materials

MDPI

Article

InfPolyn, a Nonparametric Bayesian Characterization for Composition-Dependent Interdiffusion Coefficients

Wei W. Xing [1], Ming Cheng [2], Kaiming Cheng [3,*], Wei Zhang [4] and Peng Wang [1,5,*]

[1] School of Integrated Circuit Science and Engineering, Beihang University, Beijing 100191, China; wxing@buaa.edu.cn
[2] Department of Civil and Environmental Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61801-2352, USA; ming6@illinois.edu
[3] Shandong Key Laboratory for High Strength Lightweight Metallic Materials, Advanced Materials Institute, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China
[4] Sino-French Engineer School, Beihang University, Beijing 100191, China; wei.zhang@buaa.edu.cn
[5] Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China
* Correspondence: chengkm@qlu.edu.cn (K.C.); wang.peng@buaa.edu.cn (P.W.)

Abstract: Composition-dependent interdiffusion coefficients are key parameters in many physical processes. However, finding such coefficients for a system with few components is challenging due to the underdetermination of the governing diffusion equations, the lack of data in practice, and the unknown parametric form of the interdiffusion coefficients. In this work, we propose InfPolyn, Infinite Polynomial, a novel statistical framework to characterize the component-dependent interdiffusion coefficients. Our model is a generalization of the commonly used polynomial fitting method with extended model capacity and flexibility and it is combined with the numerical inversion-based Boltzmann–Matano method for the interdiffusion coefficient estimations. We assess InfPolyn on ternary and quaternary systems with predefined polynomial, exponential, and sinusoidal interdiffusion coefficients. The experiments show that InfPolyn outperforms the competitors, the SOTA numerical inversion-based Boltzmann–Matano methods, with a large margin in terms of relative error (10x more accurate). Its performance is also consistent and stable, whereas the number of samples required remains small.

Keywords: Gaussian process; nonparametric Bayesian; interdiffusion coefficient; Boltzmann–Matano analysis

check for updates

Citation: Xing, W.W.; Cheng, M.; Cheng, K.; Zhang, W.; Wang, P. InfPolyn, a Nonparametric Bayesian Characterization for Composition-Dependent Interdiffusion Coefficients. *Materials* **2021**, *14*, 3635. https://doi.org/10.3390/ma14133635

Academic Editors: Antonio Lamura and Nicholas Fantuzzi

Received: 6 May 2021
Accepted: 23 June 2021
Published: 29 June 2021

1. Introduction

In many industrial processes that involve diffusion, e.g., alloy solidification, heat treatment, coating and electric packaging, the characterization of composition-dependent interdiffusion coefficients is a crucial task, as it quantifies a diffusion process clearly. The classic approach is based on Boltzmann-Matano analysis [1,2] which transforms the diffusion system into a linear system of equations. However, the Boltzmann–Matano analysis is only applicable to a binary system and becomes problematic for a system with more than three components, as it generates an under-determined system of equations that mathematically does not yield a unique solution. To address such a challenge, a number of methods have been developed over the years. Kirkaldy et al. [3] introduced the Kirkaldy–Matano method and provided extra equations to the linear system by adding additional M diffusion paths with intersection points. Although the results have been shown to be accurate, this method cannot generalize well to a multi-component system because the difficulty in experimentally generating intersection points grows drastically with the number of components $M+1$ [4]. Alternatively, methods based on one diffusion couple were proposed. Dayananda and Sohn [5] suggested integrating over certain composition ranges along the diffusion path to evaluate an average interdiffusion coefficient. Cermak and Rothova [6] later extended this method by choosing an infinitely small integration interval. Nevertheless, as is pointed out by Cheng et al. [7], the integration approach can lead to ill-conditional problems. A pseudobinary approach is introduced by considering only two components diffused into the diffusion zone. This method takes advantage of its time independence in the first-order linear equations and thus is very efficient when the pseudobinary condition is strictly satisfied in experiments. In practice, such experimental conditions may be difficult to meet, and in addition, for a multi-component system with a limited number of experimental samples, the linear equations are not capable of eliminating the extra solutions [8,9]. Separately, Zhang and Zhao [10] suggested a forward-simulation approach by iteratively optimizing the interdiffusion coefficients with repeated forward-simulations, similar to the classic inference approach for inverse problems. Although such a method is shown to be accurate and stable, it incurs an overwhelming computational cost because each iteration requires a complete diffusion simulation with a fine spatial-temporal grid.

Another branch of the one-diffusion-couple method lies in assuming a polynomial functional form for the interdiffusivities. Ideally, with a proper design of the polynomial function, one can compute the coefficients of the polynomial functions to estimate the interdiffusion using a numerical inverse method [11]. This numerical inverse approach is adopted by Chen et al. [4] to include the atomic mobility [12] to study the diffusion in the solution phase of a multicomponent system. To improve the efficiency of the numerical inverse method, Cheng et al. [13] recast the original parabolic inverse problem [11] as a linear multi-objective optimization to improve computation efficiency while maintaining similar accuracy. The optimization algorithm places weak limits on the experimental samples and is applied to interdiffusivities of solid solution as well as various alloy systems [14,15]. This approach was recently improved by Qin et al. [16], who suggest solving an under-determined linear system using compress sensing, a popular regularization technique, to increases stability against high order polynomial functions. However, the L^1 penalty imposed by compress sensing may introduce inappropriate prior assumptions, leading to inferior overall performance. We will see this issue in detail in the later experiment section.

Despite the notable performance and the popularity of the polynomial functional interdiffusion coefficient approaches, they share a fatal issue—how does one design the polynomial functions? Considering a quadternary system ($M = 3$), we have 3×3 polynomial functions requiring careful designs; modifying one function will affect the results of the other two. The challenge grows quadratically with the number M. Without proper design and repeated validations, the polynomial approach will lead to overfitting or underfitting, making this approach infeasible in practice.

One way to resolve this challenge is to use a complicated enough model with many polynomial terms and utilize classic Bayesian inference techniques [17] to estimate the posterior of the polynomial coefficients. In particular, Girolami [18] proposed an interesting Markov chain Monte Carlo (MCMC) for nonlinear and complex differential equations where the fully analytic expressions for the posterior distribution do not exist, which is similar to our problem. Despite its elegance and great accuracy, an MCMC approach often suffers from slow convergence and poor mixing, making it less practical for complex applications. To improve inference efficiency, the approximate Bayesian computation (ABC) and their variations, e.g., MCMC ABC and sequential Monte Carlo ABC (SMC ABC) are put forth by Alahmadi et al. [19]. However, despite being accurate and easy to implement, these types of sampling methods do not scale well with the number of parameters to be inferred. With unknown polynomials, the large number of parameters makes such methods impractical even with the latest accelerated variations [20,21].

Recently, the Gaussian process (GP) [22] has been utilized in dealing with data that are generated from a system of differential equations. As a back-box regression model, GP is proposed for fast parameter posterior estimations with the derivative information of the differential equations even with partially observed data [23]. The explicit derivative information is further utilized to improve a general GP's performance for data that are generated from differential equations [24]. The derivative in a given system of differential equations is further harnessed through a constraint manifold such that the derivatives of the Gaussian process must match an ordinary differential Equation (ODE) [25]. Despite their success, these works generally require explicitly known differential equations to work. Thus, they cannot directly be implemented for our problem.

A closely related work is [26], where GP is used as a generalization for a parametric function for binary images. However, their work cannot be directly implemented in our problem because our systems of equations will lead to a mixture of GPs that are augmented by the derivative of concentrations, whereas there is normally only one GP to estimate in most of the previous works [23–27].

To address the challenge of stable characterization of the interdiffusion coefficients, we introduce InfPolyn (Infinite Polynomial), a nonparametric Bayesian framework for the characterization of composition-dependent interdiffusion coefficients.

In particular, we first extend the general polynomial fitting method with an infinite number of polynomial terms. We then integrate out the polynomial coefficients with a Gaussian prior to derive a nonparametric functional form for the interdiffusion coefficients. To further improve our model with prior assumptions of an interdiffusion system, we introduce a diagonal-dominant prior for the functions of the interdiffusion coefficients. Unlike most Bayesian fitting problems, the interdiffusion coefficients are not known/observable to us. Thus, we introduce latent variables, the virtual ghost interdiffusion coefficients to address this issue. Finally, we derive a tractable joint likelihood function for model training. We compare InfPolyn with the state-of-the-art Matano-based numerically inverse methods and their variations. In ternary and quaternary systems with polynomial, exponential, and sinusoidal interdiffusion coefficients, InfPolyn shows a significant improvement over the competitors in terms of relative errors. In most of the experiments, our model shows an excellent performance with only 40 EPMA measurements, which is very desirable in practical interdiffusion coefficient estimations.

Essentially, InfPolyn is a functional estimation method tailored for the characterization of interdiffusion coefficients by imposing a mixture of the SOTA nonparametric models, GPs, and particular prior knowledge. Unlike the classic Bayesian inference approaches [18,19], InfPolyn does not require a time-consuming sampling process and is thus much more efficient. The highlights of this work for interdiffusion coefficient characterizations are as follows:

- InfPolyn does not require assumptions for the particular functional form of the interdiffusion coefficient; it is robust against overfitting and underfitting.
- InfPolyn does not require a significant number of training data.
- Prior knowledge of the interdiffusion system can be added easily in the framework of InfPolyn.

We hope the success of the nonparametric Bayesian framework can inspire more interesting applications in other interdiffusion coefficient estimation methods, e.g., the forward-simulation approach [10], in the material community. Thus, we publish our code and will maintain it as an open source toolbox on Github (https://github.com/wayXing/InfPolyn, accessed on 26 June 2021).

The rest of this paper is organized as follows. The interdiffusion coefficient estimation problem is introduced in Section 2, followed by a brief summary of the Matano–Boltzmann numerical inverse method with polynomial functions in Section 3. Our method is presented in Section 4, including the derivation, prior knowledge assumptions, and model training. The comparisons to the other SOTA methods through ternary and quandary systems are demonstrated in Section 5. Finally, Section 6 summarizes our work.

2. Statement of the Problem

We firstly formulate our problem mathematically as a foundation of this work. Consider a general one-dimensional diffusion system with $(M+1)$ components. According to Fick's second law [28], the diffusion process is fully characterized by

$$\frac{\partial c^i}{\partial t} = \nabla\left(\sum_{j=1}^{M} D_{ij}\nabla c^j\right), \quad i = 1, \ldots, M, \tag{1}$$

where ∇ is the partial derivative operator, c^i is the concentration of i component (note that the concentration is a function of space and time $c^i(t,x)$); D_{ij} is the interdiffusion coefficient w.r.t. the concentration gradient of component j. In many textbook examples, D_{ij} is assumed constant, but in practice, D_{ij} depends on the concentrations of all components $\mathbf{c} = (c^1, \cdots, c^M)^T$. Our goal is to find $D_{ij}(\mathbf{c})$ for all $i,j = 1, \cdots, M$ with, ideally, a concentrations profile $\mathbf{C} = (\mathbf{c}(t_e, x_1)^T, \cdots, \mathbf{c}(t_e, x_N)^T)^T \in \mathbb{R}^{N\times M}$ at some terminal time t_e and spatial locations $\{x_n\}_{n=1}^N$, where N is the number of sampling points at different locations. To avoid clutter, we denote $\mathbf{c}_n = \mathbf{c}(t_e, x_n)$. One may notice that an important factor, temperature, is not considered in the formulation. This is due to the general process of the experiment. To conduct the experiment and obtain the concentration profile, one first bonds two blocks of materials together and holds them at certain temperatures to activate interdiffusion at the initial interface. The annealing procedure may last from hours to days, depending on the speed of forming an interdiffusion zone wide enough for analysis. The temperature remains constant during the long-lasting annealing process except for the beginning and ending stages, which take short time. Thus, the temperature is considered constant for the interdiffusion coefficient characterizations. To fabricate just one diffusion couple, around 50–100 sample points are often selected in a line parallel to the direction of element diffusion within the interdiffusion zone. Each sample point is analyzed through electron probe micro-analysis (EPMA), which requires several minutes for the equipment to detect the concentrations. As a result, the experiment is time-consuming, and only a small amount of samples, i.e., small N, can be provided.

3. Boltzmann–Matano Polynomial Interdiffusion Coefficients

We follow the original work of the Boltzmann–Matano method [2], which is widely used to extract concentration-dependent interdiffusion coefficient $\{D_{ij}\}$ from experimental concentration profiles. The Boltzmann–Matano method first integrates Fick's law of diffusion (1) in time to obtain the following system,

$$\frac{1}{2t}\int_0^{c^i}(x - x_0)\mathrm{d}c^i = -\sum_{j=1}^{M} D_{ij}\nabla c^j, \quad i = 1, \ldots, M, \tag{2}$$

where c^i denotes the terminal concentration of i components, ∇c_j is the concentration gradient, and x_0 is the known Matano plane, defined by

$$\int_{-\infty}^{x_0}(1 - \mathbf{c}(x))dx = \int_{x_0}^{+\infty}\mathbf{c}(x)dx. \tag{3}$$

For a binary system, i.e., $M+1 = 2$, there is only one composition-dependent interdiffusion coefficient D_{11} to determine with one diffusion couple. Based on Equation (2), we can can directly compute $D_{11}(\mathbf{c}_n)$ for $n = 1, \cdots, N$ and then use any curve-fitting method to characterize the function of $D_{11}(\mathbf{c})$. For a ternary system, i.e., $M = 2$, we need to determine $D_{ij}(\mathbf{c})$ for $i = \{1,2\}$ and $j = \{1,2\}$. For each sample $\mathbf{c}_n$, we can write only two equations whereas there are four unknown parameters. This is an underdetermined system of equations to solve and will lead to multiple solutions. An effective and efficient solution is to assume a continuous function of interdiffusivity in a polynomial form, e.g., an independent quadratic form,

$$D_{ij}(\mathbf{c}) = w_{ij}^{(0)} + \sum_{i=1}^{M}\left(w_{ij}^{(i)}c^i + w_{ij}^{(M+i)}(c^i)^2\right), \tag{4}$$

where w is the weight coefficient in the polynomial function. Denote the flux of the L.H.S. of Equation (2) as u: we have $\mathbf{u}^i = (u_1^i, \cdots, u_N^i)$, where $u_n^i = \int_0^{c^i(x_n)}(x - x_0)\mathrm{d}c^i/2t$. Estimation of D_{ij} for $j = 1, \cdots, M$ can then be computed by solving the system of equation

$$u_n^i = -\sum_{j=1}^{M} D_{ij}(\mathbf{c}_n)\nabla c_n^j, \tag{5}$$

where $D_{ik}(\mathbf{c}_n)$ is the polynomial function fully determined by its weight coefficients given a particular functional form and $\mathbf{c}_n$. All weight coefficients $\mathcal{W} = \{w_{ij}^k\}$ in the polynomial functions can be computed by solving the optimization problem,

$$\underset{\mathcal{W}}{\operatorname{argmin}} \sum_{n=1}^{N}\left\|u_n^i + \sum_{j=1}^{M} D_{ij}(\mathbf{c}_n)\nabla c_n^j\right\|^2, \tag{6}$$

where $||\cdot||^2$ denotes the L^2 norm, which can be replaced with other norms.

Remark 1. *Since the estimation of $D_{ij}(\mathbf{c})$ for each $i = 1, \cdots, M$ only depends on $\mathbf{u}^i$ and is computed independently, we omit the index i and reformulate the Matano–Boltzmann method with polynomial interdiffusion coefficients to avoid clutter,*

$$\underset{\mathcal{W}}{\operatorname{argmin}} \sum_{n=1}^{N}\left\|u_n + \sum_{j=1}^{M} d_n^j \nabla c_n^j\right\|^2 = \underset{\mathcal{W}}{\operatorname{argmin}} \sum_{n=1}^{N}\left\|u_n + \nabla \mathbf{c}_n^T \mathbf{d}_n\right\|^2, \tag{7}$$

where u_n is the flux for any arbitrary component, and $\nabla \mathbf{c}_n^j$ is the concentration gradient for j component, both of which are computed from the profile $\mathbf{C}$; $d^j(\mathbf{c}_n)$ is the j column of any arbitrary row of $D_{ij}(\mathbf{c})$ that matches the chosen flux at concentration $\mathbf{c}_n$; $\mathbf{d}_n = \left(d^1(\mathbf{c}_n), \cdots, d^M(\mathbf{c}_n)\right)^T$ is the collection. We aim to reveal $d^j(\mathbf{c})$ for $j = 1, \cdots, M$.

Optimization for Polynomial Fitting

Equation (7) is a convex optimization problem provided that we have $N \geq 3(K+1)M$ EPMA samples and we use a K-order polynomial function of Equation (4) for all $d^j(\mathbf{c})$; the closed-form solution is presented in the Appendix A. This is certainly impractical for large M and/or K. In this case, regularization techniques, e.g., L^2-norm minimization or compress sensing, can be implemented to solve such an underdetermined system. The polynomial fitting approach with regularization is efficient in terms of computational time, space complexity, and implementation simplicity, thanks to many excellent software solutions, e.g., l_1-magic, *SPGL*1, and *SeDuMi* [29–31].

4. InfPolyn for Interdiffusion Coefficients

The challenge of the discussed polynomial based approach is the lack of guidelines on how to build the model, i.e., the selection of the order of the polynomial and the polynomial form. It is unclear how many polynomial terms are needed for each diffusion coefficient such that the model is not overfitting or underfitting. Although regularization techniques [16] can be implemented, the underlying assumptions of regularization are unclear, which can lead to unexpected performance. We need a systematic way to specify the diffusion coefficients with correct prior knowledge in order to achieve better results.

To this end, we propose a nonparametric Bayesian approach that is flexible enough to capture the complex nonlinear relation while restricting itself from overfitting the data by integrating all possible solutions.

4.1. *Infinite Order Polynomial Model*

To start with, we write the polynomial regression, e.g., Equation (4), in a compact form,

$$d^j(\mathbf{c}) = \mathbf{w}_j^T \boldsymbol{\phi}_j(\mathbf{c}) + \beta_j, \tag{8}$$

where the polynomial terms are denoted compactly as $\boldsymbol{\phi}_j(\mathbf{c}) = \left(c^1, (c^1)^2, \cdots, c^2, (c^2)^2, \cdots\right)^T$, where $\boldsymbol{\phi}_j(\cdot)$ is the predefined feature mapping that encodes the the polynomial functional form. Essentially, we can project the concentration $\mathbf{c}$ onto an $r-$dimensional feature space using an arbitrary mapping $\boldsymbol{\phi}_j(\mathbf{c}) \in \mathbb{R}^r$. Note that the constant term can also be absorbed into the feature mapping by setting the first element as 1. In the linear model case, the feature mapping is simply $\boldsymbol{\phi}_j(\mathbf{c}) = (1, \mathbf{c}^T)^T$.

Obviously, this polynomial approach is only accurate and stable when we roughly know the functional form of $\boldsymbol{\phi}_j(\mathbf{c})$. Furthermore, it requires a large number of parameters $\{\mathbf{w}_j, \beta_j\}$ to be estimated. Rather than estimating the weight parameters, we consider a matrix Gaussian prior for the weight vector $\mathbf{w}_j$,

$$\mathbf{w}_j \sim \mathcal{N}(\mathbf{0}, \Omega_j) = \frac{1}{\sqrt{(2\pi)^r |\Omega_j|}} \exp\left(\frac{-\mathbf{w}_j^T \Omega_j^{-1} \mathbf{w}_j}{2}\right), \tag{9}$$

where, $\Omega_j \in \mathbb{R}^{r\times r}$ indicates the correlation between the weight components. We then integrate out the weights and directly work with the marginal, which admits a closed-form solution,

$$\begin{aligned} p(d^j|\mathbf{c}) &= \int p(d^j|\mathbf{w}_j, \mathbf{c}) p(\mathbf{w}_j) d\mathbf{w}_j \\ &= \int \left(\mathbf{w}_j \boldsymbol{\phi}(\mathbf{c}) + \beta_j, \right) \mathcal{N}\left(\mathbf{w}_j | \mathbf{0}, \Omega_j\right) d\mathbf{w}_j \\ &= \mathcal{N}\Big(\beta_j, \left(\boldsymbol{\phi}_j(\mathbf{c})\right)^T \Omega_j \boldsymbol{\phi}_j(\mathbf{c})\Big). \end{aligned} \tag{10}$$

This is also known as the Gaussian process (GP) [22]. If we use a countably infinite feature space, i.e., $r \to \infty$, we formally define a sum over infinite polynomial terms. Thus, we call our model InfPolyn, infinite polynomial. Our model now becomes a nonparametric model that contains no explicit parameters $\mathbf{w}_j$. The model parameters are now encoded in $\left(\boldsymbol{\phi}_j(\mathbf{c})\right)^T \Omega_j \boldsymbol{\phi}_j(\mathbf{c})$, which indeed indicates a inner product in the the feature space spanned by $\boldsymbol{\phi}_j(\mathbf{c})$.

4.2. *Kernel Formulation*

Note that Ω_j is p.s.d. by its definition, and we can encode the inner product using a compact function, i.e., $k_j(\mathbf{c}, \mathbf{c}') = \left(\boldsymbol{\phi}_j(\mathbf{c})\right)^T \Omega_j \boldsymbol{\phi}_j(\mathbf{c})$. This is known as the kernel trick, which works by replacing the explicit feature mapping and covariance with a kernel function $k_j(\mathbf{c}, \mathbf{c}')$ to indicate an inner product in the feature space. Different kernels can capture different functional features. For instance, a periodic kernel can capture periodic functions such as sinusoidal functions. If we do not know the explicit form of the kernel function, which is true in most cases, the automatic relevance determination (ARD) kernel,

$$k_j(\mathbf{c}, \mathbf{c}') = \theta_{j0} \exp\left(-(\mathbf{c} - \mathbf{c}')^T \left(\mathbf{I} \odot (\tilde{\boldsymbol{\theta}}_j^T \tilde{\boldsymbol{\theta}}_j)\right)^{-1} (\mathbf{c} - \mathbf{c}')\right), \tag{11}$$

is commonly adopted as it generally provides good performance in most cases, especially in regression problems [22]. In this formulation, $\odot$ denotes the Hadamard product, $\mathbf{I}$ is an identity matrix, θ_{j0} is the scaling factor for the kernel function, and $\tilde{\boldsymbol{\theta}}_j \in \mathbb{R}^{M\times 1}$ is a vector with scaling factors for each input components, i.e., the concentrations of different elements. We denote $\boldsymbol{\theta}_j = (\theta_{j0}, \tilde{\boldsymbol{\theta}}_j^T)^T$ for clarity. These parameters $\boldsymbol{\theta}_j$ are known as the hyperparameters because they control the random process (10) statistically rather than in a determinant way (e.g., the aforementioned polynomial fitting). In this work, we use the ARD kernel throughout unless stated otherwise.

4.3. Ghost Interdiffusion Coefficients

Given that $d_j(\mathbf{c})$ is a Gaussian process as stated in Equation (10), any number of observations form a joint Gaussian distribution, based on which a closed-form likelihood can be easily calculated. Unfortunately, unlike the classic regression problems, we do not have any direct observations of $d_j(\mathbf{c})$, and we cannot directly obtain the optimized hyperparameters $\{\boldsymbol{\theta}_j, \beta_j\}$. To resolve this problem, we borrow the pseudo-inducing points idea [32] and introduce a set of virtual ghost interdiffusion coefficients, $\{h_{jg} = d_j(\mathbf{z}_{jg})\}_{g=1}^{G}$, that are sampled from the function $d_j(\mathbf{c})$ for virtual concentrations $\{\mathbf{z}_{jg}\}_{g=1}^{G}$. These latent variables must form a joint Gaussian distribution (because they are sampled from a Gaussian process of (10)),

$$\mathbf{h}_j \sim \mathcal{N}(\beta_j \mathbf{1}, \mathbf{K}_j), \tag{12}$$

where $\mathbf{h}_j = (h_{j1}, \cdots, h_{jG})^T$ is the collection of the ghost interdiffusion coefficients and $[K_j]_{gg'} = k_j\left(\mathbf{z}_{jg}, \mathbf{z}_{jg'}\right)$ is the covariance matrix computed through the kernel function and the latent locations $\mathbf{Z}_j = \{\mathbf{z}_{jg}\}_{g=1}^{G}$. Normally, $\mathbf{h}_j$ and $\mathbf{Z}_j$ are latent variables that need to be integrated out during the model training and predictions.

4.4. Diagonal-Dominating Prior

Following Occam's razor, if the dominant diagonal diffusion coefficients $D_{ii}(\mathbf{c})$ for $i = 1, \cdots, M$ can fully explain the diffusion process, it is reasonable to suppress the non-diagonal diffusion coefficients $D_{ij}(\mathbf{c})$ for $i \neq j$ to encourage a simpler model. To inject this preference of model, we design a special Laplace prior for the mean value for each Gaussian process of (12),

$$\beta_j \sim \text{Laplace}(0.01^{(1-\delta(i,j))}, 0.1), \tag{13}$$

where $\delta(\cdot,\cdot)$ is the delta function and i is the row that matches the choice of $\mathbf{d}^j(\mathbf{c})$. We use a Laplace prior rather than a Gaussian prior to encourage sparsity of the diffusion concentration for non-diagonal locations. The particular prior parameters may be adjusted according to a different system to reflect our prior knowledge.

4.5. Joint Model Training

With each interdiffusion coefficient $d^j(\mathbf{c})$ fully specified previously, the observed flux u_n can be recovered by

$$u_n = f_n + \epsilon_n = \nabla \mathbf{c}_n^T \mathbf{d}(\mathbf{c}_n) + \epsilon_n, \tag{14}$$

where we use the noise term ϵ_n to capture the model inadequacy, uncertainty, and noise as a Gaussian distribution, $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$, for the observed flux; f_n denotes the unknown true flux.

Eventually, the last piece of this work is the the estimation of the posterior of all hyperparameters $\Theta = \{\theta_j\}_{j=1}^{M}, B = \{\beta_j\}_{j=1}^{M}, Z = \{\mathbf{Z}_j\}_{j=1}^{M}, H = \{\mathbf{h}_j\}_{j=1}^{M}$, and σ. Although MCMC can be directly implemented to compute all model parameter posteriors, the computational time is overwhelming considering the large number of hyperparameters and the efficiency of an MCMC procedure. Instead, we opt for the maximum a posterior (MAP) approach. The log posterior decomposes as the log likelihood and the prior information,

$$\underset{\Theta,B,Z,H,\sigma}{\text{argmax}} \left(\mathcal{L}(\Theta, B, Z, H, \sigma) + \log p(B)\right), \tag{15}$$

where $\mathcal{L}(\Theta, B, Z, H, \sigma) = \log p(\mathbf{u})$ is the log likelihood of our model, which can be computed by comparing the predicted flux $\mathbf{f}$ and the observed flux $\mathbf{u}$. More specifically, the log marginal likelihood can be computed by

$$\log p(\mathbf{u}) = \log \int p(\mathbf{u}|\mathbf{f}) p(\mathbf{f}) d\mathbf{f}. \tag{16}$$

To complete the integration in Equation (16), we first notice that

$$p(\mathbf{u}|\mathbf{f}) = \mathcal{N}(\mathbf{u}|\mathbf{f}, \sigma^2 \mathbf{I}) \tag{17}$$

is simply a Gaussian; $p(\mathbf{f})$ is a mixture of M Gaussians, which is also Gaussian because

$$\begin{aligned} p(\mathbf{f}) &= \sum_{j=1}^{M} \nabla \mathbf{c}^T \mathbf{d}^j(\mathbf{c}) = \sum_{j=1}^{M} \nabla \mathbf{c}^T \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{Q}_j) \\ &= \sum_{j=1}^{M} \mathcal{N}\left(\nabla \mathbf{c}^j \odot \boldsymbol{\mu}_j, \nabla \mathbf{c}^j \mathbf{Q}_j \nabla (\mathbf{c}^j)^T \right) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{Q}). \end{aligned} \tag{18}$$

In this equation, $\boldsymbol{\mu}_j = \beta_j \mathbf{1} + \mathbf{k}_j (\mathbf{K}_j)^{-1} (\mathbf{h}_j - \beta_j \mathbf{1})$ is the predicted interdiffusion expectations for j; $\mathbf{Q}_j = \widehat{\mathbf{K}}_j - \widetilde{\mathbf{K}}_j \mathbf{K}_j^{-1} \widetilde{\mathbf{K}}_j^T$ is the covariance matrix, with $[\widetilde{K}_j]_{ng} = k_j(\mathbf{c}_n, \mathbf{z}_{jg}) \in \mathbb{R}^{N \times G}$ being the covariance between $\mathbf{C}$ and $\mathbf{Z}_j$ and $[\widehat{K}_j]_{nn'} = k_j(\mathbf{c}_n, \mathbf{c}_{n'}) \in \mathbb{R}^{N \times N}$ being the covariance for $\mathbf{C}$. $\boldsymbol{\mu} = \sum_{i=1}^{M} \boldsymbol{\mu}_i \odot \nabla \mathbf{c}^i$ is the joint expectations; $\mathbf{Q} = \sum_{i=1}^{M} \nabla \mathbf{c}^j \mathbf{Q}_j \nabla (\mathbf{c}^j)^T$ is the joint covariance matrix. Substituting Equations (17) and (18) into Equation (16) to derive the joint log likelihood, we get,

$$\log p(\mathbf{u}) = -\frac{1}{2} (\mu - \mathbf{u})^T \left(\mathbf{Q} + \sigma^2 \mathbf{I} \right)^{-1} (\mu - \mathbf{u}) - \frac{1}{2} \log |\mathbf{Q} + \sigma^2 \mathbf{I}| - \frac{N}{2} \log(2\pi). \tag{19}$$

We can now use any optimization techniques, e.g., gradient descent, to finish the MAP optimization. Although the fully independent training conditional (FITC) approximation [33] can be used to force $\mathbf{Q}_j$ to be a diagonal matrix and thus to enable quick computations [32], due to the multiplier $\nabla \mathbf{c}^j$, $\mathbf{Q}$ is generally non-diagonal, and this computation acceleration will not work in our case. The main computation for the joint likelihood (19) is the inverse of joint covariance matrix $\left(\mathbf{Q} + \sigma^2 \mathbf{I} \right)^{-1}$ and it log determinant $\log |\mathbf{Q} + \sigma^2 \mathbf{I}|$. Using an LU decomposition trick [22], we can compute these two terms at time complexity $\mathcal{O}(n^3)$ and space complexity $\mathcal{O}(n^2)$. For the interdiffusion problem, most of the time we have $N \leq 100$ EPMA samples, making our method practically efficient.

4.6. Interdiffusion Coefficients Predictions

With all model parameters being optimized, we can derive the posterior of the diffusion coefficients for any concentration $\mathbf{c}_*$ as

$$\begin{aligned} d^j(\mathbf{c}_*) &= \mathcal{N}(\mu_*^j, v_*^j), \\ \mu_*^j &= \beta_j \mathbf{1} + (\mathbf{k}_j^*)^T (\mathbf{K}_j)^{-1} (\mathbf{h}_j - \beta_j \mathbf{1}), \\ v_*^j &= [K_j]_{**} - (\mathbf{k}_j^*)^T (\mathbf{K}_j)^{-1} \mathbf{k}_j^*, \end{aligned} \tag{20}$$

where $\mathbf{k}_j^* = \left(k_j\left([\mathbf{c}_*, \mathbf{z}_{j1} \right), \cdots, k_j\left(\mathbf{c}_*, \mathbf{z}_{jG} \right) \right)^T$ is the covariance between $\mathbf{c}_*$ and the other ghost coefficient locations $\mathbf{z}_{jg}$ and $[K_j]_{**} = k_j(\mathbf{c}_*, \mathbf{c}_*)$. The derivation details are shown in the Appendix A for clarity.

5. Results

In practical experiments, the interdiffusion coefficients are unknown and uncontrollable, leading to difficulties for unbiased evaluations. Thus, we first assess InfPolyn on numerical examples of ternary ($M = 2$) and quaternary ($M = 3$) systems. To imitate a real system but not to lose generality, we use polynomial and exponential functions to construct the interdiffusion coefficient functions. To give an example, the fourth-order polynomial function in a two-component system is represented as

$$D_{ij}(c^1, c^2) = a_{ij}^0 + \sum_{m=1}^{2} a_{ij}^{m,1} c^m + \sum_{m=1}^{2} a_{ij}^{m,2} (c^m)^2 + \sum_{m=1}^{2} a_{ij}^{m,3} (c^m)^3 + \sum_{m=1}^{2} a_{ij}^{m,4} (c^m)^4, \tag{21}$$

where for each coefficient in the polynomial $a_{ij}^{t,r}$, the superscript r represents the degree of polynomial and the value of them are generated independently from uniform distributions $\mathcal{U}(0,1)$. We put constraints on the high order terms to prevent the diffusion coefficients from increasing/decreasing drastically with the concentrations $\mathbf{c}$; the diffusion matrix is considered symmetric to ensure numerical stability for the diffusion simulations. Note that this symmetric structure prior information is not injected into InfPolyn or other competing models. For the ternary system, the initial conditions for the forward simulation are

$$c^1(t = 0, x) = 0.6 \cdot \mathbb{1}(0.5 - x) \tag{22}$$

$$c^2(t = 0, x) = 0.4 \cdot \mathbb{1}(x - 0.5), \tag{23}$$

where $\mathbb{1}(z)$ denotes the Heaviside step function, which equals to 0 when $z < 0$ and equals to 1 when $z \geq 0$. Similarly, for the quaternary system, we defined the initial condition as

$$c^1(t = 0, x) = 0.6 \cdot \mathbb{1}(0.5 - x) \tag{24}$$

$$c^2(t = 0, x) = 0.25 \cdot \mathbb{1}(x - 0.5) \tag{25}$$

$$c^3(t = 0, x) = 0.15 \cdot \mathbb{1}(x - 0.5). \tag{26}$$

With the defined initial condition and the interdiffusion coefficient functions, we use a finite difference (DF) diffusion forward solver to simulate a diffusion process until the terminal time and obtain the terminal concentration profile $\mathbf{C}$. To remain numerically stable and accurate, we use a second-order central difference for space and a fourth-order Runge–Kutta for time. The forward simulation solver uses a spatial step $\Delta x = 0.000625$, which suggests 1601 grinds points on the space domain $[0, 1]$, the terminal time is set to 10^4 Δt.

We then take equally spaced samples from the terminal concentration profile to mimic the EPMA process to provide the terminal concentration profile $\mathbf{C}$. Unless stated otherwise, the terminal concentration profile consists of 40 samples. Since we are concerned with the center areas where the diffusion process is significant, the EPMA samples are limited in the range of $[0.44, 0.56]$ in order to avoid numerical error closed to the boundaries for all Boltzmann–Matano method. All variables are considered dimensionless in the experiments. To evaluate the performance for different methods, we follow Cheng et al. [13] and use the relative error (RE),

$$\mathrm{RE}_{ij}(\mathbf{c}) = \frac{\tilde{D}_{ij}(\mathbf{c}) - D_{ij}(\mathbf{c})}{D_{ij}(\mathbf{c})}, \tag{27}$$

where $\tilde{D}_{ij}(\mathbf{c})$ and $D_{ij}(\mathbf{c})$ are the predicted and truth interdiffusion coefficients for concentration $\mathbf{c}$, respectively. As a Boltzmann–Matano numerical inversion-based method, InfPolyn are compared with the other SOTA Boltzmann–Matano numerical inversion-based methods, i.e., the polynomial interdiffusion methods [13] with 3rd and 4th orders of the polynomial, the compress sensing approach [16] combined with 4th-order polynomial (high order model enough to capture the subtle changes), and the L^2 regularization approach, which replaces the L^1 penalty term in the work of Qin et al. [16] with an L^2 penalty term, combined with a 4th-order polynomial function.

5.1. Case Study 1: Polynomial Diffusion Coefficients

In this case study, we assess InfPolyn in a ternary system and a quaternary with 4th-order polynomial interdiffusion coefficients:

$$D_{ij}(\mathbf{c}) = a_{ij}^{m,0} + \sum_{m=1}^{M}\sum_{r=1}^{4} a_{ij}^{m,r}(c^m)^r, \tag{28}$$

where each coefficients $a_{ij}^{m,r}$ are randomly generated using independent uniform distributions. To ensure the symmetrical structure of matrix $\mathbf{A}^{m,r}$, we force $a_{ij}^{m,r} = a_{ji}^{m,r}$ by taking their average. In a general interdiffusion process, the interdiffusion coefficients are supposed to be smooth and close to constants, which also prevents instability in the numerical forward solver. To ensure this prior knowledge, we constrain the polynomial coefficients by $a_{ij}^{m,r} \sim \mathcal{U}(0,1)10^{(r-5)}$. The particularly used values are shown in the Appendix A. The REs for $x \in [0.4, 0.6]$ for the ternary and the quaternary system are shown in Figures 1 and 2. We omit areas outside $[0.4, 0.6]$ because the REs are just extended flat lines without interesting information. As expected, the 4th order polynomial method has a strong model capacity and it can thus achieve few lowest REs at as is shown in some figures within Figures 1 and 2. However, if we look at the whole area of interest, the overall performance is the worst among all methods. In particular, due to the overfitting issue, the 4th order polynomial method shows a highly fluctuational performance, which is highly depreciated for real applications. It is not surprising to learn that the 3rd-order polynomial approach shows slightly fewer fluctuations but also fewer lowest REs. This is indeed the aforementioned dilemma of model selection for the polynomial based methods. Similar to results shown in [16], adding a regularization term of L^1 can ease the overfitting issue and greatly overcome the performance fluctuation issue in both Figures 1 and 2. Unfortunately, the improvement comes with the price of low model capacity, leading to a rather flat-fitting RE. The 4th-order polynomial method combined with a L^2 regularization term shows a similar improvement. It is, however, difficult to tell which regularization terms are better. The L^1 regularization works better with the ternary system in Figure 1, whereas the L^2 approach outperforms the L^1 with a large margin in most cases of Figure 2. The inconsistency of performance for the L^1 and L^2 regularization approaches certainly hinders their applications for practical problems. In contrast, guided by the correct priors and benefited from the nonparametric nature, InfPolyn shows a consistent and accurate fitting and outperforms the competitors by a significant margin. Thanks to the model flexibility of InfPolyn, it can capture the dramatic changes in the center while maintaining a good fitting in the other flat areas. In all cases, InfPolyn can not only remain stable (indicated by a smooth RE curve) but also achieve the lowest REs in most areas. Furthermore, note that the diagonal interdiffusion coefficients in general show a lower relative error. This is because, in the simulation setting, the diagonal interdiffusion coefficients play a dominant role in the diffusion process. For the non-diagonal interdiffusion coefficients, the REs are amplified by being divided by smaller true interdiffusion coefficients.

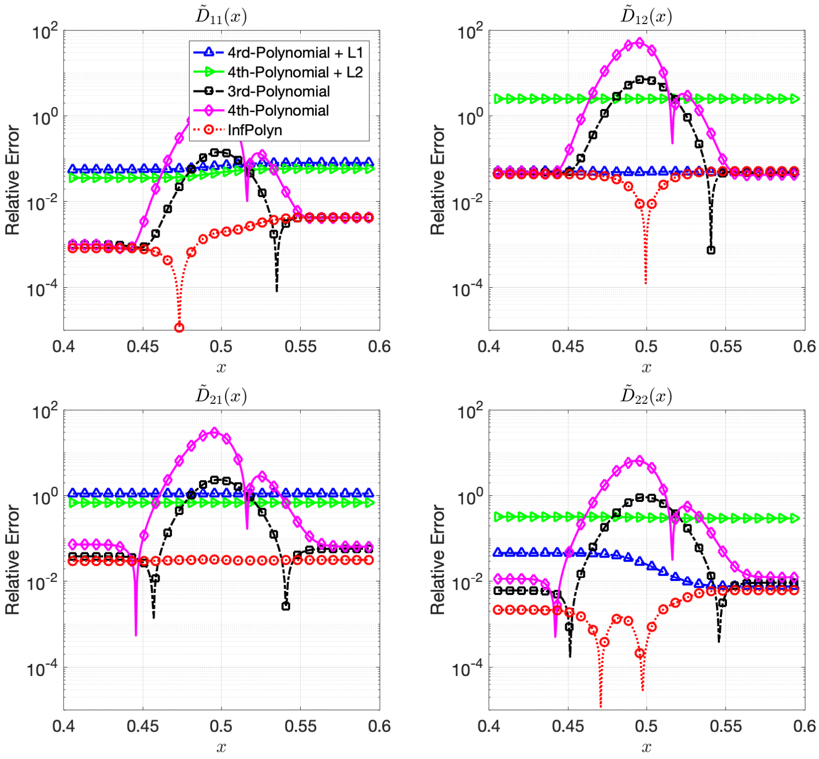

Figure 1. The relative errors (REs) of predictive diffusion coefficients $\tilde{D}_{ij}(\mathbf{c}(x))$ in the center areas $x \in [0.4, 0.6]$ for the evaluated methods in a random ternary system.

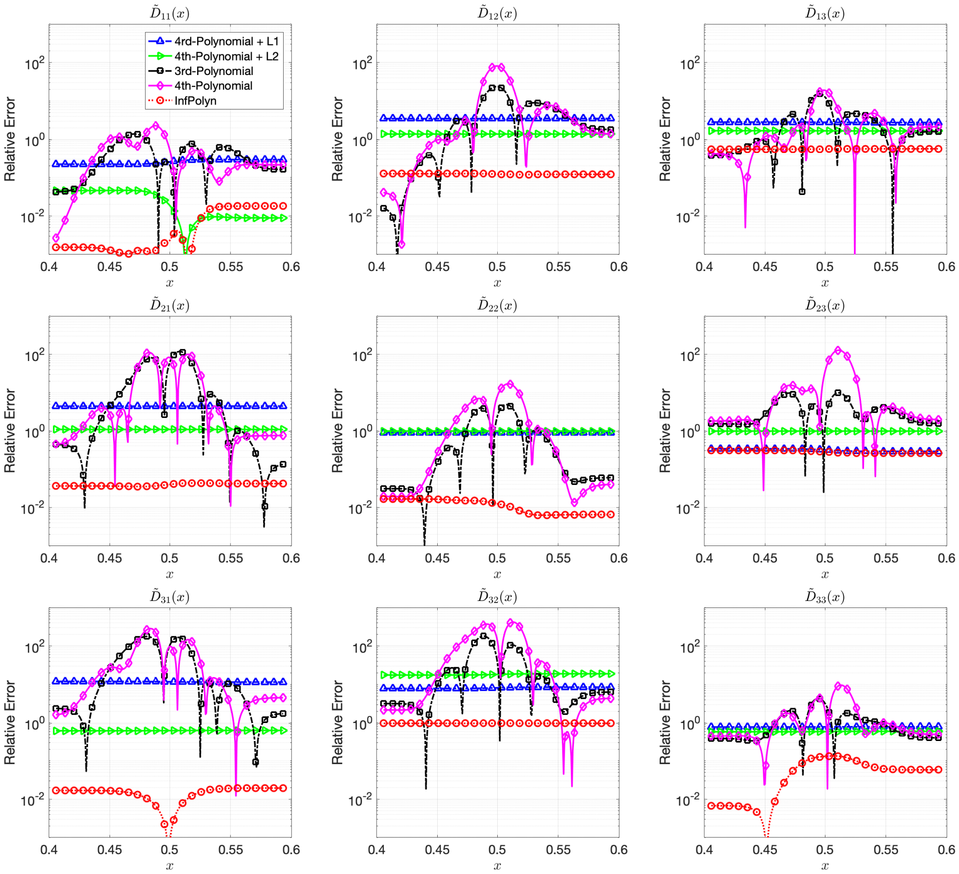

Figure 2. The relative errors (REs) of predictive diffusion coefficients $\tilde{D}_{ij}(\mathbf{c}(x))$ in the center areas $x \in [0.4, 0.6]$ for the evaluated methods in a quadternary system.

5.2. Case Study 2: Exponential Diffusion Coefficients

In general, the diffusion coefficients can be highly complex that they are not in polynomial forms. To imitate such challenging situations, in this case study, we assess InfPolyn in ternary and quaternary systems with the following interdiffusion coefficient that combines an exponential term and a sinusoidal term,

$$D_{ij}(\mathbf{c}) = a_{ij}^{0} + \sum_{m=1}^{M} a_{ij}^{m,1} \exp(-c^m) - \sum_{m=1}^{M} a_{ij}^{m,2} \cos(c^m), \tag{29}$$

where the functional coefficients $a_{ij}^{m,r}$ are similarly sampled from different uniform distributions, i.e., $a_{ij}^{0} \sim \mathcal{U}(0,1) \times 10^{-5}$, $a_{ij}^{m,1} \sim \mathcal{U}(0,1) \times 10^{-6}$, and $a_{ij}^{m,2} \sim \mathcal{U}(0,1) \times 10^{-6}$. Similarly, to ensure the forward diffusion stability, we use the previous approach to ensure the symmetrical structure of matrices $\mathbf{A}^0$, $\mathbf{A}^{m,1}$, and $\mathbf{A}^{m,2}$. The used exact values of the functional coefficients are shown in Appendix A. The model performances measured by REs are shown in Figures 3 and 4.

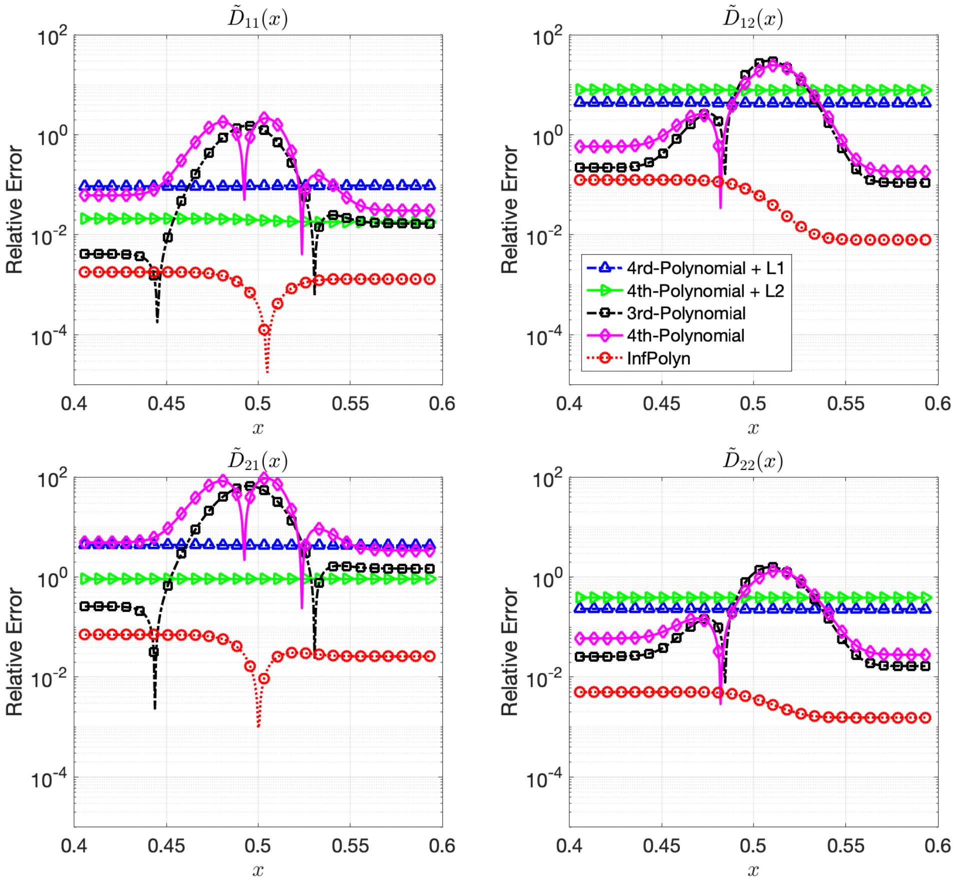

Figure 3. The relative errors (REs) of predictive diffusion coefficients $\tilde{D}_{ij}(\mathbf{c}(x))$ in the center areas $x \in [0.4, 0.6]$ for the evaluated methods in a ternary system.

In this case study, the 3rd-order polynomial slightly outperforms the 4th-order polynomial approaches in most cases in both Figures 3 and 4. Nevertheless, the performance of both 3rd-order and 4th-order polynomial approaches are depreciated due to the fluctuation across the domain. Furthermore, note that REs for the polynomial approaches in Figure 4 are flat and smooth, indicating that a rich model capacity does not necessarily lead to performance fluctuations in all cases. The L^1 and L^2 regularization combined with 4th-order polynomial degenerate the model performance rather than improving them in many cases in Figure 4. This shows evidence that inappropriate implicit prior assumptions caused by the L^1 and L^2 regularization terms can hurt model performance. It might be possible to circumvent this issue by adjusting the penalty weight. However, this will create a new issue of how to properly decide the value of the penalty weight, taking us back to the dilemma of model selections. In contrast, InfPolyn shows a consistent and accurate performance; it outperforms the competitors by a large margin for all cases except for $\tilde{D}_{31}$ of the quaternary system in the left area in Figure 4. We would also like to point out that many methods actually fail the quaternary system in Figure 4 as their REs are larger than 1, meaning a total prediction failure.

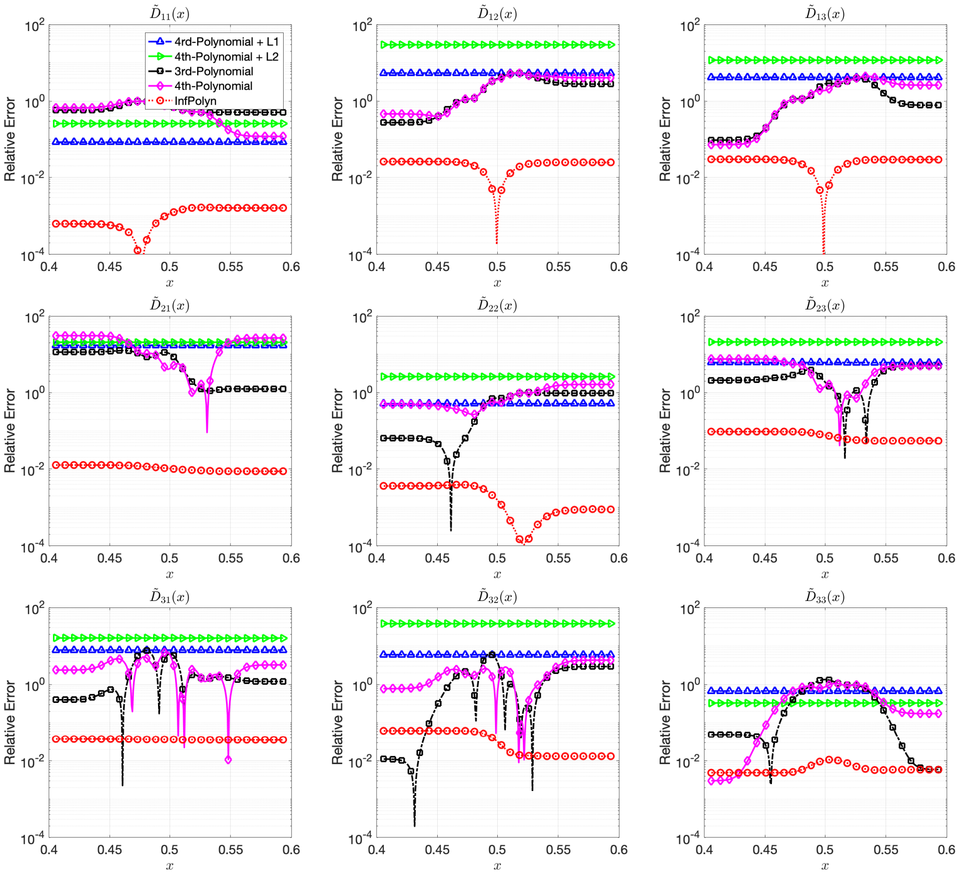

Figure 4. The relative errors (REs) of predictive diffusion coefficients $\tilde{D}_{ij}(\mathbf{c}(x))$ in the center areas $x \in [0.4, 0.6]$ for the evaluated methods in a quadternary system.

5.3. Case Study 3: Uncertainty Quantification Analysis

Finally, to assess the consistency of InfPolyn, we conducted a ternary system experiment in Case Study 1 based on five distinct random polynomial coefficient sets, which assemble five different diffusion coefficients, and show the performance statistics. To also investigate the influence of the number of the EPMA samples, we ran each experiment with $\{20, 30, 40, 50\}$ EPMA samples. The minimum number of the EPMA samples was 20 because the 4th-order polynomial has 18 coefficients and thus requires at least 18 EPMA samples to work. For each experiment with the given EPMA samples, the model performance was evaluated by average relative error (ARE),

$$\text{ARE}_{ij} = \frac{\int_{\mathcal{X}} \text{RE}_{ij}(\mathbf{c}(x))dx}{\int_{\mathcal{X}} dx} \tag{30}$$

where $\mathcal{X}$ indicates the whole spatial domain. We show the statistics of ARE_{11} and ARE_{22} over the five different diffusion coefficients in Figure 5 using the Tukey box plot. The distinct fact we immediately see is the superiority of InfPolyn compared to the competitors in terms of accuracy and consistency. We then notice that the performance does not improve gradually with the increasing number of EPMA samples for all methods except for the

4th-order polynomial. We believe that each method can already approach reasonable diffusion coefficients (by minimizing the loss function) with only 20 EPMA samples. In this case, more samples will not bring improvement, whereas the performance can fluctuate with different EPMA concentration profiles. Comparing the fluctuations, InfPolyn shows a modest level of changes, whereas the most unstable one is the 4th-order polynomial with L^2 regularization. The most stable method for both $\tilde{D}_{11}$ and $\tilde{D}_{22}$ is the 3rd order polynomial, which can indicate a lack of model capacity or an underfitting issue. The only exception of performance improvement is the 4th order polynomial, which improves with more EPMA samples. This is a clear sign of overfitting, which can be addressed by introducing more training data. This explains the overfitting phenomena we previously encountered in Case Studies 1 and 2. Will the performance keep improving and outperform InfPolyn with the trend shown in Figure 5? It may happen with more than 200 EPMA samples, which becomes infeasible in practice. Furthermore, the decreasing trend should slowly disappear at some point, which is already happening for $\tilde{D}_{22}$.

It is also noticeable that the L^1 and L^2 regularization techniques indeed can improve the performance of a 4th-order polynomial by a large margin for all cases with a different number of EPMA samples, which is consistent with the finding in [16].

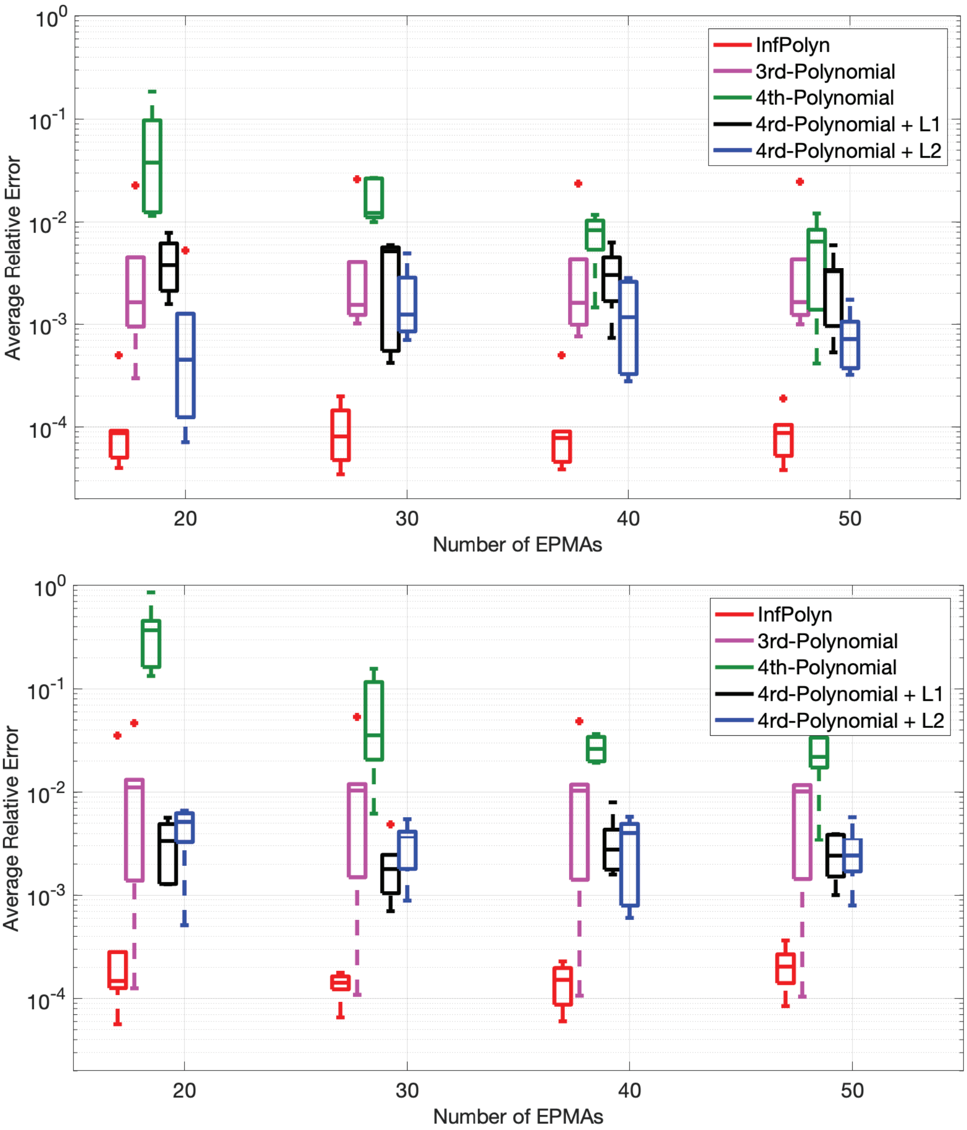

Figure 5. The Tukey box plot of average relative error of $\tilde{D}_{11}$ (**top**) and $\tilde{D}_{22}$ (**bottom**) based on computation using concentration profile consisting $\{20, 30, 40, 50\}$ EMPA samples.

5.4. Case Study 4: Experiment Verification

To present the practical applicability of InfPolyn, we then apply it to the reproduction of the interdiffusion flux from experiment data of the Mg-Al, Mg-Al-Zn, and Mg-Al-Zn-Cu systems collected from the previous literature [13]. These experimental data include composition profiles of the annealed diffusion couples of Mg-Al at 781 K for 36,960 s, Mg-Al-Zn at 868 K for 5400 s, and Mg-Al-Zn-Cu at 755 K for 75,530 s. Since the experimental measurements are taken non-uniformly on the spatial domain for all of the components, they are reprocessed with local polynomials interpolation techniques to provide values on a uniform grid, which is the common preprocessing for the Matano-based approaches. The derivative and integral terms in the Matano equation are then obtained. Given all the preprocessed data as inputs, we then randomly take all of the samples, half of the samples, and a quarter of the samples from the diffusion systems to test the robustness of the testing methods. As shown in Figure 6, the curves for all three cases computed by InfPolyn fit well with the experimental data, which lie in the 95% confident areas, indicating a good uncertainty quantification for the predictions. As for the half size and the quarter size training data, the left areas induce oscillations in some intervals. However, InfPolyn still captures the major tendency of the fluxes with slightly increasing uncertainty.

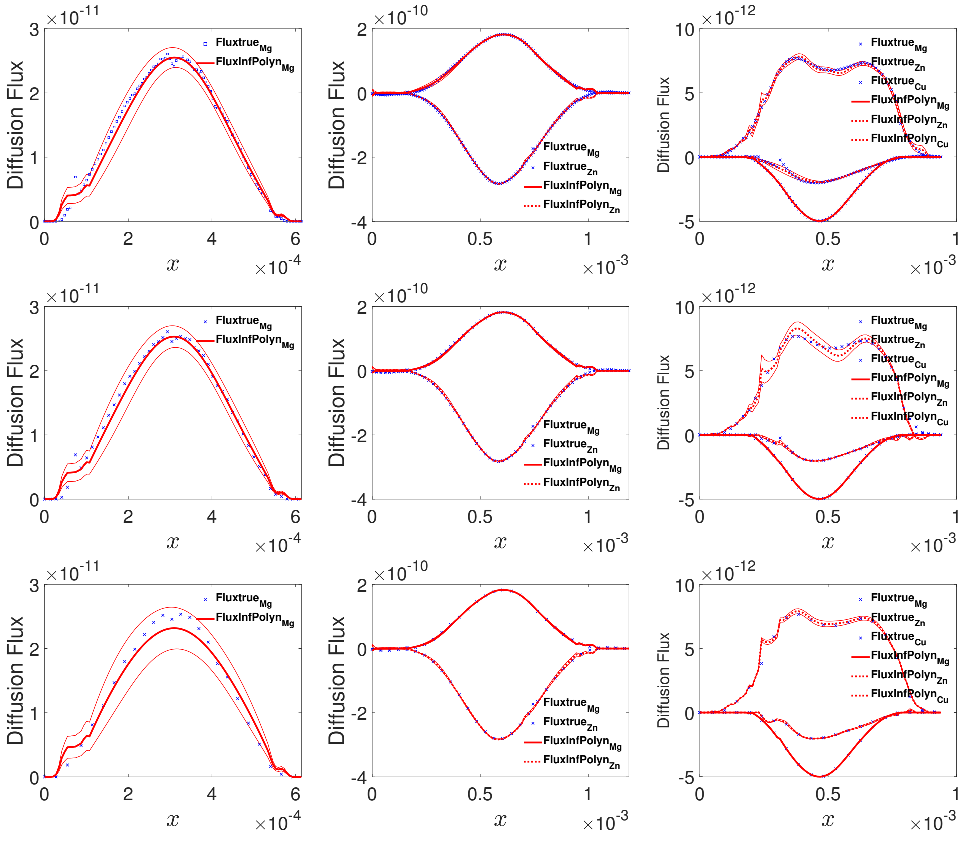

Figure 6. The actual and predicted diffusion fluxes for the Mg-Al, Mg-AL-Zn, and Mg-Al-Zn-Cu system (from left to right columns) using 100%, 50%, and 25% of all available samples (from top to bottom rows).

6. Conclusions

In this paper, we propose InfPolyn, a novel nonparametric Bayesian framework to estimate the interdiffusivity coefficients and demonstrate its superiority in terms of accuracy and consistency by combining it with the numerical inverse Boltzmann–Matano method [13]. This also becomes the limitation of InfPolyn because the numerical inverse Boltzmann–Matano method has certain limitations. For instance, it cannot generalize to a wide variety of complicated 2D diffusion processes and complex engineering interdiffusion scenarios, e.g., in semiconductors. Nevertheless, InfPolyn can be combined with other methods (such as the forward-simulation approach [10]) to fulfill its potential in the estimations of interdiffusion coefficients. This is outside the scope of this paper and we thus leave it for the future work.

The main novelty of our work is the nonparametric Bayesian framework that allows automatic model selections (resistant to overfitting and underfitting) and meaningful prior knowledge injections. The problem of recovering interdiffusion predictions from concentrations is an ill-posed inverse problem. Thus, the injection of proper priors is a necessary way to recover the ground-truth diffusion coefficients. Unlike methods such as [16] that impose nonphysical priors, our method provides an easy way to inject intuitive priors; e.g., the diagonal diffusion coefficients normally plays a dominant role in an interdiffusion process.

Author Contributions: Conceptualization, P.W. and K.C.; methodology, W.W.X.; validation, M.C.; formal analysis, M.C. and W.W.X.; investigation, W.W.X. and M.C.; resources, W.Z. and P.W.; data curation, K.C.; writing—original draft preparation, W.W.X.; writing—review and editing, W.W.X., M.C., K.C. and P.W.; visualization, W.W.X.; supervision, K.C., W.Z. and P.W.; funding acquisition, P.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China grant number 2017YFB0701700 and 2018YFB0703902.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data is in the Appendix A, numerical experiments, and the reference.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. A Gaussian Processs and Its Predicted Posterior

Consider training points $y_i = \eta(\boldsymbol{\xi}_i), i = 1, \ldots, M$ and design points $\boldsymbol{\xi}_i$. In a GP model, we place a prior distribution over $\eta(\boldsymbol{\xi})$ indexed by $\boldsymbol{\xi}$:

$$\eta(\boldsymbol{\xi})|\boldsymbol{\theta} \sim \mathcal{GP}\big(m(\boldsymbol{\xi}), c(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta})\big) \tag{A1}$$

with mean and covariance functions:

$$m_0(\boldsymbol{\xi}) = \mathbb{E}[\eta(\boldsymbol{\xi})], \quad c(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}) = \mathbb{E}[(\eta(\boldsymbol{\xi}) - m_0(\boldsymbol{\xi}))(\eta(\boldsymbol{\xi}') - m_0(\boldsymbol{\xi}'))] \tag{A2}$$

in which $\mathbb{E}[\cdot]$ is the expectation operator. The hyperparameters $\boldsymbol{\theta}$ are estimated during the learning process. The mean function can be assumed to be a identical constant, $m_0(\boldsymbol{\xi}) \equiv \mu$, by virtue of centering the data. Alternative choices are possible, e.g., a linear function of $\boldsymbol{\xi}$, but rarely adopted unless a priori information on the form of the function is available. The covariance function can take many forms, the most common being the automatic relevance determinant (ARD) kernel:

$$c(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}) = \theta_0 \exp\Big(-(\boldsymbol{\xi} - \boldsymbol{\xi}')^T \mathrm{diag}(\theta_1, \ldots, \theta_l)(\boldsymbol{\xi} - \boldsymbol{\xi}')\Big) \tag{A3}$$

The hyperparameters $\boldsymbol{\theta} = (\theta_0, \ldots, \theta_l)^T$. $\theta_1^{-1}, \ldots, \theta_l^{-1}$ in this case are called the square correlation lengths. For any fixed $\boldsymbol{\xi}$, $\eta(\boldsymbol{\xi})$ is a random variable. A collection of values $\eta(\boldsymbol{\xi}_i)$, $i = 1, \ldots, M$, on the other hand, is a partial realization of the GP. Realizations of the GP are deterministic functions of $\boldsymbol{\xi}$. The main property of GPs is that the joint distribution of $\eta(\boldsymbol{\xi}_i), i = 1, \ldots, M$, is multivariate Gaussian.

Starting from the prior (A1) and using the available data, we obtain a posterior GP distribution conditional on $\boldsymbol{\theta}$, with new mean and covariance functions. Letting $\mathbf{t} = (y_1, \ldots, y_M)^T$, the likelihood of the data (given $\boldsymbol{\theta}$) is $p(\mathbf{t}|\boldsymbol{\theta}) = \mathcal{N}(\mu\mathbf{1}, \mathbf{C}(\boldsymbol{\theta}))$. Here, $\mathcal{N}(\cdot, \cdot)$ denotes a normal distribution with mean $\mathbf{0}$ and covariance matrix $\mathbf{C}(\boldsymbol{\theta}) = [C_{ij}]$, in which

$C_{ij} = c(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j|\boldsymbol{\theta}), i, j = 1, \ldots, M$. The joint distribution of $\mathbf{t}$ and $\eta(\boldsymbol{\xi})$ (for a test input $\boldsymbol{\xi}$) has the distribution $p(\eta(\boldsymbol{\xi}), \mathbf{t}|\boldsymbol{\theta}) = \mathcal{N}(\mu\mathbf{1}, \mathbf{C}(\boldsymbol{\theta})')$, where:

$$\mathbf{C}'(\boldsymbol{\theta}) = \left[\begin{array}{c|c} \mathbf{C}(\boldsymbol{\theta}) & \mathbf{c}(\boldsymbol{\xi}) \\ \hline \mathbf{c}(\boldsymbol{\xi})^T & c(\boldsymbol{\xi}, \boldsymbol{\xi}|\boldsymbol{\theta}) \end{array}\right] \tag{A4}$$

in which $\mathbf{c}(\boldsymbol{\xi}) = (c(\boldsymbol{\xi}_1, \boldsymbol{\xi}|\boldsymbol{\theta}), \ldots, c(\boldsymbol{\xi}_M, \boldsymbol{\xi}|\boldsymbol{\theta}))^T$. Conditioning on $\mathbf{t}$ provides the conditional predictive distribution at $\boldsymbol{\xi}$ [22]:

$$\begin{aligned} \eta(\boldsymbol{\xi})|\mathbf{t}, \boldsymbol{\theta} &\sim \mathcal{GP}(m'(\boldsymbol{\xi}|\boldsymbol{\theta}), c'(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta})) \\ m'(\boldsymbol{\xi}|\boldsymbol{\theta}) &= \mu\mathbf{1} + \mathbf{c}(\boldsymbol{\xi})^T\mathbf{C}(\boldsymbol{\theta})^{-1}(\mathbf{t} - \mu\mathbf{1}) \\ c'(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}) &= c(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}) - \mathbf{c}(\boldsymbol{\xi})^T\mathbf{C}(\boldsymbol{\theta})^{-1}\mathbf{c}(\boldsymbol{\xi}') \end{aligned} \tag{A5}$$

The expected value $\mathbb{E}[\eta(\boldsymbol{\xi})]$ is given by $m'(\boldsymbol{\xi}|\boldsymbol{\theta})$, and $c'(\boldsymbol{\xi}, \boldsymbol{\xi}|\boldsymbol{\theta})$ is the predictive variance. The hyperparameters $\boldsymbol{\theta}$ are normally obtained from point estimates [34,35]. The maximum likelihood estimate (MLE), for example, is found by maximizing the log of the likelihood:

$$\boldsymbol{\theta}_{MLE} = \text{arg max}_{\boldsymbol{\theta}}\left(-\frac{M}{2}\log(2\pi) - \frac{1}{2}(\mathbf{t} - \mu\mathbf{1})^T\mathbf{t}^T\mathbf{C}(\boldsymbol{\theta})^{-1}(\mathbf{t} - \mu\mathbf{1}) - \frac{1}{2}\ln|\mathbf{C}(\boldsymbol{\theta})|\right). \tag{A6}$$

Appendix A.1. Solving a Ternary System Using Boltzmann–Matano Inverse Method

Consider a ternary system where we take EPMA samples at four random locations, $[c^1(x_n), c^2(x_n)], (n = 1, \ldots, 4)$, on the "true" concentration-distance curve at a terminal time. By substituting them into the Boltzmann–Matano equations, we have

$$\begin{bmatrix} \frac{\partial c_1}{\partial x}|_{x_1} & 0 & \frac{\partial c_2}{\partial x}|_{x_1} & 0 \\ 0 & \frac{\partial c_1}{\partial x}|_{x_1} & 0 & \frac{\partial c_2}{\partial x}|_{x_1} \\ \frac{\partial c_1}{\partial x}|_{x_2} & 0 & \frac{\partial c_2}{\partial x}|_{x_2} & 0 \\ 0 & \frac{\partial c_1}{\partial x}|_{x_2} & 0 & \frac{\partial c_2}{\partial x}|_{x_2} \\ \frac{\partial c_1}{\partial x}|_{x_3} & 0 & \frac{\partial c_2}{\partial x}|_{x_3} & 0 \\ 0 & \frac{\partial c_1}{\partial x}|_{x_3} & 0 & \frac{\partial c_2}{\partial x}|_{x_3} \\ \frac{\partial c_1}{\partial x}|_{x_4} & 0 & \frac{\partial c_2}{\partial x}|_{x_4} & 0 \\ 0 & \frac{\partial c_1}{\partial x}|_{x_4} & 0 & \frac{\partial c_2}{\partial x}|_{x_4} \end{bmatrix} \begin{bmatrix} D_{11} \\ D_{21} \\ D_{12} \\ D_{22} \end{bmatrix} = \begin{bmatrix} u_1|_{x_1} \\ u_2|_{x_1} \\ u_1|_{x_2} \\ u_2|_{x_2} \\ u_1|_{x_3} \\ u_2|_{x_3} \\ u_1|_{x_4} \\ u_2|_{x_4} \end{bmatrix}. \tag{A7}$$

We assume second-order polynomials for the functional relationships between diffusion coefficients and concentrations,

$$\begin{bmatrix} D_{11} \\ D_{21} \\ D_{12} \\ D_{22} \end{bmatrix} = \begin{bmatrix} \alpha_{11}^{(0)} + \alpha_{11}^{(1)}c_1 + \alpha_{11}^{(2)}c_2 + \alpha_{11}^{(3)}c_1^2 + \alpha_{11}^{(4)}c_2^2 \\ \alpha_{21}^{(0)} + \alpha_{21}^{(1)}c_1 + \alpha_{21}^{(2)}c_2 + \alpha_{21}^{(3)}c_1^2 + \alpha_{21}^{(4)}c_2^2 \\ \alpha_{12}^{(0)} + \alpha_{12}^{(1)}c_1 + \alpha_{12}^{(2)}c_2 + \alpha_{12}^{(3)}c_1^2 + \alpha_{12}^{(4)}c_2^2 \\ \alpha_{22}^{(0)} + \alpha_{22}^{(1)}c_1 + \alpha_{22}^{(2)}c_2 + \alpha_{22}^{(3)}c_1^2 + \alpha_{22}^{(4)}c_2^2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\phi} \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix}, \tag{A8}$$

where

$$\boldsymbol{\phi} = \begin{bmatrix} \mathbf{I} & c^1\mathbf{I} & c^2\mathbf{I} & (c^1)^2\mathbf{I} & (c^2)_2\mathbf{I} \end{bmatrix} \tag{A9}$$

$$= \begin{bmatrix} 1 & 0 & c^1 & 0 & c^2 & 0 & (c^1)^2 & 0 & (c^2)^2 & 0 \\ 0 & 1 & 0 & c^1 & 0 & c^2 & 0 & (c^1)^2 & 0 & (c^2)^2 \end{bmatrix} \tag{A10}$$

$$\mathbf{a}_i = \begin{bmatrix} \alpha_{1i}^{(0)} & \alpha_{2i}^{(0)} & \alpha_{1i}^{(1)} & \alpha_{2i}^{(1)} & \alpha_{1i}^{(2)} & \alpha_{2i}^{(2)} & \alpha_{1i}^{(3)} & \alpha_{2i}^{(3)} & \alpha_{1i}^{(4)} & \alpha_{2i}^{(4)} \end{bmatrix}^{\mathrm{T}} \quad i = 1, 2. \tag{A11}$$

We can now simply solve the linear system of equations to obtain the polynomial coefficients (and thus the diffusion coefficients as functions of the polynomials). The procedure is the same with more EPMA samples or higher orders of polynomials. The only difference is that we may solve an overdetermined system with the criteria of minimizing the L^2 loss.

Appendix A.2. Experimental Details

Tables A1 and A2 show the experimental setting of functional coefficients in Case Study 1; Tables A3 and A4 show the experimental setting of functional coefficients in Case Study 2.

Table A1. The polynomial coefficients in the random ternary interdiffusion system, where a_{ij} represents the entries in the coefficient matrix on position $\{i, j\}, i = 1,\ 2,\ j = 1,\ 2$.

A	a_{11}	a_{21}	a_{22}
A_0	6.05×10^{-5}	4.81×10^{-6}	5.12×10^{-5}
A_1^1	3.08×10^{-6}	4.18×10^{-7}	3.28×10^{-6}
A_2^1	1.82×10^{-6}	6.42×10^{-7}	2.86×10^{-6}
A_1^2	8.83×10^{-7}	6.07×10^{-8}	1.07×10^{-7}
A_2^2	2.96×10^{-7}	4.21×10^{-8}	9.63×10^{-7}
A_1^3	4.09×10^{-8}	2.82×10^{-9}	1.26×10^{-8}
A_2^3	1.19×10^{-8}	7.29×10^{-9}	1.10×10^{-8}
A_1^4	1.26×10^{-8}	5.02×10^{-10}	1.66×10^{-9}
A_2^4	6.04×10^{-9}	6.34×10^{-10}	1.09×10^{-8}

Table A2. The polynomial coefficients in the random quaternary interdiffusion system, where a_{ij} represents the entries in the coefficient matrix on position $\{i, j\}, i = 1,\ 2,\ 3;\ j = 1,\ 2\ ,3$.

A	a_{11}	a_{12}	a_{13}	a_{22}	a_{23}	a_{33}
A_0	2.03×10^{-5}	6.50×10^{-6}	5.15×10^{-6}	7.13×10^{-5}	3.58×10^{-7}	3.27×10^{-5}
A_1^1	9.15×10^{-6}	3.13×10^{-7}	6.87×10^{-7}	9.46×10^{-6}	1.35×10^{-7}	1.76×10^{-6}
A_2^1	9.56×10^{-6}	8.08×10^{-7}	3.01×10^{-7}	1.61×10^{-6}	4.89×10^{-7}	1.18×10^{-6}
A_3^1	2.01×10^{-6}	9.05×10^{-7}	2.47×10^{-7}	5.48×10^{-6}	7.41×10^{-7}	2.95×10^{-6}
A_1^2	2.61×10^{-7}	7.28×10^{-8}	6.95×10^{-8}	2.18×10^{-7}	5.91×10^{-8}	1.58×10^{-7}
A_2^2	5.90×10^{-7}	2.23×10^{-8}	4.55×10^{-8}	7.08×10^{-7}	5.79×10^{-8}	3.94×10^{-7}
A_3^2	2.17×10^{-9}	2.96×10^{-8}	5.60×10^{-8}	8.11×10^{-7}	3.82×10^{-8}	6.70×10^{-8}
A_1^3	9.00×10^{-8}	2.45×10^{-9}	2.52×10^{-10}	9.00×10^{-8}	5.90×10^{-9}	2.48×10^{-8}
A_2^3	2.70×10^{-8}	2.57×10^{-9}	2.11×10^{-9}	6.50×10^{-10}	6.48×10^{-9}	6.01×10^{-9}
A_3^3	8.04×10^{-8}	2.55×10^{-9}	3.11×10^{-9}	8.91×10^{-8}	2.76×10^{-9}	1.03×10^{-8}
A_1^4	4.82×10^{-9}	3.18×10^{-10}	1.87×10^{-10}	9.07×10^{-9}	2.45×10^{-10}	3.91×10^{-9}
A_2^4	6.58×10^{-9}	9.28×10^{-10}	2.44×10^{-10}	2.05×10^{-9}	6.07×10^{-10}	7.94×10^{-9}
A_3^4	9.38×10^{-9}	1.30×10^{-10}	9.50×10^{-10}	9.17×10^{-9}	6.05×10^{-10}	1.46×10^{-9}

Table A3. The table shows the setting for coefficient matrix of polynomials function and $A_i\ and\ A_i^{'}$ represents the entries in each coefficient matrix on position $\{i\}, i = 1,\ 2;\ j = 1,\ 2$.

A	a_{11}	a_{21}	a_{22}
A_0	6.15×10^{-5}	5.08×10^{-7}	5.46×10^{-5}
A_1	1.25×10^{-7}	3.77×10^{-7}	8.50×10^{-8}
A_2	9.09×10^{-7}	3.51×10^{-7}	9.08×10^{-7}
$A_1^{'}$	4.85×10^{-7}	2.85×10^{-8}	7.39×10^{-7}
$A_2^{'}$	4.67×10^{-7}	7.29×10^{-7}	8.27×10^{-7}

Table A4. The table shows the setting for coefficient matrix of exponential function and $A_i\ and\ A_i^{'}$ represents the entries in each coefficient matrix on position $\{i\}, i = 1,\ 2,\ ,3;\ j = 1,\ 2\ ,3$.

A	a_{11}	a_{12}	a_{13}	a_{22}	a_{23}	a_{33}
A_0	4.82×10^{-5}	4.61×10^{-6}	4.17×10^{-6}	4.26×10^{-5}	4.34×10^{-6}	6.27×10^{-5}
A_1	8.77×10^{-8}	8.40×10^{-8}	1.07×10^{-8}	9.65×10^{-8}	8.20×10^{-8}	2.62×10^{-8}
A_2	5.15×10^{-8}	8.39×10^{-9}	1.87×10^{-8}	9.39×10^{-8}	1.55×10^{-8}	9.78×10^{-8}
A_3	6.81×10^{-8}	2.67×10^{-8}	2.23×10^{-8}	7.68×10^{-9}	5.41×10^{-9}	3.78×10^{-8}
$A_1^{'}$	3.45×10^{-8}	3.79×10^{-8}	3.48×10^{-8}	3.11×10^{-9}	1.42×10^{-8}	5.00×10^{-8}
$A_2^{'}$	1.21×10^{-9}	3.53×10^{-8}	6.64×10^{-8}	7.51×10^{-8}	2.52×10^{-8}	5.45×10^{-8}
$A_3^{'}$	6.21×10^{-8}	3.77×10^{-8}	6.93×10^{-8}	1.23×10^{-9}	6.45×10^{-8}	1.75×10^{-8}

References

1. Boltzmann, L. Zur integration der diffusionsgleichung bei variabeln diffusionscoefficienten. *Ann. Phys.* **1894**, *53*, 959–964. [CrossRef]
2. Matano, C. On the Relation between Diffusion-Coefficients and Concentrations of Solid Metals. *Jpn. J. Phys.* **1933**, *8*, 109–113.
3. Kirkaldy, J.S.; Lane, J.E.; Mason, G.R. Diffusion in multicomponent metallic systems: VII. Solutions of the multicomponent diffusion equations with variable coefficients. *Can. J. Phys.* **1963**, *41*, 2174–2186. [CrossRef]
4. Chen, W.; Zhang, L.; Du, Y.; Tang, C.; Huang, B. A pragmatic method to determine the composition-dependent interdiffusivities in ternary systems by using a single diffusion couple. *Scr. Mater.* **2014**, *90–91*, 53–56. [CrossRef]
5. Dayananda, M.A.; Sohn, Y.H. A new analysis for the determination of ternary interdiffusion coefficients from a single diffusion couple. *Metall. Mater. Trans. A Phys. Metall. Mater. Sci.* **1999**, *30*, 535–543. [CrossRef]
6. Cermak, J.; Rothova, V. Concentration dependence of ternary interdiffusion coefficients in Ni3Al/Ni3Al–X couples with X= Cr, Fe, Nb and Ti. *Acta Mater.* **2003**, *51*, 4411–4421. [CrossRef]
7. Cheng, K.; Chen, W.; Liu, D.; Zhang, L.; Du, Y. Analysis of the Cermak–Rothova method for determining the concentration dependence of ternary interdiffusion coefficients with a single diffusion couple. *Scr. Mater.* **2014**, *76*, 5–8. [CrossRef]
8. Dash, A.; Esakkiraja, N.; Paul, A. Solving the issues of multicomponent diffusion in an equiatomic NiCoFeCr medium entropy alloy. *Acta Mater.* **2020**, *193*, 163–171. [CrossRef]
9. Esakkiraja, N.; Gupta, A.; Jayaram, V.; Hickel, T.; Divinski, S.V.; Paul, A. Diffusion, defects and understanding the growth of a multicomponent interdiffusion zone between Pt-modified B2 NiAl bond coat and single crystal superalloy. *Acta Mater.* **2020**, *195*, 35–49. [CrossRef]
10. Zhang, Q.; Zhao, J. Extracting interdiffusion coefficients from binary diffusion couples using traditional methods and a forward-simulation method. *Intermetallics* **2013**, *34*, 132–141. [CrossRef]
11. Bouchet, R.; Mevrel, R. A numerical inverse method for calculating the interdiffusion coefficients along a diffusion path in ternary systems. *Acta Mater.* **2002**, *50*, 4887–4900. [CrossRef]
12. Andersson, J.; Ågren, J. Models for numerical treatment of multicomponent diffusion in simple phases. *J. Appl. Phys.* **1992**, *72*, 1350–1355. [CrossRef]
13. Cheng, K.; Zhou, J.; Xu, H.; Tang, S.; Yang, Y. An effective method to calculate the composition-dependent interdiffusivity with one diffusion couple. *Comput. Mater. Sci.* **2018**, *143*, 182–188. [CrossRef]
14. Cheng, K.; Xu, H.; Ma, B.; Zhou, J.; Tang, S.; Liu, Y.; Sun, C.; Wang, N.; Wang, M.; Zhang, L.; et al. An in situ study on the diffusion growth of intermetallic compounds in the Al–Mg diffusion couple. *J. Alloys Compd.* **2019**, *810*, 151878. [CrossRef]
15. Cheng, K.; Sun, J.; Xu, H.; Wang, J.; Zhan, C.; Ghomashchi, R.; Zhou, J.; Tang, S.; Zhang, L.; Du, Y. Diffusion growth ϕ ternary intermetallic compound in the Mg-Al-Zn alloy system: In-situ observation and modeling. *J. Mater. Sci. Technol.* **2020**, in press.
16. Qin, Y.; Narayan, A.; Cheng, K.; Wang, P. An efficient method of calculating composition-dependent inter-diffusion coefficients based on compressed sensing method. *Comput. Mater. Sci.* **2020**, *188*, 110145. [CrossRef]
17. Robert, C. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007.
18. Girolami, M. Bayesian inference for differential equations. *Theor. Comput. Sci.* **2008**, *408*, 4–16. [CrossRef]
19. Alahmadi, A.A.; Flegg, J.A.; Cochrane, D.G.; Drovandi, C.C.; Keith, J.M. A comparison of approximate versus exact techniques for Bayesian parameter inference in nonlinear ordinary differential equation models. *R. Soc. Open Sci.* **2020**, *7*, 191315. [CrossRef]
20. Wilkinson, R. Accelerating ABC methods using Gaussian processes. In Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, Reykjavik, Iceland, 22–25 April 2014; pp. 1015–1023.
21. Conrad, P.R.; Marzouk, Y.M.; Pillai, N.S.; Smith, A. Accelerating Asymptotically Exact MCMC for Computationally Intensive Models via Local Approximations. *J. Am. Stat. Assoc.* **2016**, *111*, 1591–1607. [CrossRef]
22. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.

23. Calderhead, B.; Girolami, M.; Lawrence, N.D. Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 7–10 December 2009; pp. 217–224.
24. Albert, C.G. Gaussian processes for data fulfilling linear differential equations. *Proceedings* **2019**, *33*, 5. [CrossRef]
25. Yang, S.; Wong, S.W.; Kou, S. Inference of dynamic systems from noisy and sparse data via manifold-constrained Gaussian processes. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2020397118. [CrossRef]
26. Xing, W.; Elhabian, S.; Kirby, R.; Whitaker, R.; Zhe, S. Infinite ShapeOdds: Nonparametric Bayesian Models for Shape Representations. In Proceedings of the AAAI 2020: The Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020
27. Zhe, S.; Qi, Y.; Park, Y.; Molloy, I.; Chari, S. DinTucker: Scaling up Gaussian process models on multidimensional arrays with billions of elements. *arXiv* **2013**, arXiv:1311.2663.
28. Fick, A. Ueber Diffusion. *Pogg. Ann.* **1855**, *94*, 59–86. [CrossRef]
29. Candes, E.; Romberg, J. *l1-Magic: Recovery of Sparse Signals via Convex Programming*; 2005; Volume 4, p. 14. Available online: www.acm.Caltech.Edu/l1magic/downloads/l1magic.pdf (accessed on 26 June 2021).
30. Berg, E.V.; Friedlander, M.P. Probing the Pareto frontier for basis pursuit solutions. *Sci. Comput.* **2008**, *31*, 890–912.
31. Sturm, J.F. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Method Softw.* **1999**, *11*, 625–653. [CrossRef]
32. Snelson, E.; Ghahramani, Z. Sparse Gaussian Processes using Pseudo-inputs. In *Advances in Neural Information Processing Systems 19*; MIT Press: Cambridge, MA, USA, 2006; pp. 1257–1264.
33. Quiñonero-Candela, J.; Rasmussen, C.E. A unifying view of sparse approximate Gaussian process regression. *J. Mach. Learn. Res.* **2005**, *6*, 1939–1959.
34. Kennedy, M.C.; O'Hagan, A. Bayesian calibration of computer models. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2001**, *63*, 425–464. [CrossRef]
35. Bastos, L.; O'Hagan, A. Diagnostics for Gaussian Process Emulators. *Technometrics* **2009**, *51*, 425–438. [CrossRef]