_i(\boldsymbol{x}) \right), \qquad \text{where} \;\; Z = \int \exp\left(\sum_{i=1}^{N_b} \lambda_i H_i(\boldsymbol{x}) \right) d\boldsymbol{x}, \tag{A.2}$$

which is referred to as the maximum entropy distribution function. The Lagrange multipliers $\boldsymbol{\lambda}$ appearing in Eq. A.2 can be found using the Newton-Raphson approach. As formulated in (Debrabant et al., 2017), the unconstrained dual formulation $D(\boldsymbol{\lambda})$ provides us with the gradient $\boldsymbol{g} = \nabla D(\boldsymbol{\lambda})$ and Hessian $\boldsymbol{L}(\boldsymbol{\lambda}) = \nabla^2 D(\boldsymbol{\lambda})$ as

$$g_i = \mu_i - \frac{1}{Z} \int H_i \exp\left(\sum_{k=1}^{N_b} \lambda_k H_k \right) d\boldsymbol{x} \quad \text{for } i = 1, ..., N_b \tag{A.3}$$

$$\text{and} \quad L_{i,j} = -\frac{1}{Z} \int H_i H_j \exp\left(\sum_{k=1}^{N_b} \lambda_k H_k \right) d\boldsymbol{x} \quad \text{for } i, j = 1, ..., N_b \; . \tag{A.4}$$

Once the gradient and Hessian are computed, the Lagrange multipliers $\boldsymbol{\lambda}$ can be updated via

$$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \boldsymbol{L}^{-1}(\boldsymbol{\lambda}) \boldsymbol{g}(\boldsymbol{\lambda}) \tag{A.5}$$

as detailed in Algorithm 4.

Algorithm 4: Newton's method for finding Lagrange multipliers of MED given moments $\boldsymbol{\mu}$ for a given tolerance ϵ.

```
Input: $\boldsymbol{\mu}$, $\boldsymbol{\lambda}_0$
Initialize $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda}_0$;
Compute gradient $\boldsymbol{g}$ and Hessian $\boldsymbol{L}$, i.e. Eq. A.3-A.4;
while $||\boldsymbol{g}|| > \epsilon$ do
    Update $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \boldsymbol{L}^{-1}\boldsymbol{g}$;
    Update gradient $\boldsymbol{g}$ and Hessian $\boldsymbol{L}$ with the new $\boldsymbol{\lambda}$ via numerical integration of Eq. A.3-A.4;
end
Return $\boldsymbol{\lambda}$;
```

B Maximum cross-entropy distribution function

The maximum cross-entropy distribution function (MxED) finds the least-biased closure for the moment problem given N_b realizable moments $\boldsymbol{\mu} \in \mathbb{R}^{N_b}$ associated with polynomial basis functions $\boldsymbol{H}$ of the unknown distribution function along with the prior $\mathcal{F}^{\text{Prior}}$ as the input. In other words, in addition to the moments of the target distribution, in the MxED method we also have access to a prior distribution $\mathcal{F}^{\text{Prior}}$ as well as N samples of the former, i.e. $\{\boldsymbol{X}_j^{\text{prior}}\}_{j=1}^N \sim \mathcal{F}^{\text{Prior}}$. MxED is obtained by minimizing the Shannon cross-entropy from the prior with constraint on moments using the method of Lagrange multipliers via the functional

$$C[\mathcal{F}(\boldsymbol{x})] := \int \mathcal{F}(\boldsymbol{x}) \log\left(\frac{\mathcal{F}(\boldsymbol{x})}{\mathcal{F}^{\text{Prior}}(\boldsymbol{x})}\right) d\boldsymbol{x} + \sum_{i=1}^{N_b} \lambda_i \left(\int H_i(\boldsymbol{x})\mathcal{F}(\boldsymbol{x})d\boldsymbol{x} - \mu_i(\boldsymbol{x})\right) . \tag{B.1}$$

By taking the variational derivative of functional B.1, the extremum is found to be

$$\mathcal{F}(\boldsymbol{x}) = \frac{1}{Z}\mathcal{F}^{\text{Prior}}(\boldsymbol{x}) \exp\left(\sum_{i=1}^{N_b} \lambda_i H_i(\boldsymbol{x})\right), \qquad \text{where } Z = \int \mathcal{F}^{\text{Prior}}(\boldsymbol{x}) \exp\left(\sum_{i=1}^{N_b} \lambda_i H_i(\boldsymbol{x})\right) d\boldsymbol{x}. \tag{B.2}$$

Similar to the maximum entropy distribution function, the Lagrange multipliers $\boldsymbol{\lambda}$ appearing in Eq. B.2 can be found by following the Newton-Raphson approach. As formulated in (Debrabant et al., 2017), the unconstrained dual formulation $D(\boldsymbol{\lambda})$ provides us with the gradient $\boldsymbol{g} = \nabla D(\boldsymbol{\lambda})$ and Hessian $\boldsymbol{L}(\boldsymbol{\lambda}) = \nabla^2 D(\boldsymbol{\lambda})$ as

$$g_i = \mu_i - \frac{1}{Z}\int \mathcal{F}^{\text{Prior}} H_i \exp\left(\sum_{k=1}^{N_b} \lambda_k H_k\right) d\boldsymbol{x} \quad \text{for } i = 1, ..., N_b \tag{B.3}$$

$$\text{and} \quad L_{i,j} = -\frac{1}{Z}\int \mathcal{F}^{\text{Prior}} H_i H_j \exp\left(\sum_{k=1}^{N_b} \lambda_k H_k\right) d\boldsymbol{x} \quad \text{for } i, j = 1, ..., N_b \ . \tag{B.4}$$

Once the gradient and Hessian are computed, the Lagrange multipliers $\boldsymbol{\lambda}$ can be updated via

$$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \boldsymbol{L}^{-1}(\boldsymbol{\lambda})\boldsymbol{g}(\boldsymbol{\lambda}) \ . \tag{B.5}$$

Since we have access to the samples of $\boldsymbol{X}^{\text{Prior}} \sim \mathcal{F}^{\text{Prior}}$, we use the given samples to compute the gradient and Hessian, i.e.

$$g_i \approx \mu_i - \big\langle H_i\big(\boldsymbol{X}^{\text{Prior}}\big) \big\rangle_W \quad \text{for } i = 1, ..., N_b \tag{B.6}$$

$$L_{i,j} \approx -\big\langle H_i\big(\boldsymbol{X}^{\text{Prior}}\big) H_j\big(\boldsymbol{X}^{\text{Prior}}\big) \big\rangle_W \quad \text{for } i, j = 1, ..., N_b \ , \tag{B.7}$$

where $W(\boldsymbol{X}^{\text{Prior}}) = \exp\left(\sum_{k=1}^{N_b} \lambda_k H_k\big(\boldsymbol{X}^{\text{Prior}}\big)\right)$ denotes weights for calculating moments using importance sampling, i.e. $\langle \phi(\boldsymbol{X}) \rangle_W := \sum_{j=1}^N \phi(\boldsymbol{X}_j) W(\boldsymbol{X}_j) / \sum_{j=1}^N W(\boldsymbol{X}_j)$. More details can be found below (Algorithm 5).

Algorithm 5: Newton's method for finding Lagrange multipliers of MxED given moments $\boldsymbol{\mu}$ and samples of prior $\boldsymbol{X}^{\text{Prior}} \sim \mathcal{F}^{\text{Prior}}$ for a given tolerance ϵ.

```
Input: $\boldsymbol{\mu}$, $\boldsymbol{X}^{\text{Prior}} \sim \mathcal{F}^{\text{Prior}}$
Initialize $\boldsymbol{\lambda} \leftarrow \boldsymbol{0}$;
Compute gradient $\boldsymbol{g}$ and Hessian $\boldsymbol{L}$, i.e. Eq. B.6-B.7;
while $||\boldsymbol{g}|| > \epsilon$ do
    Update $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \boldsymbol{L}^{-1}\boldsymbol{g}$;
    Update gradient $\boldsymbol{g}$ and Hessian $\boldsymbol{L}$ with the new $\boldsymbol{\lambda}$ using samples, i.e. Eq. B.6-B.7;
end
Return $\boldsymbol{\lambda}$;
```

C Solution found by MESSY for the considered test cases

Table 1: Density expressions recovered by our MESSY estimation method for several distributions.

Example	**Expression**	
Bimodal	MESSY-P	$\hat{f}(x) = 0.288e^{-0.017x^{10}+0.106x^9-0.084x^8-0.659x^7+1.209x^6+1.179x^5-3.722x^4+0.075x^3+2.693x^2-0.612x}$
	MESSY-S	$\hat{f}(x) = 0.993e^{-1.85x^2-1.162x\cos(1.5x)+0.232x-0.652\cos(x)-0.424\cos(2x)-0.591\cos(3.5x)+0.47\cos(\cos(3.5x))}$
Limit of Realizability	MESSY-P	$\hat{f}(x) = 1.591 \cdot 10^{-6}e^{-12.876x^4-56.46x^3-38.072x^2+62.617x} + 5.282 \cdot 10^{-27}e^{-7.969x^4-28.862x^3-4.342x^2+20.938x}$
	MESSY-S	$\hat{f}(x) = 4.134 \cdot 10^{81}e^{-21.893x^2\sin(x)+0.025x^2+117.267x\cos(x)+0.861x+395.584\sin^2(x)-57.393\sin(x)+200.421\cos(x)-744.874\cos(\cos(x))}$
Discontinuous	MESSY-P	$\hat{f}(x) = \begin{cases} 1.096\, e^{0.086x^2-1.298x} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$
	MESSY-S	$\hat{f}(x) = \begin{cases} 0.293\, e^{-0.145x^2+0.018x+0.251\cos(x)\cos(1.5x)+0.713\cos(x)+0.09\cos(1.5x)\cos(3x)+0.076\cos(3.5x)} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$