Published in final edited form as:

J Chem Phys. 2016 April 21; 144(15): 154101. doi:10.1063/1.4945380.

Bayesian Calibration of Coarse-Grained Forces: Efficiently Addressing Transferability

Paul N. Patrone^{a)}, Thomas W. Rosch, and Frederick R. Phelan Jr.

National Institute of Standards and Technology

Abstract

Generating and calibrating forces that are transferable across a range of state-points remains a challenging task in coarse-grained (CG) molecular dynamics (MD). In this work, we present a coarse-graining *workflow*, inspired by ideas from uncertainty quantification and numerical analysis, to address this problem. The key idea behind our approach is to introduce a Bayesian correction algorithm that uses functional derivatives of CG simulations to rapidly and inexpensively recalibrate initial estimates f_0 of forces anchored by standard methods such as Force-Matching (FM). Taking density-temperature relationships as a running example, we demonstrate that this algorithm, in concert with various interpolation schemes, can be used to efficiently compute physically reasonable force curves on a fine grid of state-points. Importantly, we show that our work flow is robust to several choices available to the modeler, including the interpolation schemes and tools used to construct f_0 . In a related vein, we also demonstrate that our approach can speed up coarse-graining by reducing the number of atomistic simulations needed as inputs to standard methods for generating CG forces.

I. INTRODUCTION

In recent years, the high-throughput capabilities of molecular dynamics (MD) simulations have attracted the interest of companies involved in materials research and development.^{1–3} In such settings, these tools are becoming an integral component of the decision-making process, given their ability to rapidly search design spaces for novel systems that warrant further experimental study. While it is generally agreed that this procedure can significantly reduce material development costs, a key theme has repeatedly emerged from practical implementations thereof: in order to be informative for commercial applications, computational models require techniques that can rapidly calibrate inputs and assess uncertainties in simulated predictions.^{2–4}

In the context of coarse-grained (CG) MD, this theme has proven to be especially germane. In particular, high-throughput applications generally require the ability to rapidly customize the level of coarse-graining for arbitrary systems across a range of state points (e.g. temperatures); *inexpensive and robust* calibration techniques are critical for enabling this degree of exibility. Often, the key input to be determined is the CG interparticle forcefield

^{a)} paul.patrone@nist.gov.

This work is a contribution of the National Institute of Standards and Technology and is not subject to copyright in the United States.

$f(r, q)$ as a function of separation r and state-point q , and many techniques have been developed to compute these functions from thermodynamic considerations of atomistic systems.^{5–13} However, such methods typically draw information from a limited number of state-points, and the resulting forces can yield incorrect predictions for even small changes away from the calibration conditions (see Fig. 1). This has led several authors to consider the following questions:^{14–25}

1. Is there a general and inexpensive method for calibrating $f(r, q)$ so that a CG simulation yields the same predictions as a corresponding atomistic simulation on a large interval of state-points?
2. Can uncertainty quantification (UQ) be used to speed up the CG process?

In this paper, we propose a workflow that addresses these questions through a combination of techniques inspired by Bayesian probability, uncertainty quantification, and numerical analysis. This main steps in this work flow amount to:

- I. building an analytical, target model $\mathbb{T}(q)$ to describe the predictions that the CG simulations should reproduce;
- II. postulating an initial or guess set of forces $f_0(r, q)$;
- III. iteratively correcting f_0 via a pointwise (in q) but inexpensive Bayesian algorithm that
 - A. estimates functional derivatives $\frac{\delta}{\delta f}[f_0, q]$ of the CG prediction $\mathbb{S}[f_0, q]$, and
 - B. uses these derivatives to correct f_0 by comparing CG simulations to the target.

Importantly, this work flow overcomes the difficulties of generating transferable forces via the low cost of Step III, which can quickly calibrate f_0 constructed by non-robust but cheap interpolations in state-space. Moreover, this approach is meant to complement existing methods for coarse-graining insofar as it can use their outputs to inform these initial guesses. In contrast to previous works, our procedure also yields a probability $P[f; q]$ expressing our belief that a given f is a physically plausible and well-calibrated input to the CG simulations.²⁶ By means of this P , we can compute an average (or most-plausible) estimate $\bar{f}(r, q)$ for which the difference $\mathbb{S}[\bar{f}(r, q), q] - \mathbb{T}(q)$ can be rendered statistically insignificant by iteratively applying Step III.

A central idea that facilitates our analysis is a spectral decomposition of the forces via

$$f(r, q) = \sum_{m=1}^M w_m(q) \phi_m(r) = \mathbf{w} \cdot \boldsymbol{\phi}, \quad (1)$$

where $\phi_m(r)$ are a set of orthogonal basis functions, $w_m(q)$ are q dependent weights, M is a cutoff selected according to the needs of the modeler, and bold letters denote vector

representations of the corresponding variables. Several notable works have addressed issues related to development and practical implementation of basis functions ϕ ; see, for example, Refs. 13, 27, and 28. Here we do not advocate for one choice or another, but rather emphasize that appropriately selecting ϕ allows one to represent f using a modest number $M \lesssim \mathcal{O}(10)$ of weights. This, in combination with the low-cost of running CG simulations, allows us to numerically approximate enough functional derivatives

$$\frac{\delta}{\delta f} \approx \frac{[w + \varepsilon \Delta w] - [w]}{\varepsilon}, \quad \varepsilon \ll 1 \quad (2)$$

(for a given direction Δw) to alter the weights in systematic and meaningful ways.

As we have suggested, though, a key difficulty repeatedly appears: it is generally too expensive to realize any simulated quantity at a dense set of state-points. The problem is often most pronounced when generating individual f_0 by CG methods such as Force Matching (FM), which can take thousands of CPU hours; similar problems arise when estimating functional derivatives for large M . As a result, a frequent and unavoidable task in our analysis is to interpolate weights and derivatives between so-called anchor points in the variable q . *Importantly, we desire that these interpolations yield an estimate of their uncertainties in order to avoid overstating our confidence in $\tilde{f}(r, q)$.* In what follows, we make extensive use of Gaussian-process regression (GPR). We refer the reader to Refs. 29–31 for excellent reviews of this subject.

The problem of *non-transferable* force fields has been considered by several authors; see, for example, Refs. 7, 9, 12, 14–18, 21–25, 32–36 and Sec. IV A. In many instances, these works have been successful in computing interparticle potentials that correctly predict material properties across a range of state points. To the best of our knowledge, however, these methods are limited in at least one of the following ways: they (i) cannot be calibrated to arbitrary reference data; (ii) are tailored for specific systems and/or material properties; (iii) lack a systematic correction algorithm to update force curves that yield incorrect predictions. Here our goal is to propose a generic, inexpensive, and *robust* calibration method whose usefulness does not strongly depend on either the system at hand or the details of coarse-graining. By making uncertainty quantification an integral part of the workflow, we also emphasize the *subjective* nature of coarse-graining, thereby uncovering decision-making elements of the problem. Within this framework, the CG methods advocated by the aforementioned references can be incorporated to inform such decisions in a systematic way.

We mention a few limitations of our analysis. For one, estimates of $\tilde{f}(r, q)$ only yield CG predictions that are as good as our knowledge of the target properties in the first place. Given that the latter are never known with infinite precision, there is inherent uncertainty in our assessment of what constitutes a calibrated force curve. Moreover, the Bayesian algorithm may not necessarily calibrate f against all properties of interest, even though the procedure is meant to be general in nature. This question is difficult to address insofar as its answer

depends on how smoothly the CG predictions change relative to w_m . We expand on these and further limitations in the following sections.

The rest of the paper is organized as follows. In Sec. II, we present the main ideas and key equations behind our calibration workflow. We begin by assuming an initial set of force curves and show how corrections are computed. In Sec. III, we consider stability of the correction algorithm to changes in the underlying inputs, e.g. the interpolation methods used in Sec. II. Section IV frames our analysis in the context of past works and more fully considers extensions and limitations of our approach. Throughout, we illustrate aspects of the analysis via a running example of calibrating density-temperature relations for CG polystyrene. Appendices A–D provide relevant information on simulation techniques, FM and the coarse-grained system, building target models, and interpolation procedures.

A. Notation and conventions

We review notation and other conventions used throughout this work.

- The capital letters N, M, Q, S denote positive integers.
- The letter q denotes a state-point such as temperature or pressure. In our running examples throughout, we will take q to be the temperature T .
- Non-bold letters are scalars *if no argument or a non-bold (scalar) argument is indicated*. Examples include q , w , and $w(q)$.
- Bold letters $\mathbf{q} = (q_1, \dots, q_Q)$ and $\mathbf{w} = (w_1, \dots, w_M) = (w_1(q), \dots, w_M(q)) = \mathbf{w}(q)$ denote vectors. In particular, $\mathbf{q}$ is a vector of state-points, while $\mathbf{w}(q) = (w_1(q), \dots, w_M(q))$ denotes the weights in Eq. (1) evaluated at a fixed q .
- In contrast to normal conventions, we allow a scalar function of a scalar argument to also take a vector argument. In such cases, this denotes the vector whose components are the scalar function evaluated at each element of the vector. That is, $\mathbf{w}(\mathbf{q}) = (w(q_1), \dots, w(q_Q))$ if $\mathbf{q} = (q_1, \dots, q_Q)$.
- We use $k(q, q')$ to denote a scalar function of two variables. If we replace the arguments with Q -dimensional vectors $\mathbf{q}$ and $\mathbf{q}'$, the object $k(\mathbf{q}, \mathbf{q}')$ is a $Q \times Q$ matrix

$$k(\mathbf{q}, \mathbf{q}') = \begin{bmatrix} k(q_1, q'_1) & k(q_1, q'_2) & \dots & k(q_1, q'_Q) \\ k(q_2, q'_1) & k(q_2, q'_2) & \dots & k(q_2, q'_Q) \\ \vdots & \vdots & \ddots & \vdots \\ k(q_Q, q'_1) & k(q_Q, q'_2) & \dots & k(q_Q, q'_Q) \end{bmatrix}. \quad (3)$$

Likewise, $k(q, \mathbf{q})$ and $k(\mathbf{q}, q)$ denote the corresponding row and column vectors formed by pairing q with the elements of $\mathbf{q}$.

- We indicate elements of a vector $\mathbf{v}$ or matrix $\mathbf{\Sigma}$ with the subscripted, non-bold versions v_i and Σ_{ij} . When order of operations is otherwise unclear, we use the notation $[\mathbf{v}]_j$ and $[\mathbf{\Sigma}]_{ij}$, see immediately below.

- The use of a Roman T as a superscript denotes the transpose, not the temperature as an exponent. That is, $[\Sigma^T]_{i,j} = \Sigma_{j,i}$ for matrix Σ .
- Unless otherwise noted, we distinguish the meaning of indices according to the parent symbol as follows.
 - q_k refers to the k th state-point in a collection thereof.
 - w_m refers to the m th mode in Eq. (1).
 - Importantly, w_n , f_n , P_n , and variations thereof refer to the n th updates of (or corrections to) the weights, forces, and probabilities.
 - The double-indexed symbol $w_{m,n}(q)$ denotes the n th correction of the m th mode evaluated at q .
 - Superscripts in braces, such as $\{i\}$ in $w^{\{i\}}$, denote the i th independent, identically distributed (i.i.d.) realization of a random quantity.
- We sometimes denote a probability over forces using $P[f; q]$. In this expression, P depends on q but is not normalized with respect to this variable.

II. CALIBRATION WORKFLOW

A. Goals, tasks, and the Bayesian framework: a conceptual overview

Stated imprecisely, our goal is to calibrate f on an interval of state-points by ensuring that it (a) is thermodynamically consistent with an atomistic model, and (b) yields the same predictions as some target $\mathbb{T}(q)$, e.g. experimental data or atomistic simulations. Given infinite computational resources, any of the techniques advocated by Refs. 5–9, 12, 15–18, 21–23, 32–34 determine functions f_0 that satisfy these criteria pointwise in q ; thus the problem is solved in theory.

In practice, however, several difficulties arise. First, coarse-graining methods generally rely on exhaustive sampling of configurations from fully atomistic simulations. This sampling is expensive, and often one can only realize such f_0 at a small set of anchor points q_k . Interpolation is sometimes used to overcome this limitation, but such techniques suffer from a second issue: CG simulations may be sensitive to minor changes in f . When so, reasonable interpolations of f_0 can yield incorrect predictions at intermediate q . A third problem complicates the picture further: coarse-graining computations are rarely converged, so that non-negligible uncertainties accompany estimates f_0 . Thus, as we show, even our anchor points may yield incorrect predictions, sometimes by a factor of two or more.

These observations suggest the need to augment a typical calibration procedure with systematic correction and uncertainty quantification (UQ) steps that address the limitations of interpolation. To achieve this, we begin by invoking a Bayesian perspective wherein we model our *belief* that a given f is the best choice to satisfy the criteria (a) and (b) above. Operationally, we ask: what is the probability $P[f; q]$ that f is well-calibrated at state-point q .

We caution that this probability should not be understood in the frequentist sense; i.e. there is no notion of a random draw from a set of optimal force curves. Rather, P quantifies belief in a way that allows for analytical and systematic treatment.³⁷ This is useful insofar as we can encode information about our initial estimates f_0 (computed, e.g. from coarse-graining or interpolation) and their uncertainties in a prior probability $P_0[f; q]$, which we update by comparing CG simulations with the target model. In distinctly Bayesian language, we compute a *posterior* $P_1[f; q]$ via a relation of the form

$$P_{n+1}[f; q] \propto P[(q)|f] P_n[f; q] = \mathcal{L}(f|\delta, q) P_n[f; q] \quad (4)$$

where $P[(q)|f] = \mathcal{L}(f|\delta, q)$ is the probability that f yields the target predictions $\mathbb{T}(q)$, or equivalently a likelihood function $\mathcal{L}$ expressing our belief that f is well calibrated, given that its CG predictions differ from the target by an amount $\delta(q)$. Importantly, this procedure can be iterated indefinitely by treating the posterior as a new prior and re-updating according to Eq. (4). See Fig. 2 for a flowchart of this correction loop. Within this loop, it is straightforward to extract an average $\bar{f}_n$ from P_n , which can be input into the CG simulations when a specific realization is needed. Henceforth, we refer to *any* initial estimate of forces entering the Bayesian correction loop as f_0 , irrespective of how it is computed.

In order to generate *transferable* forces, we propose to use this correction loop twice as follows (see also Secs. II B – II D). First, assume that $\mathbb{T}(q)$ has been constructed. Second, use some preferred CG method (such as FM) to estimate f_0 at a small set of anchor points q_k and correct these estimates according to the algorithm in Fig. 2. Third, interpolate the resulting force curves in state-space between the anchors, and again apply the correction algorithm to update forces pointwise. Note that the only significant difference between the correction loops amounts to how we compute an initial estimate f_0 ; see Fig. 3. We correct the anchor points first in order to render the interpolations more stable.

In Secs. II B–II D, we discuss in more detail the mathematics needed to use Eq. (4). Specifically, we outline procedure by which the ingredients P_0 , $\mathcal{L}$, and P_1 can be constructed for one iteration of the Bayesian correction and show results for the work flows in Figs. 2 and 3. Before continuing, however, a few comments are in order.

First, we can draw a close analogy between Eq. (4) and other well known optimization techniques. Specifically, in Sec. II C we will adopt the perspective that $\mathcal{L}$ is the probabilistic counterpart of a Newton's method update equation of the form “ $f_{n+1} - f_n = -\frac{[f_n]_q}{[f_n]}$,” where, informally, $[f_n]_q$ is a derivative with respect to f and the quotes emphasize that this equation is not expressed rigorously. As Newton's method solves $g(x_0 + x) = 0$ (for arbitrary g) by computing corrections x to an initial guess x_0 , similarly $\mathcal{L}$ weights changes f according to the likelihood that $\mathbb{S}[f_0 + f; q] - \mathbb{T}(q) = 0$. In perfect correspondence, P_0 is the probabilistic counterpart of an initial guess, and the product $P_1 \propto \mathcal{L} P_0$ ensures that the posterior P_1 is (loosely speaking) localized close the corresponding f_0 . In Sec. II D, we make these ideas more precise by showing that, for specific choices of priors and likelihoods, the mean $\bar{f}_1$ of

P_1 is the solution to a constrained optimization problem whose metric $\|f - f_0\|$ is weighted according to the uncertainty in f_0 .

Second, we assumed that an analytical or surrogate model $\mathbb{T}(q)$ is already known. While the construction of such models is non-trivial, many well-known techniques have been developed to accomplish similar tasks; moreover, the formulation of $\mathbb{T}(q)$ is often problem dependent. For these reasons, we leave the discussion of surrogate building for Appendix C. We note, however, that it is useful to express $\mathbb{T}$ in the form

$$\mathbb{T}(q) = \bar{\mathbb{T}}(q) + \eta_{\mathbb{T}}(q), \quad (5)$$

where $\bar{\mathbb{T}}$ is the average value of $\mathbb{T}$ and $\eta_{\mathbb{T}}(q)$ is a random process accounting for uncertainty through its variance $\sigma^2 = \text{Var}[\eta_{\mathbb{T}}(q)]$. Figure 4 provides an example of what Eq. (5) could look like for density-versus-temperature data. The yellow stripe indicates our level of uncertainty in the black curve, which is fit to atomistic simulation data ($\times$). For the purposes of calibrating f , any force curve that yields CG predictions within the yellow stripe is statistically consistent $\mathbb{T}$, and thus acceptable. This highlights the usefulness of adopting a probabilistic framework inherent in Eq. (4): we avoid overcorrecting our force curves to match predictions that may be unreliable or have large uncertainties.

Third, when the total number of modes M is finite, it is convenient to express $P[f; q] = P[\mathbf{w}; q]$ in terms of a vector $\mathbf{w}$ of weights. The equivalence of these expressions follows from Eq. (1), since a force curve can be reconstructed from only knowledge of $w_m(q)$. In general, it is desirable that M be as small as possible to reduce work associated with computing P . In the examples that follow, a Chebyshev polynomial basis set allows us to truncate Eq. (1) at values of $M \lesssim \mathcal{O}(10)$ when modeling quantities such as the density.

Fourth, we have noted that Eq. (4) operates pointwise in q ; that is, the Bayesian update is agnostic to whether P_0 is constructed from CG forces f_0 or interpolations. For this reason, we combine the discussions of interpolation and building priors in the next section. Gaussian processes, in particular, provide a natural means of accomplishing this task.

B. Building a prior

From a mathematical perspective, we find it useful to view a non-ideal (or finite-resource) CG algorithm as generating i.i.d. realizations from the statistical model

$$f_0(r, q) = \tilde{f}_0(r, q) + \eta_f(r, q), \quad (6)$$

where $\tilde{f}_0(r, q)$ is a deterministic force curve coming from an ideal (or infinite-resource) GC algorithm, and $\eta_f(r, q)$ is a zero-mean random process. In the context of FM used in our running examples, η_f accounts for the (sometimes neglected) fact that f_0 are constructed from atomistic simulations that are finite in both size and time. As a result, considerable

thermal noise propagates into the coarse-graining computations, so that realizations $f_0^{\{i\}}$ differ from one another; see the last paragraph of Sec. IV A (as well as Appendix B) for a discussion of why this uncertainty matters. We expect that related arguments justify Eq. (6) as a model for the output of other CG methods such as iterative Boltzmann inversion (IBI), etc..³⁸

Assuming orthogonality of the basis functions, we formally decompose $f_0(r, q)$ according to Eq. (1) to yield a set of random weights $w_0(q)$ whose average values $\mathbf{E}[w_0(q)] = \mathbf{w}_0(q)$ reconstruct $\tilde{f}_0(r, q)$. We also assume that for fixed q , the functional form of each $f_0^{\{i\}}(r, q)$ is given on the entire domain $0 \leq r \leq 1$, where r is a non-dimensionalized separation, and $r = 1$ is a cutoff distance beyond which particles do not interact;³⁹ see Appendix A for details on non-dimensionalization.

In order to generate P_0 for the first correction loop (cf. Fig. 3), we generate $N - 1$ realizations of $f_0^{\{i\}}$ at anchor points q_k in a vector $\mathbf{q} = (q_1, q_2, \dots, q_Q)$ of Q state-points. Given these, we construct the sample mean

$$\tilde{w}_0(q_k) = \frac{1}{N} \sum_{i=1}^N w_0^{\{i\}}(q_k) \quad (7)$$

(which is random!) and, when we can compute it, the sample covariance,

$$\text{Cov}[\tilde{w}_{m,0}(q_k), \tilde{w}_{m',0}(q_k)] \approx \frac{1}{N(N-1)} \sum_{i=1}^N \left[w_{m,0}^{\{i\}}(q_k) - \bar{w}_{m,0}(q_k) \right] \left[w_{m',0}^{\{i\}}(q_k) - \bar{w}_{m',0}(q_k) \right], \quad (8)$$

which quantifies correlations between modes and our uncertainty in $\mathbf{w}_0$ when approximating this quantity with $\tilde{\mathbf{w}}_0$. Given Eqs. (7) and (8), we then postulate that a reasonable prior at each anchor point is

$$P_0[\mathbf{w}; q_k] = \frac{1}{\sqrt{(2\pi)^M |\mathbf{C}|}} \exp \left[-\frac{[\mathbf{w} - \tilde{\mathbf{w}}_0(q_k)]^T \mathbf{C}^{-1} [\mathbf{w} - \tilde{\mathbf{w}}_0(q_k)]}{2} \right], \quad (9)$$

where $\mathbf{C} = \text{Cov}[\mathbf{w}_0(q_k), \mathbf{w}_0(q_k)]$ and $\mathbf{w}_0 = \tilde{\mathbf{w}}_0$ is defined for later convenience.⁴¹ Note that Eq. (9) is just the Gaussian probability density with a mean and covariance computed from Eqs. (7) and (8).

In order to generate P_0 for input into the second correction loop of Fig. 3, we must interpolate the weights in q . In general, we would like for P_0 to account for additional uncertainty that arises when q is far from any of the anchor points q_k . To achieve this, we use GPR, a common tool in the machine learning and uncertainty quantification communities;^{29–31} cf. App. D for a brief review of the key ideas. What we wish to emphasize is that GPR outputs multivariate Gaussian probabilities of the form

$$P_0[\mathbf{w};q] = \frac{1}{\sqrt{(2\pi)^M |\Sigma|}} \exp \left\{ \frac{-[\mathbf{w} - \bar{\mathbf{w}}_0(q)]^T \Sigma^{-1} [\mathbf{w} - \bar{\mathbf{w}}_0(q)]}{2} \right\} \quad (10)$$

where Σ is a covariance matrix, $|\Sigma|$ is the corresponding determinant, and the $\bar{\mathbf{w}}_0(q)$ are linear combinations of input weights (typically output by the first correction loop) at the anchor points; see also Eqs. (D5) and (D6) in Appendix D. In both Eqs. (9) and (10), the motivation for using Gaussian probabilities arises from the observation that realizations $f_0^{\{i\}}$ are themselves averages constructed from many i.i.d. samples of a thermodynamic ensemble. As such, it is reasonable to assume that a Central Limit Theorem (and hence a normal distribution) accurately describes the fluctuations of $f_0^{\{i\}}$ about their mean.

In the special case that the weights $w_m(q)$ are uncorrelated, the covariance matrix becomes diagonal (i.e. $\sum_{m,n} = \lambda_m^2 \delta_{m,n}$), and Eq. (10) takes the simple form

$$P_0[\mathbf{w};q] = \prod_{m=1}^M P[w_m;q], \quad (11)$$

$$P[w_m;q] = \frac{1}{\sqrt{2\pi\lambda_m^2(q)}} \exp \left\{ \frac{-(w_m - \bar{w}_{m,0}(q))^2}{2\lambda_m^2(q)} \right\}. \quad (12)$$

While Eq. (11) neglects dependency between weights, we show in Sec. II D that this example provides an important connection between Eq. (4) and other well known optimization techniques.

In Fig. 5, we illustrate priors calculated according to Eq. (12) for an initial set of force curves $\hat{f}^{\{i\}}$ that were generated using the FM method. For eight temperatures $\mathbf{q} = (100 \text{ K}; 200 \text{ K}, \dots, 800 \text{ K})$, we generated 100 realizations $f_0^{\{i\}}(r)$, which were used to construct sample means and variances according to Eqs. (7) and (8); see the top sub-plots, where ϕ are the Chebyshev polynomials and the number of modes $M = 10$ (only 7 are shown). The bottom-left sub-plot shows estimates $\bar{\mathbf{w}}_0$ and their uncertainties according to Eq. (12), where we used the output of the first correction loop (iterated twice) as inputs to the GPR. In the

bottom-right sub-plot, we reconstruct the mean force curves from these average weights according to Eq. (1). See Appendix D for more details on how these computations were performed.

We end this section with the caveat that, in general, there are many ways to construct reasonable priors. In essence, the task amounts to one of modeling, and this science is well developed; see, for example, Chapter 3 of Ref. 42. Our analysis is meant to be exploratory, and it is likely that more advanced interpolation methods based on statistical mechanics (e.g. as in Ref. 15) could lead to more suitable priors than we derive here. Nonetheless, we believe the usefulness of our approach here is born out by the results of the following sections and omit more in-depth studies as being beyond the scope of this work.

C. Sensitivity analysis and the likelihood function

The key idea underlying the construction of our likelihood function is to invoke a linear correction equation based on Taylor's theorem and Newton's method. To motivate this formalism, fix q , denote an initial set of weights as $\mathbf{w}_0$, and temporarily assume that $\mathbb{T}(q)$ is deterministic.⁴³ Provided $\mathbb{T}(q) = \mathbb{S}[\mathbf{w}_0, q]$, we can estimate new weights $\mathbf{w}$ via an equation of the form

$$(q) - [\mathbf{w}_0, q] = (\mathbf{w} - \overline{\mathbf{w}}_0) \cdot \nabla_{\mathbf{w}} [\mathbf{w}, q] \big|_{\mathbf{w}=\overline{\mathbf{w}}_0} + \mathcal{O}(|\mathbf{w} - \overline{\mathbf{w}}_0|^2) \quad (13)$$

where $\nabla_{\mathbf{w}} \mathbb{S}[\mathbf{w}, q] \big|_{\mathbf{w}=\mathbf{w}_0}$ is the gradient of the CG simulations with respect to the weights at $\mathbf{w}_0$, and terms $\mathcal{O}(|\mathbf{w} - \mathbf{w}_0|^2)$ are assumed to be negligible.⁴⁴ Equation (13) is equivalent to requiring that $\mathbb{S}[\mathbf{w}, q] = \mathbb{T}(q)$ under a linear approximation of the CG simulations.

In order to make use of Eq. (13), we must estimate the functional derivatives $\nabla_{\mathbf{w}} \mathbb{S}[\mathbf{w}, q]$, which we also refer to as the linear sensitivities. Let $\mathbf{s} = (s_1, s_2, \dots, s_D)^T$ denote the vector of such sensitivities at fixed q . We formally define these quantities via

$$s_j(f, q) = \lim_{\varepsilon \rightarrow 0} \frac{[f(r) + \varepsilon \phi_j(r), q] - [f(r), q]}{\varepsilon}. \quad (14)$$

In general, it is impossible to evaluate this limit analytically, so we resort to approximating

$$s_j(f, q) \approx \frac{[f(r) + \varepsilon \phi_j(r), q] - [f(r), q]}{\varepsilon}, \quad (15)$$

for $\varepsilon \ll 1$. In practice, evaluating Eq. (15) simply amounts to running CG simulations with the inputs $f(r) + \varepsilon \phi_j(r)$ and computing the relevant finite differences. Keeping $M = \mathcal{O}(10)$ modes, we are able to estimate the functional derivatives at a modestly large number $\mathcal{O}(10)$

of anchor points $q_k \in [q_{\min}, q_{\max}]$ using a few hundred CPU hours and $\varepsilon = 0.01$. See Fig. 6 for sensitivities of density-versus temperature simulations.

When computing sensitivities for the second correction loop of Fig. 3, it may be necessary to interpolate their values between a grid of anchor points. We omit discussion of this interpolation, as it can be done by the same steps that lead to the probabilistic models of $\mathbf{w}_0$ and $\mathbb{T}$; see Appendix D. The bottom sub-plot of Figure 6 shows an example of interpolated sensitivities for our running example of density-versus-temperature relations. Note that while we use the same anchor points as in Sec. II B, this is done entirely for convenience. In principle any set of anchors is acceptable so long as they are chosen where one has estimates of $\mathbf{w}_0$. As in the case of $\mathbb{T}(q)$, it is useful to express the sensitivities via an expression of the form

$$\mathbf{s}(q) = \bar{\mathbf{s}}(q) + \mathbf{D}\boldsymbol{\eta}_s(q), \quad (16)$$

where $\boldsymbol{\eta}_s(q) = (\eta_1(q), \eta_2(q), \dots, \eta_S(q))^T$ is a vector of i.i.d. Gaussian random variables, S is the number of anchor points, and $\mathbf{D}$ is an $S \times S$ matrix (determined during GPR; cf. Appendix D) that describes correlations between the sensitivities. The probability density for $\mathbf{s}(q)$ is identical to Eq. (10), with $\mathbf{w}_0(q)$ replaced by $\bar{\mathbf{s}}(q)$ and $\boldsymbol{\Sigma}$ replaced by $\mathbf{D}\mathbf{D}^T$. In Fig. 6, we take $\mathbf{D}$ to be diagonal with respect to mode number and assume that the covariance between state-points is given by Eq. (D2) when used in the second correction loop.

In order to construct a likelihood function, we first re-write Eq. (13) by replacing deterministic quantities with their probabilistic counterparts. Defining $\mathbf{w}_0 := \mathbf{w} - \mathbf{w}_0$, we find that up to first order, corrections must satisfy

$$\delta - \Delta \mathbf{w}_0 \cdot \bar{\mathbf{s}}(q) = -\eta_T(q) + \Delta \mathbf{w}_0 \cdot \mathbf{D}\boldsymbol{\eta}_s(q), \quad (17)$$

where $\delta = \mathbb{T}(q) - \mathbb{S}[\mathbf{w}_0, q]$; cf. Eq. (5). The above expression states that that we can effect a change δ in the CG predictions via $\mathbf{w}_0$, but subject to the fact that our target $\mathbb{T}$ and sensitivities are known imprecisely. We identify a likelihood function by noting that the left-hand side of Eq. (17) is normally distributed with mean zero and variance $\sigma^2 + (\Delta \mathbf{w}_0 \cdot \mathbf{D})^2$; viz

$$\mathcal{L}[\Delta \mathbf{w}_0 | \delta] := P[\delta | \Delta \mathbf{w}_0] = \frac{1}{\sqrt{2\pi[\sigma^2 + (\Delta \mathbf{w}_0 \cdot \mathbf{D})^2]}} \exp \left[\frac{-(\delta - \Delta \mathbf{w}_0 \cdot \bar{\mathbf{s}})^2}{2[\sigma^2 + (\Delta \mathbf{w}_0 \cdot \mathbf{D})^2]} \right], \quad (18)$$

where we equate the likelihood with $P[\delta | \mathbf{w}_0]$, the probability of effecting a change δ in the CG predictions conditioned on $\mathbf{w}_0$.

D. Bayesian update: construction and interpretation

Given the prior probability in Eq. (10) and likelihood function in Eq. (18), Bayes theorem yields

$$P_1[\mathbf{w};q] \propto P_0[\mathbf{w};q]\mathcal{L}[\Delta\mathbf{w}_0(q)], \propto \frac{1}{\sqrt{2\pi[\sigma^2+(\Delta\mathbf{w} \cdot \mathbf{D})^2]}} \exp \left\{ \frac{-1}{2} \left[\frac{(\delta - \Delta\mathbf{w} \cdot \bar{\mathbf{s}})^2}{\sigma^2 + (\Delta\mathbf{w} \cdot \mathbf{D})^2} + \Delta\mathbf{w}^T \Sigma^{-1} \Delta\mathbf{w} \right] \right\} \quad (19)$$

where a constant of proportionality is required to ensure that P_1 is normalized. Several comments are in order. For one, δ , $\mathbf{D}$, and σ^2 are all functions of q . Second, the general form of Eq. (19) makes it difficult to analytically compute the mean $\int \mathbf{w} P_1[\mathbf{w}] d\mathbf{w}$ and associated covariances, which are likely to be of most interest to modelers. Nonetheless, it is still possible to understand key properties P_1 , which can lead to simplifications and/or aid in numerical integration. In particular, P_1 becomes large when both the square root and exponent are close to zero. Minimizing the former entails orienting $\mathbf{w}_0$ in directions for which the sensitivities are well characterized (i.e. the corresponding eigenvalues of $\mathbf{D}$ are small). Maximizing the exponent (which is always non-positive) requires that $\mathbf{w}_0$ be small relative to the eigenvalues of Σ and oriented in directions of high sensitivity. From these observations, we conclude that P_1 is a compromise seeking the smallest change from the original $\mathbf{w}_0$ while maximizing the effect on $\mathbf{s}$.

In practical applications, the inexpense of CG simulations may allow one to characterize the sensitivities $\mathbf{s}$ sufficiently well so that we can approximate $\mathbf{D} = 0$. In this case, the posterior takes the simple form of a Gaussian

$$P_1[\mathbf{w}] \propto \exp [(\mathbf{w} - \bar{\mathbf{w}}_1)^T \mathbf{\Lambda} (\mathbf{w} - \bar{\mathbf{w}}_1)] \quad (20)$$

$$\Lambda_{i,j} = [\Sigma^{-1}]_{i,j} + \frac{s_i s_j}{\sigma^2} \quad (21)$$

$$\bar{\mathbf{w}}_1 = \bar{\mathbf{w}}_0 + \frac{\delta}{\sigma^2} \mathbf{\Lambda}^{-1} \mathbf{s}, \quad \bar{w}_{1,i} = \bar{w}_{0,i} + \delta \frac{[\Sigma \mathbf{s}]_i}{\sigma^2 + \mathbf{s}^T \Sigma \mathbf{s}}, \quad (22)$$

where the $\bar{w}_{1,i}$ are computed with the aid of the formula

$$[\Lambda^{-1}]_{i,j} = \sum_{i,j} - \frac{[\sum \mathbf{s}]_i [\sum \mathbf{s}]_j}{\sigma^2 + \mathbf{s}^T \sum \mathbf{s}}, \quad (23)$$

due to Miller.⁴⁵ The mean weights $\mathbf{w}_1$ should yield improved predictions when used in place of $\mathbf{w}_0$ in the CG simulations.

We highlight several features of Eq. (22). For one, using the prior Eq. (12) reduces $\mathbf{w}_1$ to the much simpler

$$\bar{w}_{1,i} - \bar{w}_{0,i} = \delta \frac{s_i \lambda_i^2}{\sigma^2 + (\mathbf{s} \cdot \boldsymbol{\lambda})^2} \quad (24)$$

If we further suppose that a single sensitivity s_i or uncertainty λ_i^2 is much greater than all other terms, then the corresponding difference $\bar{w}_{1,i} - \bar{w}_{0,i}$ dominates the vector $\mathbf{w}_1 - \mathbf{w}_0$. In these limits, $\mathbf{w}_1$ modifies only (i) the most sensitive element of $\mathbf{w}_0$ for large s_i , or (ii) the most uncertain element if λ_i^2 dominates. Furthermore, it is possible to show that in the limit $\sigma^2 \rightarrow 0$, $\mathbf{w}_1$ is the solution to the constrained optimization problem wherein we seek to minimize the norm

$$L_{2,\lambda}(\mathbf{w} - \bar{\mathbf{w}}_0) = \sum_i \frac{([\mathbf{w} - \bar{\mathbf{w}}_0]_i)^2}{\lambda_i^2} \quad (25)$$

subject to the constraint that $\mathbf{w}_0 \cdot \mathbf{s} = \delta$. Equation (25) is simply the Euclidean norm of the vector $\mathbf{w}_0$ with each term reduced according to the uncertainty with which we know $\mathbf{w}_0$ in the prior P_0 .

When σ^2 is finite, Eq. (24) also optimizes Eq. (25) subject to the modified constraint

$$\Delta \mathbf{w}_0 \cdot \mathbf{s} = \delta \frac{(\mathbf{s} \cdot \boldsymbol{\lambda})^2}{\sigma^2 + (\mathbf{s} \cdot \boldsymbol{\lambda})^2} \leq \delta. \quad (26)$$

Again, limiting cases bring clarity to this equation. In particular, when $\sigma^2 \gg (\mathbf{s} \cdot \boldsymbol{\lambda})$, one finds that $\mathbf{w}_0 \approx 0$; i.e. the uncertainty in the atomistic results is too high to justify forcing the CG simulations to agree with the surrogate. On the other hand, when $\sigma^2 \ll (\mathbf{s} \cdot \boldsymbol{\lambda})$, the sensitivities and/or uncertainties $\boldsymbol{\lambda}$ are so large that we incur little inconsistency with the prior by requiring exact equality between the CG predictions and their mean atomistic counterparts.

Figure 7 shows the results of correcting force curves for our running example of density-temperature data first discussed in Figs. 5 and 6. We used our training data at $\mathbf{q} = (100 \text{ K}, 200 \text{ K}, \dots, 800 \text{ K})$ to estimate the mean weights and their uncertainty via Eqs. (7) and (8), which are inputs to the first correction loop in Fig. 3. Priors were computed according to Eq. (9), and sensitivities were recomputed for each iteration. After two passes through this loop, we interpolated the posterior mean forces at 10 K intervals and used these as inputs to the second correction loop in Fig. 3. We assumed independence of the modes to generate an interpolated prior according to Eq. (11) and set $\mu_0 = 0$ and $p = 2$, which are inputs to the GPR [cf. Eqs. (D1) and (D2) in Appendix D]. Approximating $\mathbf{D} = 0$, we were able to generate well-calibrated forces after only one iteration through the second correction loop.

III. DEPENDENCE ON UNDERLYING ASSUMPTIONS AND ACCELERATED COARSE-GRAINING

As is frequently the case with numerical optimization recipes, the Bayesian correction algorithm requires certain assumptions and inputs for practical implementation, e.g. hyperparameters (i.e. inputs to GPR) and prior probabilities. As many of these quantities are not typically encountered in the context of coarse-graining, we explore the dependence of our method on its underlying assumptions (Secs. III A and III B). A key theme repeatedly appears: knowledge of the target data is more important than details of the prior. In this vein, we also demonstrate how Eq. (4) can accelerate coarse-graining by correcting initial force curves that are constructed by force matching undersampled atomistic simulations (Sec. III C).

A. Dependence on basis set

Formally speaking, the posterior probability $P_1[f; q]$ that we recover from Eq. (4) should be independent of the basis set ϕ , provided (i) the latter is complete (in a mathematical sense), (ii) $M \rightarrow \infty$, and (iii) the linear approximation in Eq. (13) is valid. From a practical standpoint, however, P_1 will depend on the size of M and how well a given basis set captures relevant features of the CG forces. While it is beyond the scope of this work to present general statements about convergence, we do demonstrate that calibrating an initial set of force curves with different ϕ can lead to approximately the same CG predictions and posterior mean.

Here we consider two additional bases: Bessel functions with $M = 15$ modes and Legendre polynomials with $M = 10$ modes. For each set, we use our training force curves at $\mathbf{q} = (100 \text{ K}, 200 \text{ K}, \dots, 800 \text{ K})$ and update the modes according to the procedure underlying Figs. 3 and 7; see Sec. II D. In Fig. 8, we show the results of these computations. In both cases, we only require a few updates of the anchor points, followed by one update of the interpolated curves to yield good agreement with the target data. Thus for the three basis sets considered (including Chebyshev polynomials in Fig. 7), the algorithm converges with roughly the same amount of computational work.

B. Dependence on interpolation method

In this section, we verify that corrections according to Eq. (4) are relatively stable to changes in the hyperparameters (i.e. underlying assumptions) of the GPR. We fix $\{\phi_m\}$ to be the Chebyshev polynomials, set $M = 10$, and consider four additional interpolation schemes corresponding to the following hyperparameter choices (see also Appendix D):

- $\mu_0 = 0$ and $p = 1$,
- $\mu_0 = 0$ and $p = 1.5$,
- $\mu_0 = bT + c$ and $p = 2$,
- $\mu_0 = aT^2 + bT + c$ and $p = 2$,

where p is the exponent appearing in Eq. (D2), μ_0 is a prior assumption of the functional form of each weight as a function of temperature, and a , b , and c are free parameters given by a least squares fit to the posterior mean weights output by the first correction loop of Fig. 3.

Given these interpolation methods, we then perform the same steps as in the previous section and Sec. II D, interpolating weights as a function of T , computing sensitivities and updating force curves at 10 K increments. As before, we approximate $\mathbf{D} = 0$. Figure 9 shows the results of this process. While each interpolation scheme yields somewhat different predictions, the interpolated forces all converge to acceptable targets within a few updates.

In this figure, it is interesting to note that the linear interpolation ($\mu_0 = bT + c$) behaves far worse than the others. While it is not clear *why* this is so, our work flow nonetheless succeeds in calibrating the underlying forces. This example points to the difficulty of interpolation and need for robust work flows. In such cases, a reasonable approach is to accept the *ad hoc* nature of a given choice and attempt to systematically account for and correct its effects.

C. Accelerated coarse-graining

In the previous section, we considered how arbitrary changes in the prior affect the correction algorithm. Here we examine more closely the special case in which we change P_0 through the number N of force curves realized at each q_k . Specifically, we demonstrate that it is possible to find a reasonable $\bar{f}$ by constructing priors with one tenth of the atomistic simulation resources used in the FM algorithm.

Figure 10 illustrates the results of this task. To generate this data, we followed the steps leading to Fig. 7, but using one tenth of the i.i.d. force curves $f_0^{\{i\}}$ available at each of the anchor temperatures when constructing the prior. The target data was kept the same. To arrive at convergence, we require more iterations of the update procedure applied to the interpolated forces. However, given the computational cost associated with running atomistic simulations, significantly less resources are required to calibrate f as compared to Fig. 7. In particular, each realization of $f_0^{\{i\}}$ takes roughly 40 CPU hours to generate, while each CG simulation takes ≈ 10 CPU hours. Thus, Fig. 7 required about 32,000 CPU hours to generate

all the $f_0^{\{i\}}$ and roughly 4,500 CPU hours to run CG simulations, for a total of 36,500 CPU hours.⁴⁶ In contrast, Fig. 10 required 3,200 CPU hours to generate $f_0^{\{i\}}$ fig and 8,100 CPU hours for CG simulations, for a total of about 11,300 CPU hours, less than a third of the case for Fig. 7.

IV. DISCUSSION

A. Comparison with past works

Given the usefulness of CG simulations, it is not surprising that a vast amount of literature has been devoted to the subject of constructing and calibrating coarse-grained force fields; see, e.g. 7, 10, 11, 15–18, 20, 22, 32, 33, and 47. To the best of our knowledge, however, none of these works implement uncertainty quantification as a widespread and integral part of their analysis. Here we wish to emphasize that UQ can play a central role in these analyses by (i) explicitly revealing the subjective aspects of calibration, (ii) recasting the latter as a decision-making problem, and (iii) providing systematic and robust methods for choosing calibrated forces.

To be more specific, the Bayesian framework illustrates that infinitely many plausible force curves yield the target predictions. This degeneracy (in a mathematical sense) is perhaps already foreshadowed by the variety of CG methods that have been successfully developed: empirical Leonard Jones (LJ) potentials,^{11,18,47} iterative Boltzmann inversion (IBI),^{10,16,17} FM,^{5,6} etc.. But even within the context of a single method, statistical under-sampling, numerical error, and uncertainty in target data suggest that it is difficult, if not impossible to identify a unique, optimal force for a given coarse-graining procedure. Thus, any method for determining force curves involves a subjective choice by the modeler at some level.

Implementation of UQ throughout a CG procedure is useful in that it assesses the extent to which certain choices are more reasonable than others. This is illustrated by Eq. (25), wherein updates w are computed by minimizing the norm

$$L_{2,\lambda}(\Delta w) = \sum_i \frac{([\Delta w]_i)^2}{\lambda_i^2}. \quad (27)$$

Within the Bayesian framework, changing modes with small uncertainties would be heavily penalized, and hence implausible. We may interpret this as a compatibility statement: if a given weight output by the CG algorithm is very certain, a force curve with a very different value of this weight is inconsistent with the underlying assumptions or physics of coarse-graining. Thus, the simplest method of calibration (e.g. changing a single mode) could potentially be the least plausible.

Along related lines, our correction algorithm is robust, which is of particular importance when generating transferable potentials; see, e.g. Refs. 10, 16–18. These works showed that IBI typically outputs non-transferable forces, and other authors found that individually tailored methods (e.g. empirical potentials or specific interpolation methods) were required

to overcome this limitation.^{11,15,18,20,22,47} Figure 6 illustrates a possible culprit: large sensitivities to small uncertainties in a given mode can lead to wildly incorrect CG predictions. To highlight this issue, consider that for the Chebyshev basis at 800 K, an uncertainty of ± 2 GN / mol in the first mode (approximately 2.5% of the weight, corresponding to a vertical translation of the force) leads to a variation of roughly ± 0.25 g/cm³ (about $\pm 30\%$!) in the density.⁴⁸ Thus, small differences in reasonable interpolation methods can still lead to vastly different CG predictions. By updating forces according to a paradigm according to Eq. (4), we both (i) acknowledge this problem, while (ii) providing a systematic method for overcoming it; see Figs. 8–10.

B. Extensions

Given its similarity to Newton's method, our interpretation and formulation of Eq. (4) lends itself to several extensions. In particular, decades of research have produced a variety of higher-order update equations that are known to have faster rates of convergence than Eq. (13); e.g. see Ref. 49. Generally speaking, it should be straightforward to write expressions equivalent to our likelihood $\mathcal{L}$ using any of these updates. In many instances, however, the corresponding algorithms may require running additional CG simulations in order to compute changes in sensitivities or higher order corrections to the linear approximation. Moreover, the resulting likelihoods may not be Gaussian, making it harder to analytically compute posterior means. Nonetheless, such extensions could provide promising improvements over the simple linear approximation that we invoke.

A second extension of this work is to enable calibration of multiple physical properties simultaneously. Within the context of the linear approximation, it should be relatively straightforward to solve such a problem. In particular, one can define a sensitivity vector s_k for the k th physical property of interest and construct the corresponding likelihood $\mathcal{L}_k$ from s_k and the difference between the CG and atomistic results. Assuming, for example, independence of these sensitivities, Eq. (4) takes the form

$$P_1[w, q] \propto P_0[w, q] \prod_k \mathcal{L}_k[\delta_k, q]. \quad (28)$$

Our analysis is also easily generalized to the case in which the target data and sensitivities are both known precisely by reinterpreting likelihoods as Dirac delta functions. In a similar fashion, the above formulas can be generalized to multi-dimensional state points (e.g. temperature - pressure pairs) by recasting Eq. (13) in terms of the appropriate functional derivatives.

C. Limitations

A key limitation of our analysis is the need to run a large number of CG simulations. Often times, these simulations are so inexpensive that the practical limitations of the method are negligible. However, in certain cases the cost of a CG simulation may be high enough so that, perhaps only $\mathcal{O}(100)$ CG runs are available. In this case, the accuracy and/or efficiency of the method is limited by the available resources.

A related issue arises from the fact that, in general, CG simulations are not linear functions of the weights. Figure 7 illustrates the consequences of this; specifically, the correction algorithm sometimes over- or under-corrects the CG predictions, requiring additional iterations of the Bayesian update to fully optimize the forces. While this is a common occurrence in numerical root finders that use, e.g. Newton's methods, such algorithms often benefit from the extreme inexpense [often less than $\mathcal{O}(0.01)$ s per computation] of evaluating a function. Given that our function *is an entire simulation*, the impact of a poor guess f_0 on efficiency cannot be overstated; see Sec. III C. As suggested in the previous section, higher-order methods could also mitigate this issue.

Acknowledgments

All authors thank Debra Audus, Timothy Burns, Andrew Dienstfrey, and Anthony Kearsley for useful comments during preparation of the manuscript. TR thanks Sergei Izvekov (Army Research Lab) for the use of his FM code implemented on the Garnet cluster located at the ERDC DSRC. Part of this work was completed while PP was a postdoctoral fellow at the Institute for Mathematics and its Applications (IMA) under a 506 grant from NIST to the IMA. The IMA is a National Science Foundation Math Institute 507 funded under award DMS-0931945.

References

1. Li C, Strachan A. J Polym Sci, Part B: Polym Phys. 2015; 53:103.
2. Allison, J.; Collins, P.; Spanos, G., editors. Proceedings of the 1st World Congress on Integrated Computational Materials Engineering (ICME); John Wiley & Sons, Inc; 2011.
3. Li, M.; Campbell, C.; Thornton, K.; Holm, E.; Gumbsch, P., editors. Proceedings of the 2nd World Congress on Integrated Computational Materials Engineering (ICME); John Wiley & Sons, Inc; 2013.
4. Dienstfrey A, Phelan FR Jr, Christensen S, Strachan A, Santosa F, Boisvert R. Journal of the Minerals Metals and Materials Society (JOM). 2014; 66:1342.
5. Izvekov S, Voth G. J Chem Phys. 2005; 123:134105. [PubMed: 16223273]
6. Izvekov S, Voth G. J Phys Chem B. 2005; 109:2469. [PubMed: 16851243]
7. Moore TC, Iacovella CR, McCabe C. J Chem Phys. 2014; 140:224104. [PubMed: 24929371]
8. Bayramoglu B, Faller R. Macromolecules. 2012; 45:9205.
9. Izvekov S, Chung PW, Rice BM. J Chem Phys. 2010; 133:064109. [PubMed: 20707563]
10. Bayramoglu B, Faller R. Macromolecules. 2013; 46:7957.
11. Hsu DD, Xia W, Arturo SG, Keten S. J Chem Theory Comput. 2014; 10:2514. [PubMed: 26580772]
12. D'Adamo G, Pelissetto A, Pierleoni C. Soft Matter. 2012; 8:5151.
13. Das A, Andersen HC. J Chem Phys. 2012; 136:194114. [PubMed: 22612087]
14. Jacobson LC, Kirby RM, Molinero V. J Phys Chem B. 2014; 118:8190. [PubMed: 24605768]
15. Krishna V, Noid WG, Voth GA. J Chem Phys. 2009; 131:024103. [PubMed: 19603966]
16. Carbone P, Varzaneh HAK, Chen X, Müller-Plathe F. J Chem Phys. 2008; 128:064904. [PubMed: 18282071]
17. Ghosh J, Faller R. Mol Simul. 2007; 33:759.
18. Maerzke KA, Siepmann JI. J Phys Chem B. 2011; 115:3452. [PubMed: 21395331]
19. Brini E, Herbers CR, Deichmann G, van der Vegt NFA. Phys Chem Chem Phys. 2012; 14:11896. [PubMed: 22722404]
20. Farah K, Fogarty AC, Boehm MC, Müller-Plathe F. Phys Chem Chem Phys. 2011; 13:2894. [PubMed: 21157607]
21. Martin MG, Siepmann JI. J Phys Chem B. 1998; 102:2569.

22. Qian H-J, Carbone P, Chen X, Karimi-Varzaneh HA, Liew CC, Müller-Plathe F. *Macromolecules*. 2008; 41:9919.
23. Ganguly P, van der Vegt NFA. *J Chem Theory Comput*. 2013; 9:5247. [PubMed: 26592264]
24. Hsu DD, Xia W, Arturo SG, Keten S. *J Chem Theory Comput*. 2014; 10:2514. [PubMed: 26580772]
25. Hsu DD, Xia W, Arturo SG, Keten S. *Macromolecules*. 2015; 48:3057.
26. The Bayesian interpretation of probabilities is distinct from ones based on ideas of chance and random draws; see, for example, Ref. 55. For a good example of Bayesian calibration techniques applied to computer models, see Ref. 56 and the references contained therein. See Sec. II A for more discussion.
27. Das A, Andersen HC. *J Chem Phys*. 2012; 136:194113. [PubMed: 22612086]
28. Das A, Andersen HC. *J Chem Phys*. 2009; 131:034102. [PubMed: 19624176]
29. Rasmussen, CE.; Williams, CKI. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press; 2005.
30. Santner, TJ.; Williams, B.; Notz, W. *The Design and Analysis of Computer Experiments*. Springer-Verlag; 2003. p. 283
31. Sacks J, Welch WJ, Mitchell TJ, Wynn HP. *Stat Sci*. 1989; 4:409.
32. Potestio R, Peter C, Kremer K. *Entropy*. 2014; 16:4199.
33. Peter C, Kremer K. *Faraday Discuss*. 2010; 144:9. [PubMed: 20158020]
34. Louis A. *J Phys Condens Matter*. 2002; 14:9187.
35. Lu J, Qiu Y, Baron R, Molinero V. *J Chem Theory Comput*. 2014; 10:4104. [PubMed: 26588552]
36. Lobanova O, Avendao C, Lafitte T, Müller EA, Jackson G. *Mol Phys*. 2015; 113:1228.
37. Several additional observations suggest that this perspective is appropriate: (i) there are many reasonable coarse-graining methods for computing f_0 ; (ii) it is often difficult to identify a best interpolant; and, as we show, (iii) many plausible force curves can yield the same target predictions. Thus, calibration of CG forces already requires subjective decision-making, which we only seek to acknowledge and systematize.
38. In the unusual case that $\eta_f \rightarrow 0$, is known to arbitrary accuracy, one can take $\tilde{f}_0$, which leads to Dirac delta-function priors following the procedure in this section.
39. Generally speaking, an arbitrary CG algorithm may yield $f(r, q)$ at a finite, but large number of discretized points on the r domain. Here we assume that the spacing between such points is so small that the weights in Eq. (1) can be computed to arbitrary precision using numerical approximations of the appropriate projection integrals.
40. For IBI, it is reasonable that $N = 1$, so that we might need to invoke different methods for determining uncertainties computed in Eq. (8).
41. We use $\bar{w}_0$ to denote the mean of a probability density in general so that we do not have to write separate versions of Eq. (4) for the first and second correction loops in Fig. 3.
42. Berger, J. *Springer Series in Statistics*. Springer; 1985. *Statistical Decision Theory and Bayesian Analysis*.
43. Given that CG simulations are inexpensive, we assume that their outputs are always deterministic.
44. Equation (13) does not specify a unique w whenever $M \geq 2$; in such cases, one could use the Moore-Penrose pseudoinverse to solve for w . The Bayesian update that we describe is a variant of this procedure.
45. Miller KS. *Math Mag*. 1981; 54:67.
46. We approximate 900 CPU hours for each iteration of the CG simulations, be it an anchor point update, computation with interpolated forces, or computation with corrected interpolated forces.
47. Mognetti BM, Virnau P, Yelash L, Paul W, Binder K, Müller M, Mac-Dowell LG. *J Chem Phys*. 2009; 130:044101. [PubMed: 19191371]
48. In this light, it is perhaps no surprise that transferable forces are notoriously difficult to calibrate.
49. Stoer, J.; Bartels, R.; Gautschi, W.; Bulirsch, R.; Witzgall, C. *Introduction to Numerical Analysis, Texts in Applied Mathematics*. Springer; New York: 2002.
50. “Scienomics, maps platform, version 3.4, 2014, France,”.

51. Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.
52. Plimpton S. J Comput Phys. 1995; 117:1.
53. Sun H, Mumby S, Maple J, Hagler A. J A Chem Soc. 1994; 116:2978.
54. Patrone PN, Dienstfrey A, Browning AR, Tucker S, Christensen S. Submitted to Polymer. 2015
55. Gelman, A.; Carlin, J.; Stern, H.; Dunson, D.; Vehtari, A.; Rubin, D. Bayesian Data Analysis. 3. Chapman & Hall/CRC Texts in Statistical Science, Taylor & Francis; 2013.
56. Kennedy MC, O'Hagan A. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2001; 63:425.

Appendix A: Simulation Details

The reference system used to construct all of the figures consisted of 10 chains of 50 monomers of atomistically resolved atactic polystyrene. To generate an initial configuration we used the amorphous builder within the commercial package MAPS by Scienomics^{50,51} All molecular dynamics simulations were performed with the open source package LAMMPS⁵²; temperature and pressure was maintained with a Nose-Hoover thermostat and barostat with coupling constants of 0.1 ps and 1.0 ps, respectively. The forcefield applied to the atomistic system was PCFF⁵³ with a cutoff of 1.2 nm for both Lennard-Jones (van der Waals) and Coulombic nonbonded interactions. Long-range electrostatic interactions were calculated using the particle-particle particle-mesh method with a precision of 1.0×10^{-6} . Due to ease of implementation, the tabular force fields generated by the Bayesian calibration found in this work (e.g. Fig. 5) were calculated using dimensionless separations. Because very few interactions for CG sites in the atomistic reference system existed for distances less than 0.48 nm apart, this was defined as $r = 0$. To determine forces at separations less than 0.48 nm the first 600 values were fit to a linear function and extrapolated to 0 nm. The spacing between force values in real units was 5.0×10^{-5} nm. A real space cutoff of 1.4 nm was used in all CG simulations, corresponding to the dimensionless $r = 1$. Bond and bond angle potentials were derived using Inverse Boltzmann inversion at 500 K and applied to all temperatures.

To relax the initial atomistic configuration, we first minimized the energy of the amorphous structure using the Polak-Ribiere conjugate gradient method with a force tolerance of 1.0×10^{-8} . Four consecutive NVT simulations at 298 K were then performed for 5.0×10^5 timesteps each, with a timestep of 0.001 fs, 0.01 fs, 0.1 fs, and 1.0 fs, respectively. All following simulations were performed with a timestep of 1.0 fs. A 0.5 ns NVT simulation that continuously increased the temperature from 298 K to 600 K was then performed. Finally, a 2 ns NPT simulation at 600 K was run to ensure an equilibrated structure. To obtain initial configurations for the higher temperatures investigated in this study, we further equilibrated for 0.5 ns at those temperatures. To obtain initial configurations for the lower temperatures we decreased the temperature continuously for 5 ns and then equilibrated for another 0.5 ns. After equilibration, a production run at each temperature was performed for 12 ns with coordinates, velocities, and forces recorded every 1 picosecond. This capture rate determines the number of frames used in the FM averaging of Eq. (B3) in Appendix B.

All CG simulations were performed in the NPT ensemble with a timestep of 1.0 fs. Initial configurations were generated from a direct mapping of the atomistic system at each temperature. Equilibration consisted of 100 ps, followed by a 10 ns production run.

Appendix B: Force Matching

The force matching method applied to coarse-grained systems has been described in detail elsewhere.^{5,6} Here we only outline specifics as they pertain to the initial guess for the CG force curves.

Force matching yields the optimal (in the least-squares sense) pairwise decomposition of the atomistic free-energy surface in CG coordinates. The result is a series of tabular functional-free force curves $f^{(j)}(r, T)$ that we process via the work flow illustrated in Fig. 3.

Given an ensemble of atomic positions, momenta, and forces, $\mathbf{r}_j$, $\mathbf{p}_j$, and $\mathbf{f}_j$, define the following mappings:

$$\begin{aligned} \mathbf{R}_J &= \sum_{j=1, a_J} c_{jJ} \mathbf{r}_j, \\ \mathbf{P}_J &= \sum_{j=1, a_J} \mathbf{p}_j, \\ \mathbf{F}_J^{atm} &= \sum_{j=1, a_J} \mathbf{f}_j, \end{aligned} \quad (\text{B1})$$

where a_J is the number of atoms in the J th group (in this case 16) and $c_{jJ} = m_j/M_J$ is the ratio of an atoms mass to the CG total mass; see Fig. 11. Next, we postulate that the total force on the J th group can be expressed in the form

$$\mathbf{f}_J^{FM}(\Omega) = \sum_{K \neq J} f^{FM}(\mathbf{R}_{JK}, \Omega) \mathbf{n}_{JK}, \quad (\text{B2})$$

where $\mathbf{R}_{JK} = |\mathbf{R}_{JK}| = |\mathbf{R}_J - \mathbf{R}_K|$, $\mathbf{n}_{JK} = \mathbf{R}_{JK}/R_{JK}$, and $f^{FM}(r, \Omega)$ is a yet-to-be-determined function of some free parameters. Here we represent f^{FM} as a cubic spline on a large collection of points that mesh the r -domain out to the cutoff distance; thus correspond to the spline parameters. For N_{total} CG sites, we then determine f^{FM} by minimizing

$$\chi^{(\Omega)} = \left\langle \sum_{J=1}^N |\mathbf{F}_J^{atm} - \mathbf{f}_J^{FM}(\Omega)|^2 \right\rangle, \quad (\text{B3})$$

subject to the constraint that the CG force field also reproduce the atomistic virial, viz.

$$\sum \mathbf{f}_J^{FM} \mathbf{R}_J = 3VP^{inst} \quad (\text{B4})$$

where P^{inst} is the instantaneous pressure of the reference configuration at a volume V . In Eq. (B3), the averaging (indicated by brackets $\langle \cdot \rangle$) is taken with respect to all of the frames that are saved during the atomistic simulation. The output $f^M(r, \Omega)$ of optimizing Eq. (B3) is a realization $f_0^{\{i\}}$ of Eq. (6). In order to compute multiple such realizations, we iterate this procedure using new intervals (or trajectories) of atomistic simulations to compute the averages in Eq. (B3).

Appendix C: Surrogate building and target models

In general, there is no universal technique for constructing a surrogate or target model; details often depend on the problem at hand. However, certain generic features are useful: (i) $\mathbb{T}(q)$ should provide estimates of the quantity of interest at all state-points of interest; (ii) such estimates should also have a quantified uncertainty; (iii) $\mathbb{T}(q)$ should be quick to evaluate, so that differences $\delta(q)$ are easy to compute.

In the case of simulated density-vs-temperature curves, we construct a surrogate by fitting a hyperbola to atomistic data.⁵⁴ The motivation for representing the data via hyperbolas stems from the observation that, when a polymer system is deep within the glassy or rubbery regimes, density-temperature relationships $\rho(T)$ should be approximately linear. The asymptotes of a hyperbola can capture this bilinear behavior, while its transition region can model intermediate temperatures. If we let γ denote the parameters of a hyperbola, we determine this quantity for our data by least-squares fit. For more details, see Ref. 54.

Given a hyperbola parameterized by γ , we model uncertainty in the atomistic predictions by a simple stochastic process. Specifically, letting $\rho_A(\gamma, T)$ denote the densities predicted by the hyperbola, we assume that the atomistic data $\rho_a(T)$ can be modeled via

$$\rho_a(T) = \rho_A(\gamma, T) + \eta_a(T), \quad (C1)$$

where $\eta_a(T)$ is a zero mean, delta-distributed Gaussian random process. In other words, we assume that $\eta_a(T)$ is uncorrelated across temperatures and has a fixed variance

$\text{Var}[\eta_a(T)] = \sigma^2$. In order to estimate σ^2 , we compute the residuals $R(T_k) = \rho_a(T_k) - \rho_A(\gamma, T_k)$ at the simulated anchor temperatures T_k and identify

$$\sigma^2 = \frac{1}{Q} \sum_{k=1}^Q R(T_k)^2 \quad (C2)$$

for Q anchor points. Given this parameter, we identify our target $\mathbb{T}(T) = \rho_a(T)$ according to Eq. (C1).

We note in passing that this simple model neglects effects (e.g. correlations) that could be represented by more sophisticated models. Here we do not pursue such models since our focus is on developing tools for updating CG force curves.

Appendix D: Gaussian-process regression

In the UQ and machine learning communities, Gaussian-process regression is a well-known set of techniques that can interpolate data while simultaneously assessing confidence in the interpolated results. As there are many references that provide comprehensive reviews of the subject, we only introduce the main ideas and highlight issues that are relevant for constructing force-curves.^{29–31} We confine our discussion to a simple example of interpolating a single weight as a function of temperature, although the formulas we discuss are easily generalized to more complicated scenarios.

To begin, let $\mathbf{q} = (q_1, \dots, q_Q)$ be a vector of Q state-points and $\mathbf{w}(\mathbf{q}) = (w(q_1), \dots, w(q_Q))$ correspond to an average weight estimated at individual q_k . The starting point of GPR is to assume that the $\mathbf{w}(\mathbf{q})$ are points sampled from a Gaussian stochastic process, i.e. a random function $\mathbf{w}(q)$ of whose joint probabilities between finitely many values of q is multivariate Gaussian. Mathematically, this means that the probability density of $\mathbf{w}$ taking values $\mathbf{w}(\mathbf{q})$ is given by

$$P[\mathbf{w}(\mathbf{q})] = \frac{1}{(2\pi)^{Q/2} |K(\mathbf{q}, \mathbf{q})|} \exp \left\{ -\frac{1}{2} [\mathbf{w}(\mathbf{q}) - \mu(\mathbf{q})]^T K(\mathbf{q}, \mathbf{q})^{-1} [\mathbf{w}(\mathbf{q}) - \mu(\mathbf{q})] \right\}, \quad (\text{D1})$$

where $\mu(q)$ is the average behavior of $\mathbf{w}$ as a function of q and $K(q, q')$ is a covariance matrix element indicating the extent to which $\mathbf{w}(q)$ is correlated with $\mathbf{w}(q')$. Loosely speaking, the goal of GPR is to use the data $(\mathbf{q}, \mathbf{w}(\mathbf{q}))$ to determine the functions μ and K . In effect, this amounts to finding the stochastic process $\mathbf{w}$ that is consistent with the available measurements.

Now, it is generally impossible for data to effect a regression by itself; we must invoke some additional or outside information. In the case of GPR, one postulates (i) a prior guess $\mu_0(q)$ for the mean, and (ii) a functional form of the covariance $K(q, q'; \psi)$, parameterized by some unknown constants ψ . The constants are often referred to as hyperparameters. For all of the regressions in this work, we use an altered form of the squared exponential covariance function

$$K(q, q'; \sigma, \ell, p) = \sigma^2 \sqrt{qq'} \exp \left[\frac{-|q - q'|^p}{\ell^2} \right], \quad (\text{D2})$$

p , σ^2 , and ℓ are free parameters. The exponent p alters the smoothness properties of the underlying random functions, while the term $\sqrt{qq'}$ is motivated by the observation that MD predictions are noisier at higher temperatures [recall K is applied to our running example of $\rho(T)$]. The σ^2 determines the scale of the overall noise and ℓ determines the temperature scale over which different observations are correlated. In this work, we manually control the values of p and $\mu_0(q)$, but determine σ and ℓ by means of a maximum

likelihood analysis, which works as follows. First define $\Lambda = \text{diag}(\text{Var}[\bar{w}(\mathbf{q})])$ to be a diagonal matrix of the uncertainties in $\bar{w}(\mathbf{q})$ [e.g. according to Eq. (8)]. Given this, we insert $K = K + \Lambda$ into Eq. (D1) and maximize the probability of finding our data $\mathbf{q}$, $\bar{w}(\mathbf{q})$ as a function of these two parameters (assuming that p and μ_0 are fixed). Including the term Λ accounts for the uncertainties in the sample estimates of $\bar{w}(\mathbf{q})$ when computing the hyperparameters.

Given the covariance function $K(q, q')$, it is a straightforward matter to estimate $\mu(q)$ and $\text{Var}[w(q)]$. Specifically, define $\hat{\mathbf{q}} = (q_1, \dots, q_Q, q)^T$ as the vector of state-points $\mathbf{q}$ plus the q of interest, and let $\hat{\mathbf{w}} = (\bar{w}(q_1), \dots, \bar{w}(q_Q), w(q))$ be the corresponding vector for w . The joint probability of finding $(\hat{\mathbf{q}}, \hat{\mathbf{w}})$ is

$$P(\vec{w}) = \frac{1}{\sqrt{(2\pi)^{1+Q} |\vec{K}(\vec{q}, \vec{q})|}} \exp \left\{ -\frac{1}{2} [\vec{w}(\vec{q}) - \mu_0(\vec{q})]^T \vec{K}(\vec{q}, \vec{q})^{-1} [\vec{w}(\vec{q}) - \mu_0(\vec{q})] \right\}, \quad (\text{D3})$$

where $K(q_k, q_k) = K(q_k, q_k) + \text{Var}[\bar{w}(q_k)]$ for $q_k = q$ and $K = K$ for all other pairs of state-points. Given that $w(q)$ is the only unknown quantity appearing in the above expression, we can rewrite the probability in terms of this single variable to find

$$P[w(q)] \propto \exp \left\{ -\frac{(w - \mu_1(q))^2}{2\text{Var}[w(q)]} \right\}, \quad (\text{D4})$$

where

$$\mu_1(q) = \mu_0(q) + K(q, \mathbf{q}) [K(\mathbf{q}, \mathbf{q}) + \Lambda]^{-1} \bar{w}(\mathbf{q}) \quad (\text{D5})$$

$$\text{Var}[w(q)] = K(q, q) - K(q, \mathbf{q}) [K(\mathbf{q}, \mathbf{q}) + \Lambda]^{-1} K(\mathbf{q}, q). \quad (\text{D6})$$

Note that this approach updates the prior $\mu_0(q)$ by taking into account correlations in the data relative to our initial guess for a functional form. Also, when $\Lambda \rightarrow \infty$, note that $\mu_1 \rightarrow \mu_0$; that is, data with large uncertainties provide us with no information about how we should update our prior μ_0 .

It is straightforward to generalize these results to the case where the w are correlated across modes and/or additional information about uncertainties in our estimates of $\bar{w}$ are brought to bear. We refer the reader to Refs. 29 for more information.

FIG. 1.

Comparison of coarse-grained density-temperature relation (solid line) and atomistic results ($\times$) for polystyrene. To generate atomistic data, we simulated a roughly 8000 atom system for 10 ns at anchor temperatures $T = 100$ K, 200 K, ..., 800 K and averaged the densities recorded over the corresponding intervals of time (cf. Appendix A). To generate CG forces, we applied the Force-Matching (FM) algorithm to the respective atomistic trajectories (cf. Appendix B). It was necessary to add a temperature-dependent constant to the CG forces so that they would agree with the atomistic predictions at the anchor temperatures; see Sec. II and Figs. 7 for deeper discussion on this point. For a fixed f computed at anchor T_i , we then simulated densities at 10 K intervals in the domain $[T_i - 50$ K; $T_i + 40$ K]. The densities exhibit sharp jumps when the force curve changes. Such behavior is unphysical for systems like polystyrene and indicates that a given f is unsuitable for simulating the density outside of a small window around the temperature at which it was calibrated.

FIG. 2.

Process (solid black arrow) and information (dotted red arrow) flow in the Bayesian correction algorithm, corresponding to Step III outlined in Sec. I; see also Fig. 3. Symbol colors are used as follows: green denotes atomistic simulations, analysis thereof, and coarse-graining computations; blue denotes steps performed using custom scripts; yellow denotes CG simulation runs; two-color gradients denote steps that may take computations from either color classes. Process flow indicates the order of operations, whereas the information flow indicates how data is transferred between steps that are not directly connected by process flow. In this work, we do not recompute sensitivities for $n > 0$ when correcting interpolated forces.

FIG. 3.

Process (solid black arrow) and information (dotted red arrow) flow associated with our calibration work flow. Rectangles encompassing many symbols group the corresponding work flow components by the section where they are discussed. Symbol colors have the same meaning as in Fig. 2. See the latter for an expanded view of the correction loops (large, light-blue rectangles), corresponding to the Bayesian correction algorithm and Step III outlined in Sec. I.

FIG. 4.

Example of a surrogate model $\mathbb{T}(q) = \rho_d(T)$ for atomistic density-temperature relations $\rho_d(T)$. The $\times$ denote atomistic simulation results, while the black curve is the hyperbola fit to the data. The inset shows the underlying probability density at 100 K, while the yellow stripe indicates 3-sigma uncertainties in the atomistic predictions at all temperatures. It is reasonable to only require that the CG simulation results fall within the yellow stripe. See Appendix C for more details of how these uncertainties were calculated.

FIG. 5.

Raw and interpolated weights, along with corresponding uncertainties and force curves.

Upper left: average weights $w_0(T)$ at temperatures $T = 100$ K, 200 K, ..., 800 K and computed according to Eq. (7). The dashed lines are included to aid the eye in seeing trends; they do not represent the values of weights between integer modes. The weights are approximately zero for all p modes beyond the fifth. *Upper right:* Uncertainties

$\lambda_m = \sqrt{\text{Var}(w_m)}$ in the modes that are shown in the upper-left figure, computed according to Eq. (8). *Bottom left:* Modes 1 and 2 interpolated as a function of temperature using Gaussian-process regression. The $\times$, which are inputs to the GPR, correspond to mean weights that have been iterated twice through the first correction loop; see the anchor points in Fig. 7. The black-dotted lines are 3-sigma values indicating uncertainty in the interpolation. The inset shows the mean interpolated weights without uncertainties (for clarity). *Bottom right:* Average 800 K force-curve (solid) computed according Eq. (1) using the first 10 Chebyshev modes. The black-dotted line indicates the averaged, raw output of FM. The inset shows the average curves using 10 modes and computed at 100 K intervals from 100 K to 800 K.

FIG. 6.

Raw sensitivities and interpolations thereof. *Top plot:* Sensitivities for 8 temperatures as a function of mode number. The dotted lines are meant to aid the eye in seeing trends; they do not indicate values of sensitivities between integer modes. Note that the sensitivities tend to zero quickly. This suggests that there is little benefit to using high-frequency modes in Eq. (1) when modeling density. *Bottom plot:* Sensitivities of the first and second modes interpolated by temperature; black dotted lines indicate 3-sigma uncertainty intervals, while $\times$ indicate data points corresponding to the top plot. The inset shows modes 1 through 10 without uncertainties.

FIG. 7. Illustration of the workflow according to Fig. 3. To generate this plot, we proceeded as follows. First, we generated prior force curves according to Eq. (9) at 8 anchor temperatures. Then, we applied Eq. (4) to update the anchor points individually in the first correction loop (cf. Figs. 2 and 3), recomputing the sensitivities at each iteration. After two iterations in this loop, the anchor densities were within the yellow target region. We then interpolated the weights and sensitivities using a round of GPR to generate a prior and likelihood at 10 K intervals. After one iteration in the second correction loop, the CG densities fell within the target stripe.

FIG. 8. CG predictions according to different basis sets and calibrated using the work flow in Fig. 3. Bessel function results are indicated by $\times$ and dashed lines; Legendre polynomials are indicated by squares and solid lines. Note that despite minor differences in the CG predictions, both basis sets can be used to calibrate the force curves within a few iterations.

FIG. 9.

CG predictions arising from different interpolations of the Chebyshev weights. The anchor points were computed in the same way as Fig. 7 and are not shown. The main figure shows densities computed from interpolations of the posterior mean weights output by the irrst correction loop, shown in Fig. 7. The red dash-dot curve is the first correction of the $\mu_0 = a + bT$ interpolation, which was not sufficiently calibrated around 250 K. The inset shows near total overlap of the CG predictions after applying the correction algorithm. Despite significant differences in the initial CG densities, one iteration (two for the $\mu_0 = a + bT$) of the correction algorithm is sufficient to bring the results into agreement with one another and the atomistic predictions.

FIG. 10. Correction algorithm applied to a prior P_0 constructed from one tenth of the force curves in Fig. 7. Although we must run more CG simulations, the algorithm converges to forces that yield a reasonable density-temperature curve.

FIG. 11.
Illustration of CG beads compared to their atomistic counterparts.